Imperial College of Science, Technology and Medicine
Department of Physics

Computational acquisition of knowledge in small-data environments: a case study in the field of energetics

Sinead O'Brien

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Physics of Imperial College London, September 2022

Abstract

The UK's defence industry is accelerating its implementation of artificial intelligence, including expert systems and natural language processing (NLP) tools designed to supplement human analysis. This thesis examines the limitations of NLP tools in small-data environments (common in defence) in the defence-related energetic-materials domain. A literature review identifies the domain-specific challenges of developing an expert system (specifically an ontology). The absence of domain resources such as labelled datasets and, most significantly, the preprocessing of text resources are identified as challenges. To address the latter, a novel general-purpose preprocessing pipeline specifically tailored for the energetic-materials domain is developed. The effectiveness of the pipeline is evaluated.

Examination of the interface between using NLP tools in data-limited environments to either supplement or replace human analysis completely is conducted in a study examining the subjective concept of importance. A methodology for directly comparing the ability of NLP tools and experts to identify important points in the text is presented. Results show the participants of the study exhibit little agreement, even on which points in the text are important. The NLP, expert (author of the text being examined) and participants only agree on general statements. However, as a group, the participants agreed with the expert. In data-limited environments, the extractive-summarisation tools examined cannot effectively identify the important points in a technical document akin to an expert.

A methodology for the classification of journal articles by the technology readiness level (TRL) of the described technologies in a data-limited environment is proposed. Techniques to overcome challenges with using real-world data such as class imbalances are investigated. A methodology to evaluate the reliability of human annotations is presented. Analysis identifies a lack of agreement and consistency in the expert evaluation of document TRL.

Acknowledgements

Firstly, I would like to thank my supervisor Dr Proud for giving me the opportunity to undertake this research and for our discussions during our lunches together. Additionally, I would like to thank Dr Wilson for her research input and invaluable direction on the psychology. I would also like to thank my colleagues at the ISP. Thank you to all the people who participated in this research, notably, Clive, Dr Turnell and Eamon.

I would also like to thank my sponsors Qinectiq and Mr Callaway for funding this research and providing invaluable feedback throughout the course of this project. In addition, thank you to my thesis examiners Dr Vrcelj and Prof. Chana for the effort and time they spent assessing my work.

Thank you to my friends (L, BY, M, S and MM) for brightening my life. Thank you to Emma for being a true friend, I would not have made it to this point without you. I can't believe we have both OFFICIALLY done it - after 8 years!!!

Furthermore, I would like to thank Mum, Dad, Liam, Ciaran, Sharon, Dave, Darcy, Fish, Chip, Lord Fishfinger, Midnight and Mercury for their encouragement. Finally, I could not have undertaken this journey without Connor and his unwavering support and belief in me. Thank you for your tremendous understanding and encouragement, especially for reading this thesis from start to finish and suggesting improvements.

'Kim, there's people that are dying.'

Kourtney Kardashian

Contents

Abstract i

Acknowledgements iii

1 Introduction 1

1.1 Outline and contributions . . . 3

1.2 Statement of originality . . . 5

1.3 Copyright declaration . . . 5

1.4 Publications . . . 6

2 Knowledge representation and ontologies 7

2.1 Knowledge representation and expert systems . . . 8

2.2 Knowledge representation with ontologies . . . 10

2.3 A definition of ontology . . . 12

2.4 The Semantic Web . . . 14

2.5 Ontology representation languages . . . 15

2.6 Classifying ontologies . . . 17

2.7 Ontology construction . . . 18

2.8 Conclusion . . . 19

3 Ontology construction from text **21**

3.1 The Ontology Layer Cake . . . 22

3.2 Language models . . . 23

3.3 Ontology learning . . . 25

3.4 Ontology evaluation . . . 37

3.5 Ontology learning tools . . . 41

3.6 Applying ontology learning to the energetic-materials domain . . . 47

3.7 Conclusion . . . 50

4 Comparing NLP tools' and experts' ability to summarise technical text **53**

4.1 A literature review of NLP-summarisation techniques . . . 54

4.2 Methodology . . . 72

4.3 Results and discussion . . . 81

4.4 Conclusion . . . 101

Chapter 4 appendices . . . 106

5 Elicitation of knowledge from a defence expert **113**

5.1 Elicitation of knowledge . . . 114

5.2 Capturing expert understanding . . . 115

5.3 Results . . . 118

5.4 Analysis . . . 122

5.5 Conclusion . . . 128

6 Comparing NLP tools', experts' and novices' analysis of technical text 130

6.1 Inter-rater reliability . . . 131

6.2 Methodology . . . 134

6.3 Results . . . 139

6.4 Conclusions and future work . . . 161

7 Developing a preprocessing pipeline for the energetic-materials domain 167

7.1 Preprocessing conference proceedings in the energetic-materials domain . . . 168

7.2 Methodology . . . 175

7.3 Evaluation and discussion of pipeline performance . . . 182

7.4 Conclusion . . . 184

8 Predicting the Technology Readiness Level (TRL) of journal articles 187

8.1 Background . . . 189

8.2 Methodology . . . 198

8.3 Results and discussion . . . 204

8.4 Conclusions of preliminary study . . . 213

9 Comparing experts' and NLP tools' classification of journal articles by TRL 216

9.1 Class imbalance . . . 217

9.2 Methodology . . . 219

9.3 Results and discussion . . . 225

9.4 Conclusions and further work . . . 247

10 Conclusion **253**

10.1 Research summary 253

10.2 Research impact 257

10.3 Future work 259

Bibliography **260**

List of Tables

3.1 Adapted from Ref. [220]. Overview of ontology evaluation showing the layers of an ontology each method can evaluate. 38

3.2 Summary of ontology learning systems and which components of the Ontology Layer Cake each system generates. If a particular system generates a part of the Ontology Layer Cake, it is marked with a '✓' in the corresponding cell of the table. 46

4.1 Percentage of correctly-stemmed words in four pages of the text using the Lovins, Paice-Husk and Porter stemming algorithms. 76

4.2 Evaluation of summaries generated by SumBasic, LSA, TextRank and bert-extractive-summarizer with Sci-BERT, BERT and GTP-2 against the expert's keypoints using the metrics $F_{\text{ROUGE-1}}$ and F_C. 81

4.3 Evaluation of 'TextRank Basic Jacquard', 'TextRank cosine TF-ISF' and 'TextRank Word2Vec cosine' against the expert's keypoints using $F_{\text{ROUGE-1}}$ and F. 85

5.1 Expert's keypoints, assigned importance rating and a label assigned to each keypoint. 119

5.2 Keypoints sorted into five categories according to importance, 1 being the least important and 5 being the most important. 120

5.3 Keypoints sorted into four distinct categories generated by the expert. 121

5.4 Generalised categories of importance, the corresponding definition of each category and the points sorted into each category. 124

5.5 Categories generated by the expert with the points replaced by their corresponding reasoning categories. 126

5.6 Expert's top-rated points alongside the importance category assigned to each point in the interview task. 127

6.1 Contingency table for two raters: 1 and 2 on a binary scale (Outcome A and Outcome B). 131

6.2 Interpretation of Cohen's kappa according to the Landis and Koch benchmark scale [355, 353]. 132

7.1 Average percentage of correct sentences returned by the pipeline ($S_{\%A\cap B}$), incorrect sentences retrieved ($S_{\%A'\cap B}$), chemical entities returned by the pipeline ($CE_{\%A\cap B}$), and unique chemical entities returned by the pipeline ($CE_{\%A*\cap B*}$) for the HEMCE and NTREM test sets. 183

8.1 UK MoD's definition of Technology Readiness Levels (TRLs) [415] 188

8.2 Lists of keywords associated with High-TRL and Low-TRL generated by the expert. 199

8.3 Performance of the algorithms evaluated using the metrics F_1, R, P, ROC AUC and A. Estimated values of A are displayed for the linear SVM, decision tree, k-NN, NB, logistic regression and random forest classifiers, as these methods used nested cross-validation. 205

9.1 Performance of supervised classifiers using S-BERT for feature representation with PCA included as part of the nested cross-validation (CV) and without. Evaluation is conducted using the metrics: $F_{<01>}$, AUC ROC and A. Estimated A is generated using nested cross-validation. 226

9.2 Performance of decision tree, logistic regression and random forest classifiers with and without techniques to mitigate the class imbalance (RUS, class weighting, synthetic labels and SMOTE). Evaluation is conducted using the metrics: $F_{<01>}$, AUC ROC and A. Values of Estimated A are generated using nested cross-validation 229

9.3 Performance of the K-means clustering classifier for S-BERT and TF-IDF feature-representation methods. Evaluation is conducted using the metrics: $F_{<01>}$, AUC ROC and A. 234

9.4 Results of K-means clustering (TF-IDF 9), logistic regression (S-BERT SL) and a keyword-matching algorithm for document classification by TRL on the NTREM dataset. 239

9.5 Results of K-means clustering (TF-IDF 9), logistic regression (S-BERT SL) and keyword-matching classifiers on the HEMCE dataset. 241

9.6 κ for all possible pairings of experts on their classification of 108 NTREM articles by TRL. Interpretation of κ is completed using Landis and Koch benchmark scales. 243

9.7 Marginals for all possible pairings of experts on their classification of 108 NTREM articles by TRL. The marginals are examined for bias and prevalence. 243

9.8 $\hat{\kappa}_G$ and associated error (SE) for all possible pairings of experts on their classification of 108 NTREM articles by TRL. Interpretation of $\hat{\kappa}_G$ is completed using a statistical approach described in Section 6.1 with Landis and Koch benchmark scales. Interpretation of κ using Landis and Koch benchmark scales is computed for each pair. 244

9.9 Marginals for all possible pairings of experts on their classification of 252 NTREM articles by TRL. The marginals are examined for bias and prevalence. 245

9.10 $\hat{\kappa}_G$ and associated error (SE) for all possible pairings of experts on their classification of 108 NTREM articles by TRL. Interpretation of $\hat{\kappa}_G$ is completed using a statistical approach described in Section 6.1 with Landis and Koch benchmark scales. For comparison, agreement using κ is calculated. Interpretation of κ using Landis and Koch benchmark scales is computed for each pair. 245

9.11 $\hat{\kappa}_G$ for the dataset of 104 documents and participants A, B, C, D)and the dataset of 252 documents and Participants B, C, D. 246

List of Figures

2.1 RDF graph of a triple. 16

3.1 Adapted from Ref. [115]. The task of developing an ontology can be viewed as a collection of subtasks, each represented as a layer of the Ontology Layer Cake. These subtasks are completed in order starting at the bottom of the cake. Each subtask builds on those preceding it. 22

4.1 Tasks completed by the NLP are shown in turquoise, those completed by the readers in lilac and those completed by the expert in orange. First, the text was preprocessed. Then the readers and the expert were requested to fill in a background-information questionnaire (BI). Next, the NLP summarised the document, and the readers and expert completed the keypoints exercise. Finally, a comparison was conducted on the output of the summarisation/keypoints tasks. 73

4.2 Overview of the methodology used to preprocess the document. 75

4.3 Proportion of readers by: numbers of years of experience in academic research (a) or commercial research (c), number of years total experience (b) and expertise rating (d). 87

4.4 Distribution of $MRIS$ (a) and $RAIS$ (b) for the reader-agreed points (RAPs). . 89

4.5 Distribution of $MRIS$ (a) and $RAIS$ (b) for the reader-agreed points (RAPs) in agreement with the expert and not in agreement with the expert. Red crosses indicate outliers and orange lines indicate median scores. 90

4.6 Distribution of $MRIS$ (a) and $RAIS$ (b) for the reader-agreed points (RAPs) for those in agreement with the NLP and not in agreement with the NLP. Red crosses indicate outliers and orange lines indicate the median scores. 93

4.7 Matrix displaying the F_C overlap of the keypoints/summaries for every possible pairing of individuals in the group and the NLP. 95

4.8 Average F_C among the readers ($\langle F_C \rangle$) for each participant and the NLP. 96

4.9 $F_{\text{ROUGE-1}}$ overlap of the keypoints/summaries for every possible pairing of individuals in the group containing participants and the NLP. 97

4.10 Average $F_{\text{ROUGE-1}}$ among the readers ($\langle F_{\text{ROUGE-1}} \rangle$) for each participant and the NLP. 98

4.11 F versus $F_{\text{ROUGE-1}}$ for overlap between participants, expert and NLP summaries. A regression line is fitted to the plot and 95% confidence intervals are bootstrapped, represented by the red-shaded region of the graph. 100

4.12 Overview of the decision-making process for determining the appropriate statistical test. Statistical tests are shown in blue. It is noted that the Mann-Whitney U test is also referred to as the Wilcoxon Rank Sum test or the Mann-Whitney Wilcoxon test. 112

5.1 Example of a keypoints card. Points were labelled A-Q in the top right-hand corner to aid the recording process. 116

5.2 Set-up for the rating of the keypoints. Keypoints are placed into one of five categories according to their relative importance. 117

5.3 Sorting task configuration. Keypoints are sorted into categories. 117

6.1 Tasks completed by the NLP/participants/expert are shown in turquoise/lilac/orange respectively. First, the preprocessing methodology from Chapter 4 is applied to the text. The participants are asked to complete a three-part online questionnaire. Only the first two parts of the questionnaire are relevant to this method. Part 1 asks the participants to answer background-information questions. Part 2 is a summarisation exercise. Next, the best-performing NLP summarisation algorithm from Chapter 4 is used to generate a summary. Finally, a comparison of the generated summaries is conducted. 136

6.2 Example of a question from the statement-analysis task. Participants are asked to assign statements an importance rating from a drop-down menu. 138

6.3 Example of a question from the statement-analysis task. Participants are asked to assign a category of importance to statements using a drop-down menu. . . . 139

6.4 Overview of the number of years of experience of the readers in academic research (a), commercial research (b), government research (c) and total (d). The number of years of experience are divided into ordinal categories as shown in the legends. 140

6.5 Overview of the STEM higher-education qualifications and self-assigned expertise ratings of the readers. 141

6.6 Index of each sentence plotted against the assigned $RAIS'$ score. 143

6.7 Distribution of $RAIS'$ for sentences identified (respectively, not identified) as important by the expert are on the left (respectively, right). 144

6.8 Distribution of $RAIS'$ for sentences identified (respectively, not identified) as important by the NLP are on the left (respectively, right). 145

6.9 Similarity (F_A) between the readers, experts and NLP summaries. The colour bar displays the relationship between F_A and the cell colour. 147

6.10 Average overlap of each of the participants/NLP summaries with all other reader summaries, $\langle F_A \rangle'_{R_i}$. Values are calculated with all readers (red line) and with the novices removed from the dataset (blue line). 149

6.11 Agreement matrix calculated using $\hat{\kappa}_G$. For interpretation of $\hat{\kappa}_G$, the Landis and Koch scale is shown on the colour bar. 152

6.12 $\langle \hat{\kappa}_G \rangle_{R'}$ for each participant, where all readers, and without the novices in the dataset, are shown in red and blue respectively. 153

6.13 Distribution of $MRIS'$ for statements rated as important by the author and rated as not important by the expert. 155

6.14 Distribution of importance ratings by the expert for statements identified as important and not identified as important by the expert in the preliminary study. 156

6.15 $\hat{\kappa}_G$ between every possible pair of participants for assigning categories of importance to statements about the text. Interpretation of $\hat{\kappa}_G$ using the Landis and Koch scale is shown on the colour bar. 158

6.16 $\langle \hat{\kappa}_G \rangle_{R'}$ for each participant. This calculation was done with all readers and with the novices removed, shown in red and blue respectively. 159

7.1 Overview of the proposed preprocessing pipeline, consisting of three main stages: text extraction, tokenisation and text normalisation. 175

8.1 5-fold cross-validation. A training dataset is partitioned into 5 subsections. Then, the model is trained on 4/5 of the training dataset and tested on the remaining 1/5 of the training dataset to obtain an estimate of model performance. This process is repeated 5 times: testing and training on different combinations of the subsets of the training data. Finally, an estimated accuracy (Estimated A) and a generalisation error is generated by taking the mean and standard deviation of the 5 estimated accuracies A_{e_i}. 195

8.2 Distribution of A_e of decision tree, K-NN, Gaussian NB, logistic regression, random forest and linear SVM classifiers with TF-IDF and S-BERT feature-representation methods. A_e are obtained using nested cross-validation. 208

8.3 Frequency of selection of the hyperparameter C as a result of hyperparameter optimisation in the inner loop of nested cross-validation over the hyperparameter grid [0, 1, 0.5, 1.5, 0.1, 0.01, 0.001, 0.0001, 5] for a linear SVM classifier. 209

8.4 Visualisation of K-means clustering algorithm performance. Each document in the test set is transformed to 2D coordinates using PCA. Documents labelled as High-TRL are shown in blue and documents labelled as Low-TRL as shown in green. Misclassified documents have a superimposed red cross. 212

9.1 Visualisation of K-means clustering classifier performance with feature representation of TF-IDF (a) and S-BERT (b). Each document in the test set is transformed to 2D coordinates using PCA. Documents labelled as High-TRL are shown in blue and those labelled as Low-TRL are shown in green. Misclassified documents have a superimposed red cross. 233

9.2 2D PCA representation of document vectors labelled by their assigned TRL. Documents labelled as Low-TRL are shown in green, and documents labelled as High-TRL are shown in blue. 236

9.3 AUC ROC, $F_{<01>}$ and S_c of K-means clustering classifier with TF-IDF whereby a PCA transform has been applied versus the d_n after reduction. 237

9.4 Frequency of selection of d_n for PCA as a result of optimisation in the inner loop of nested cross-validation over the grid [50,60,70,80,90,100], for logistic regression (S-BERT SL). 240

Chapter 1

Introduction

An increasingly pressing problem in the field of defence is the loss of expert knowledge from the retirement of key personnel[1]. Over a prolonged period in an organisation, an employee will develop deeply contextualised knowledge of specific business systems and procedures. When this employee retires, their expert knowledge is lost and is difficult to replace [2]. The significance of this problem is becoming more apparent globally due to an increasingly ageing population [3]. For example, by 2040, approximately one in seven people in the UK will be over the age of 75 [4]. As the UK population ages, retirement rates will rise, leading to an increasing number of workers leaving the workforce; hence, some roles will be left vacant due to a lack of suitable replacements [5]. In extreme cases, this can result in organisations resorting to rehiring retired employees at consultancy rates to resolve complex problems [6].

One potential solution to the loss-of-knowledge problem is the development of expert systems that could aid the preservation and transfer of expert knowledge by emulating expert decision-making. Expert systems generally consist of a knowledge base and an inference system [7]. The knowledge base, typically designed by a knowledge engineer, consists of facts and information as well as data considered in the context of the domain in question. The inference system uses the knowledge base to obtain a solution to a problem [8].

However, since the earliest work on expert systems, computer scientists and psychologists have struggled with issues around eliciting expert knowledge for the knowledge base [9, 10]. Popular

[1]The present introduction is based on the introduction to my paper in Ref. [1], see Section 1.4 for details on the rights to reproduce Ref. [1].

knowledge acquisition methods in the early 1990s relied on knowledge acquisition methods from psychology [11]. In recent years these methods have become less popular due to 'information overload' [11]. The quantity of information on the web has dramatically increased through the development of social media and the web-publishing of journal articles. Information overload is a pressing issue in industry, government, and academia [12]. A 2020 report on the future of artificial intelligence and UK national security described this as 'one of the greatest technical challenges facing the UK's national security community' [13].

Expert systems have consequently evolved to exploit the large quantities of data readily available [14]. As a result, those involved in developing expert systems have turned to computerised methods of processing large quantities of text data, such as Natural Language Processing (NLP) tools [15]. NLP is defined as the application of computational techniques to analyse language and speech [16][2]. In the era of big data, harnessing the power of NLP to obtain actionable insights from text is a rapidly increasing area of research, not just in the context of expert systems [17, 18, 19, 20]. For instance, in the biomedical domain, biomedical text mining (or BioNLP) is a well-established area of research [21, 22], with applications to finding protein-gene interactions [23] and identifying biomedical relations, such as between diseases and medications [24, 25].

The race to adapt to 'information overload' by leveraging AI capabilities is a race in which the UK and its allies are falling behind [26]. In light of this, the UK's defence and security industries are taking steps to accelerate the implementation of AI. This includes implementing a range of NLP systems that can 'support' human operators to analyse and organise large volumes of unstructured text [13], e.g. by flagging significant items.

State-of-the-art NLP tools rely on large labelled datasets (either to demonstrate the performance of unsupervised methods or to train the NLP algorithms themselves). Development of these datasets is a costly and time-consuming process, one that is often not feasible when resources are constrained [27]. This is a hurdle in information-restricted fields, such as the defence industry, which are often the most affected by the loss-of-knowledge problem. In defence, information can be complex and hidden in data silos, making the development of these datasets unfeasible [28]. This thesis examines the limitations of NLP tools in small-data environments.

[2]This definition of NLP will be used throughout this thesis. It is noted that the acronym 'NLP' can be an abbreviation of phrases not relevant to this thesis, such as the pseudoscientific field of Neuro-linguistic programming.

Since NLP tools are designed to supplement human analysis, rather than replace the role of the analyst altogether, the focus of this thesis exists at this interface, comparing the ability of NLP tools in data-limited environments to domain experts on the analysis of technical information concerned with inherently subjective concepts.

The scope of this work is focused on the defence-related field of 'energetic materials'. This is an interdisciplinary field where limited work has been conducted on the use of NLP tools. The use of NLP tools in this field poses a unique set of challenges. This thesis will examine the challenges that using NLP tools to develop an expert system in the domain of energetic materials may face.

To conclude, the underlying theme of this thesis could be considered as the limitations of NLP tools in small-data environments in the domain of energetic materials. The aims of this thesis are summarised as follows

- Identification of the challenges of using NLP tools to develop an expert system in the energetic materials domain.
- Examination of the interface between using NLP tools to either supplement or replace human analysis.

1.1 Outline and contributions

This thesis consists of 10 Chapters.

Chapter 1: Context and motivations for this work are introduced. Thesis aims, outline and contributions are summarised.

Chapter 2: First part of a two-chapter literature review investigating the challenges of using NLP tools to develop an expert system in the energetic materials domain. This chapter provides an in-depth introduction to knowledge representation and expert systems with a focus on ontology. A literature review consisting of the definition of an ontology, the semantic web, ontology representation languages, classification of ontologies and ontology construction is presented.

Chapter 3: Second part of a two-chapter literature review investigating the challenges of using NLP tools to develop an expert system in the energetic materials domain. This chapter provides the reader with a comprehensive overview of ontology learning from text with a focus on tools that are applicable for small datasets. Domain-specific challenges for ontology construction from text in energetic materials are identified.

Chapter 4: Preliminary study investigating the boundary between using NLP tools to either supplement or replace human analysis. This chapter provides the reader with a review of existing summarisation techniques for the suitability of summarising a single document in a niche domain (energetic materials) and introduces an adapted methodology for comparing the ability of NLP tools in data-limited environments and domain experts to summarise technical information in a single document in the energetic materials domain according to a subjective concept (importance). Furthermore, a clear set of guidelines for future studies of this nature are specified.

Chapter 5: First part of a two-chapter follow-up study to Chapter 4. This chapter provides an overview of techniques for knowledge elicitation and presents a novel methodology for elicitation of expert knowledge. Reasons why the expert finds specific points in a single document important are elicited. This is the first open-source investigation into the elicitation of knowledge in the domain of energetic materials.

Chapter 6: This is the second part of a two-chapter follow-up study to Chapter 4. This chapter applies recommendations from Chapter 4 and insights from Chapter 5 to develop a novel methodology for comparing the ability of NLP tools, novices and experts to identify important points in a single document. The reader is provided with an overview of inter-rater reliability metrics and introduced to an adapted methodology for assessing the agreement amongst experts and novices on how they evaluate the importance of technical information in a single document.

Chapter 7: This chapter addresses a significant challenge for ontology construction from text in the energetic materials domain identified in Chapter 3, preprocessing and extraction of text. The challenges of preprocessing PDFs of conference proceedings in the energetic materials domain are identified through a literature review. A novel general-purpose preprocessing pipeline catered to small datasets of conference proceedings in the energetic materials domain is described. Then, an in-depth qualitative evaluation of the preprocessing pipeline is conducted.

Chapter 8: This chapter addresses the limitations of using NLP tools in a small-data environment. Chapters 3 and 4 discussed the approaches to various NLP tasks, such as text summarisation in the data-limited environment of energetic materials. Supervised algorithms were identified as high-performance methods but were ruled out as they require large labelled datasets. In this chapter, a preliminary study compares supervised and unsupervised techniques for document classification on a small dataset of NTREM journal articles. This work establishes a series of recommendations applicable to future studies classifying long text in data-limited environments. Furthermore, the work in this chapter demonstrates the usefulness of the Chapter 7 preprocessing pipeline.

Chapter 9: This chapter is a follow-up study to Chapter 8. Recommendations from Chapter 8 are applied to compare the ability of supervised and unsupervised classifiers to classify the TRL of a small dataset of NTREM articles. The flexibility of the classifiers is tested by examining their ability to classify a dataset of HEMCE articles. The work in this chapter will refine the methodology presented in Chapter 8 by examine the effectiveness of a range of techniques to mitigate the negative effects of a class imbalance and examining the impact of PCA transforms on classifier performance. In addition, the reliability of the data used to train these classifiers is examined by examining inter-participant agreement and test-retest agreement on TRL classification of NTREM articles. This work demonstrates the importance of identifying the most suitable agreement metric for analysis.

Chapter 10: This chapter summarises the key findings in relation to the aims of this thesis, the impact of the research conducted in this thesis and makes recommendations for future work.

1.2 Statement of originality

I declare the work in this thesis to be my own except where referenced in the text.

1.3 Copyright declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial-No Derivatives 4.0 Interna-

tional Licence (CC BY-NC-ND).

Under this licence, you may copy and redistribute the material in any medium or format on the condition that; you credit the author, do not use it for commercial purposes and do not distribute modified versions of the work.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

1.4 Publications

- O'Brien, Sinead, William G. Proud, and Margaret A. Wilson. 'Elicitation of knowledge from a defence expert.' Journal of Physics: Conference Series. Vol. 1507. No. 10. IOP Publishing, 2020.[3]
- O'Brien, Sinead, William G. Proud. 'Predicting the technology readiness level of NTREM papers.' New trends in research of energetic materials conference proceedings, 2022.[4]

[3]Work from this paper is included in this thesis and is reproduced under the Creative Commons Attribution 3.0 licence which allows me to copy and redistribute the material in any medium or format.

[4]Work from this paper is included in this thesis. After checking with the editor of the journal I confirmed I have retained all rights to reproduce work in this paper including for use in this thesis.

Chapter 2

Knowledge representation and ontologies

In Chapter 1, loss-of-knowledge due to the retirement of key personnel was identified as a critical problem in defence and security-related fields such as energetic materials. Expert systems and other knowledge-representation methods may be a solution to this problem. In the last two decades, methods of producing such systems have evolved considerably to leverage NLP tools. This chapter forms the first part of a two-Chapter literature review on the challenges of using NLP tools to develop an expert system in the energetic-materials domain. This chapter aims to provide the reader with an overview of knowledge-representation and ontologies.

Chapter achievements:

- In-depth introduction to knowledge-representation with a focus on ontologies
- Literature review comprising the definitions of an ontology, the Semantic Web, ontology representation languages, classification of ontologies and ontology construction.

Section 2.1 gives an overview of knowledge-representation and expert systems. In Section 2.2, knowledge-representation with ontologies is introduced. In Section 2.3, definitions of ontologies are reviewed before establishing a preferred definition for this thesis. A noticeable driver of change in the field of ontology lies in one of its applications - the Semantic Web. In Section

2.4, a brief introduction to the Semantic Web is given. Section 2.5 gives an overview of ontology representation languages and then, in Section 2.7, ontology construction methods are introduced. Finally, in Section 2.8, the conclusions of the Chapter are presented.

2.1 Knowledge representation and expert systems

Knowledge-representation (KR) could provide a possible solution to the loss-of-expert-knowledge problem. The ultimate goal of KR is to develop a computational model that replicates human intelligence [29]. KR is arguably the most fundamental problem in artificial intelligence (AI) [30, 31]. Artificial intelligence is the development of computer systems able to complete tasks requiring human intelligence [32]. Examples of such tasks include visual perception, decision making and translation between languages. An intelligent computer system is required to comprehend the extent of the domain it is assigned to understand how to compute necessary tasks [30]. KR tackles the problem of how to represent knowledge in a machine- and human-understandable format that enables intelligent (human-like) inferencing in knowledge-based systems. Knowledge-based systems are an umbrella term for an AI that represents and performs inferences on knowledge. Such a system comprises a knowledge base and an inferencing facility. An inferencing facility aims to perform expert-like reasoning to find solutions to specific problems using the knowledge in the knowledge base [33]. The knowledge base is a structure (such as an ontology) used to store knowledge.

The first knowledge-representation methods, also referred to as expert systems, were designed to solve specific problems [34, 35]. For example, identifying and prescribing Meningitis treatments from a list of symptoms [36]. Generally, these early expert systems consisted of production rules (IF-THEN statements). The computer would then iterate over these rules to generate an outcome. An example of such a system is given in Ref. [37], this system is designed to aid analytical chemists in selecting the appropriate methodology for high-performance liquid chromatography. IF-THEN statements such as, 'IF the Compound has a functional group AND this group absorbs in the UV, THEN the Compound can be detected by a UV absorbance detector.', are used to determine the appropriate method from a list of the properties of the compound. Knowledge for these systems was acquired in what was viewed as a transfer process, from an expert to a knowledge engineer (whose role is to encode the knowledge into the

system) [38]. Viewing knowledge acquisition for these systems as a transfer problem raised a fundamental problem: experts find it challenging to formalise their knowledge [39, 31]. The difficulty surrounding obtaining knowledge for a knowledge-based system is commonly referred to as the knowledge-acquisition bottleneck [40].

The knowledge-acquisition bottleneck came to be viewed as a psychological problem caused by most of the expert's knowledge being implicit and needing to be 'mined' using knowledge-elicitation techniques [41, 42, 39]. Consequently, the knowledge-acquisition process evolved from interviews lacking any psychological justification to structured interviews using these elicitation techniques. While these knowledge-representation techniques facilitate a more effortless transfer of knowledge, it became apparent that the transfer process perspective of knowledge acquisition is not adequate [43, 44, 45]. Therefore, knowledge acquisition is now viewed as the process of modelling knowledge [43, 44]. Instead of aiming to recreate knowledge in the mind of an expert (transfer approach), the modelling approach attempts to recreate an approximation of the expert's mind [44].

The popularity of expert systems declined in the late 1980s as companies realised these systems often were not worth their upkeep [46]. Knowledge encoded in an expert system would quickly become outdated as practises related to the purpose of the system evolved, leaving the expert system redundant. Furthermore, the primary method of acquiring knowledge was based on sit-down, face-to-face interviews with an expert. As a result, the development of expert systems was time and monetarily expensive [30, 47]. Maintenance also requiring expert input was a costly and challenging task.

Expert systems did not separate procedural knowledge, designed to help the system solve a specific task, from domain knowledge. However, in the 1990s, work such as in Ref. [48] led to a new approach to developing knowledge bases whereby domain and procedural knowledge are stored separately. An example of this type of knowledge-based system is an ontology. Throughout the 1990s and early 2000s, interest in ontologies as a knowledge-representation method grew due to a motivation to create machine-understandable representations of domain knowledge for artificial intelligence [49].

Today, ontologies are widely established methods of representing knowledge in a structured and machine-readable format, facilitating knowledge sharing and reuse. Ontologies are widely used in a variety of domains [50, 51, 52, 53, 54, 55, 56]. In the biology domain, ontologies such as the

Gene Ontology (GO) [57], have been used to aid understanding in this instance by providing researchers with useful and current information concerning genes and hence facilitate scientific research [58]. However, ontologies are not used as widely in the physics and chemistry domains [59, 60]. A notable example in the physics domain is the Orbital Debris Ontology (ODO) [61]. This ontology aims to facilitate knowledge sharing and integration of multi-source data to allow for improved orbital debris tracking and increased awareness of orbital hazards, thus allowing for safer space exploration.

Given the popularity in the KR field of ontologies, and their demonstrated benefits in other scientific domains, the focus of this review is confined to ontologies. Currently, there are no open-source ontologies in the domain of energetic materials; therefore, exploring ontologies in this context is of interest to the field of energetics.

2.2 Knowledge representation with ontologies

The term 'ontology' is derived from philosophy, specifically from a subsector of metaphysics concerned with the nature of being. Ontology in this context is the study of categories of things that exist or may exist in the world [62]. AI adopted the term ontology to describe computational representations of the world [43]. Most modern ontologies represent domain knowledge.[1]

Ontologies aim to capture knowledge in a manner that is independent of use. Therefore, the knowledge represented in an ontology should reflect an objective description of an area of interest (domain). This method of representing knowledge is in contrast to the more traditional view of knowledge in expert systems which views the knowledge in a knowledge base as a repository of knowledge extracted from the mind of an expert [64]. This viewpoint aligns with the transfer perspective on knowledge acquisition introduced in Section 2.1. Domain knowledge and task-specific knowledge were elicited together in expert systems. Knowledge was hence considered in a strictly functional manner [65], whereby the criteria used to evaluate its relevance is for a particular function as opposed to whether or not it is objectively true in all contexts in the domain.

[1] There are some exceptions such as the ambitious Cyc project which aims to create a knowledge base of all 'common-sense' knowledge in the world [63].

The involvement of philosophers in the knowledge engineering community led to a shift in perspective away from the functional view of knowledge. Clancy argued, 'the primary concern of knowledge engineering is modelling systems in the world not replicating how they think' [48]. This point is further emphasised in Ref. [66], where the author argued that domain knowledge should be obtained in a manner that is separate from procedural (task-specific) knowledge. By constructing domain knowledge in a manner that is independent of the task in hand, the closer the domain-specific knowledge base will be to the truth. Furthermore, a knowledge base close to the fundamental truth is likely to be more reusable for various applications [48]. Generating reusable knowledge bases for various applications generates cost and time savings as well as increasing the ontology's reliability [67]. A highly reused knowledge base for a wide range of applications indicates its knowledge is generally accepted. These arguments support the theory that the knowledge represented in a knowledge base should reflect an objective understanding of a domain, as in an ontology. The separation of domain and procedural knowledge in an ontology facilitates knowledge sharing and reuse, a clear benefit of using ontologies [68].

An additional primary goal of developing ontologies is to allow for a common understanding of the structure of information in a domain to be shared [69, 70, 68]. One way this is achieved is by describing key concepts of the domain and specifying the relationship between the concepts, from which a vocabulary is defined [43]. These concepts are defined in a manner that is either objective or reflects the consensus among domain experts. These concepts' definitions are often referred to as a 'shared vocabulary' [71]. In generating a shared vocabulary, knowledge can be centralised, and ambiguity among the meaning of terms is removed. In turn, it aids non-expert understanding of the domain. Conceptual and terminological confusion is also reduced through domain vocabulary, making knowledge sharing and analysis easier.

Domain assumptions are made explicit in ontologies. By making these assumptions explicit, a better understanding of the domain is facilitated, particularly for new researchers [68]. This is particularly important in fields where domain knowledge is dynamic and underlying assumptions may need to be changed to adapt to the state-of-the-art understanding. Making domain assumptions explicit also enables the reuse of ontologies.

Ontologies ultimately create a shared common understanding of a domain in a machine-readable format. Because of this, they are widely used in modern knowledge-based systems. The most notable application of ontologies is in the Semantic Web [72]; this will be discussed further in

Section 2.4.

2.3 A definition of ontology

As seen in Section 2.2, ontology is a field intertwined with philosophy and AI, meaning that the concept of an ontology can be interpreted from various perspectives. As a result, there are many definitions of an ontology in the literature [73]. Some of these definitions are conflicting. In this section, these definitions will be briefly surveyed, and a definition of ontology for this thesis will be selected.

The most commonly cited definition of an ontology is 'an explicit specification of a conceptualisation' [70]. From this definition, a family of similar definitions arose; a notable variation is that of Borst, who defined an ontology as a 'formal specification of a shared conceptualisation' [74]. Studer et al., building on these definitions, defined an ontology as 'a formal, explicit specification of a shared conceptualisation' [43].

Despite being widely cited, these definitions have also been criticised. One primary criticism is that the definitions often need further definitions to describe the terms within them, making them not particularly useful for understanding [75]. In Ref. [43] the authors outlined the meaning of several terms in the definition, noting the use of the word 'explicit' refers to the fact that the types of concepts used and the constraints on their use must be explicitly defined and hence machine-readable. 'Formal' means an ontology should be machine-readable, whilst 'shared' means an ontology should not just reflect an individual's knowledge but should be accepted by a consensus of experts, i.e. an objective truth, as discussed in Section 2.2.

The use of the term 'conceptualisation' has sparked a debate of a philosophical nature. In Ref. [66] it is argued that Gruber's original interpretation of conceptualism is limited by its existential nature. To understand his interpretation of conceptualism, we can consider an ontology that represents knowledge in a particular domain. We consider what is represented in our ontology as all knowledge that exists in this domain [70]. Our ontology is therefore based on an abstract model of the word (or conceptualisation) where this is true. This conceptualisation is based on the objects hypothesised to exist in this world, and their interrelationships [76]. This viewpoint is existential as it only considers what exists as opposed to what could exist;

it depends too much on a specific state of the world [77]. A conceptualisation should not be dependent on whether or not the world is in a particular state [66]. In Studer's definition, a less existential notion of conceptualism was considered [43]. Conceptualisation is considered an abstract model of some phenomena in the world by having found the relevant concepts needed to describe that phenomena [43]. It is this interpretation of conceptualism that we will consider.

Further criticism of these definitions is discussed in Ref. [49]. Due to, in part, the complex nature of the definitions and because they are broad, there is a wide range of interpretations of exactly what an ontology is. There are examples in the literature of ontologies labelled as a wide range of information systems, including glossaries and as a set of logical constraints [49, 78, 79]. A later paper by Gruber attempted to clarify the salient points of his definition to resolve the confusion surrounding the definition [80]. Primarily, the paper makes the point that an ontology defines the concepts, relationships, and other relevant distinctions for modelling a domain [80]. Hence, this excludes information systems such as glossaries from the definition of an ontology.

Recently, Feilmayr extended Studer's definition of an ontology in an attempt to eliminate further misuse of the term: 'An ontology is a formal, explicit specification of a shared conceptualisation that is characterised by high semantic expressiveness required for increased complexity' [81]. Despite these efforts, it can still be argued these definitions are subject to interpretation and the complexity of the terms used to describe them mean that further clarification is needed [79]. Other definitions of note include those presented in Refs. [82, 83, 84].

Defining ontologies can be considered from an alternative perspective in terms of how an ontology functions, as opposed to the philosophical approaches to defining ontologies outlined above. An ontology can be considered in terms of its components, classes, attributes, relations and axioms. The base units of an ontology are called 'instances'. For example, in a rocket fuel ontology, a specific fuel such as Dinitrogen tetroxide (N_2H_2) would be an example of an instance. Classes are sets or collections of objects, and a class represents a concept. Following on from the example of a rocket fuel ontology, the type of propellant, e.g. liquid chemical, could be a class. Classes are organised typically in taxonomies and related to one another through taxonomic or non-taxonomic relations. For example, the relation 'Dinitrogen tetroxide is a liquid chemical rocket propellant' would be an example of a taxonomic relation. Axioms are defined as 'propositions or sentences that are all ways taken to be true' [85]. For instance,

'Liquid chemical rocket propellants are storable' would be an example of a general axiom.

An ontology is a formal representation of the concepts and relations of a shared conceptualisation. It can further be defined as the concepts, relations, attributes and hierarchies present in a domain. Defining an ontology in this way removes the confusion and complexity associated with the other definitions surveyed above.

For the sake of this review we will refer to an ontology as a tuple [72, 40],

$$\theta = (C, R, H^C, \text{rel}, A^\theta), \tag{2.1}$$

where C is a set of concepts (entities represented using natural language terms), $H^C \subseteq C \times C$ is a set of taxonomic relationships between the concepts showing the hierarchy of the concepts, R denotes the set of non-taxonomic relationships. The function, rel: $R \rightarrow C \times C$ relates a relationship label to the actual relationship and the set of axioms is denoted by A^θ. Other constraints on the ontology are given by axioms which are represented in some logical language. Ontology representation languages are introduced in Section 2.5.

2.4 The Semantic Web

A significant application of ontologies is the Semantic Web [72]. The Semantic Web's development has heavily influenced the evolution of the field of ontology. Therefore a brief overview of the Semantic Web will be given in this section.

The Semantic Web is an extension of the World Wide Web; it is a framework for data sharing and reuse of information in the web [86]. Ontologies are the backbone of the Semantic Web; they are machine-readable information structures which give information on the web meaning and context. For instance, web searches currently retrieve web pages using keywords entered by the user. However, these keywords can be misinterpreted, resulting in non-relevant web pages being returned. For example, a user wishing to obtain information on electromagnetic fields could enter the phrase 'field' into the Google search engine, which would return web pages related to agriculture as opposed to those on electromagnetic fields. The Semantic Web enhances the accuracy of web searches by identifying web pages that correspond to a particular concept as opposed to keywords [86].

The Semantic Web's success depends on the quick and cheap construction of domain-specific ontologies. The introduction of automatic ontology construction from text remains the key to the success of the Semantic Web [72]. This, in turn, has influenced ontology construction methods, see Section 2.7. The application of ontologies to the Semantic Web has also led to changes in ontology representation languages, which is explored in Section 2.5.

2.5 Ontology representation languages

The definitions of an ontology given in Section 2.3 make it clear that an ontology must be formal. This means it must be expressed in a language with a 'formal syntax and semantics' [87], hence allowing for machine interpretation. In the 1980s and 1990s, there was a range of knowledge-representation formats generated by distinct research groups [88]. However, these representation formats did not facilitate the exchange of data between various computer systems, i.e. they were not interoperable.

In 2001, Berners-Lee [86] published an article in Scientific American introducing the Semantic Web. The article highlighted the need for a transition from the traditional approach to knowledge-representation, whereby representations are centralised to open knowledge-based representation systems. Centralised knowledge-representation systems contain single definitions of a concept. However, language can contain multiple definitions of the same word; therefore, increasing the size of centralised knowledge-representation systems can become problematic. On the other hand, open knowledge-representation systems allow for multiple definitions of the same term. They hence can allow different systems to be merged and encourage interoperability.

As a result, the World Wide Web Consortium (W3C) developed a resource description framework (RDF). This represents the semantics of information on the web in a machine-understandable format using triples [89]. A triple consists of a subject, predicate and object [90]. The predicate of a sentence usually contains a verb and makes a statement about the subject [91]. For example, in the sentence 'The propellant was ignited on the top surface.', the subject is 'the propellant', the object is 'the top surface', and the predicate is 'was ignited on', the relevant triple is then (the propellent, was ignited on, the top surface) [92]. A set of triples is referred to as an RDF graph [89], with the subject and object represented as nodes and the predicate represented as an edge between them. Figure 2.1 depicts the example sentence, 'The propellant

was ignited on the top surface.', as an RDF graph [93]. There are three types of nodes: Internationalised Resource Identifiers (IRIs), literals and blank nodes. IRIs and literals are used to denote something in the world such as a resource. An RDF triple states that some relationship, indicated by a predicate, holds between the subject and object. Blank nodes state that relationships exist but do not specifically name them. Whilst RDF does support interoperability; it is limited to the exchange of simple binary predicates [94].

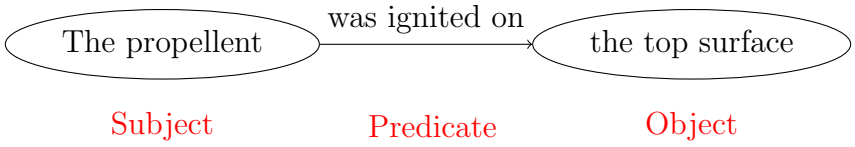


Figure 2.1: RDF graph of a triple.

A Resource Description Framework Schematic (RDFS) is a more semantically expressive version of RDF, describing resources with classes, properties and values [95]. RDF only describes binary relations between two resources (subject and object); RDFS introduces a hierarchy and inheritance for classes and properties. However, RDFS does have some limitations [96]. RDFS gives the essential elements of ontologies but is limited in its expressive power [97]. This is problematic since more expressive languages can better represent complex concepts and hence produce better ontologies [51].

Ontology Web Language (OWL) is designed for use on the Semantic Web. OWL was developed to address the limitations of RDF and RDFS. OWL has a significantly more extensive vocabulary, allowing for data to be defined in terms of set operations and for equivalences across databases to be defined [98]. For example, the RDFS triplet: (ChemicalName: Hexogen, owl: SameAs, MolecularFormula: $C_3H_6N_6O_6$) allows the chemical name to be mapped to the molecular formula. In addition, OWL allows property values to be restricted. For example, we can restrict the property of 'toxicity' to a categorical scale: (Example: Toxicity, owl:allValuesFrom, (Toxicity:1,....)). In addition to a more expressive vocabulary, OWL is more rigid, specifying not just how it can be used but how it cannot be used. OWL facilitates the machine interoperability of web content [99].

2.6 Classifying ontologies

There is a wide range of applications of ontologies. As a result, different classifications of ontologies have emerged. Guarino organised ontologies into three categories according to their purpose [100]:

- Top-level or upper ontologies describe general concepts of knowledge as a whole and relations between the ontologies that are independent across several domains. An example of such an ontology is Cyc, a long-term ontology project aiming to encode 'common-sense' knowledge [63]. Other examples are shown in Refs. [101, 102, 103].

- Domain ontologies represent concepts belonging to a particular subsection of knowledge. A domain-specific ontology will obtain domain-specific terms and their associated meanings, specialising the more generic terms in upper ontologies. The primary aim of a domain ontology is to resolve conceptual and terminological misunderstandings within the domain enabling knowledge sharing [104, 44]. Domain ontologies have been used in domains such as education [105], molecular biology [106] and space [61].

- Application ontologies describe concepts that are domain- and task-dependent. They often represent roles played by concepts in performing a specific task. For example, OntoNeuroBase is designed for sharing and reuse of tools in neuroimaging and is an example of an application ontology [107]. Other examples are shown in Refs. [108, 109, 110].

Classification of ontologies can also be performed based on the formality of the representation language used. In Ref. [111], the authors proposed the following classifications:

- Informal ontologies are taxonomies; they only contain information about hierarchical arrangements. For instance, web dictionaries or glossaries.

- Formal ontologies are built using formal ontology languages such as OWL1 and OWL2. Examples include Cyc [63].

- Semi-formal ontologies use resource description frameworks as opposed to a formal language like OWL.

2.7 Ontology construction

As illustrated in Ref. [112], ontology construction consists of 5 order-independent stages. These stages are

- Specify the domain to develop terms and concepts.
- Identify key terms, concepts and relations.
- Establish rules and axioms that describe the structure of the domain.
- Encode the ontology in a representation language, for example, RDF, RDFS or OWL.
- Evaluate the performance of the ontology.

Ontology construction can be classified into three classes, the first being manual construction, whereby an expert encodes domain knowledge into the ontology by hand. This is a more traditional approach to ontology construction but is prone to bias, as different experts can have varying opinions on the same concept. In addition, manual construction is costly both monetarily and time-wise due to the large number of tasks that require expert input [113].

Cooperative construction is when the experts supervise the construction of the ontology for the vast majority of tasks [112]. Whilst the issues associated with manual construction, such as cost, bias and time, are less significant due to the reduction in the quantity of expert-related tasks, these problems are still present.

Finally, semi-automatic ontology construction is the construction of ontologies with limited input from experts; this is also referred to as ontology learning. Ontology learning methods tend to rely on NLP techniques to identify the various components of an ontology from domain-specific texts. There has been increased interest in the development of ontology learning techniques [114], as the development of the Semantic Web has created a need for cheap, accurate ontology construction methods that can leverage multiple sources of data [72]. Fully automatic ontology construction may, however, not be possible [115, 85].

2.8 Conclusion

This chapter's purpose was to provide the reader with an overview of knowledge representation and ontologies as part of a wider investigation into the feasibility of using NLP tools to develop an expert system in the energetic-materials domain.

Section 2.1 gave an overview of knowledge representation (KR) and expert systems. Ontologies were identified as a popular KR technique widely implemented in many other scientific domains. As a result, the focus of the scope of this investigation was narrowed from expert systems to ontologies. In Section 2.2, knowledge representation with ontologies was discussed.

In Section 2.3, a brief literature review of the definition of an ontology was conducted. Ontology is an interdisciplinary field drawing influences from artificial intelligence and philosophy. There are hence a variety of different approaches to defining an ontology. It was determined that for this thesis, those definitions routed in philosophy, such as Gruber's [116], despite being heavily cited, were found to be too complex and broad. Defining an ontology in terms of its components (see Eq. (2.1)) was determined to be the most useful definition for the purposes of this thesis. Work in later chapters of this thesis will reference this definition.

In Section 2.4, the most salient application of ontologies, the Semantic Web, was introduced. The Semantic Web is an extension of the World Wide Web, which uses ontologies to give information on the web meaning and context. The Semantic Web is partly responsible for the popularity of ontologies and has consequently influenced the evolution of the field of ontology in terms of both ontology construction and ontology representation languages.

In Section 2.5, the reader was introduced to ontology representation languages. An overview of three popular ontology representation languages, RDF, RDFS and OWL, was discussed. The development of how the Semantic Web shaped the development of ontology representation languages, from primitive languages such as RDF to more expressive and rigid languages such as OWL, was also discussed. With a range of representation languages and applications for ontologies, classifications of ontologies were presented in Section 2.6. Two classifications of ontologies, on the basis of the formality of the representation language and the purpose of the ontology, were outlined in Section 2.6.

In Section 2.7, ontology construction was introduced. The 5-order independent stages used

to develop an ontology were outlined. Ontology construction methods that rely on human intervention (manual and cooperative construction) are costly, prone to bias and unable to leverage large quantities of data. On the other hand, semi-automatic ontology learning methods (ontology learning) are the construction of ontologies with minimal expert intervention. These methods tend to rely on NLP techniques to leverage vast quantities of text data to develop ontologies. The development of the Semantic Web has created a need for cheap, efficient methods of ontology construction that leverage vast quantities of data. There is hence increased interest in the development of these methods. With the broader goal of this investigation to identify the challenges of using NLP tools to develop an ontology in the energetic-materials domain, ontology learning is a research area requiring further investigation. Ontology learning is explored further in Chapter 3.

Chapter 3

Ontology construction from text

In Chapter 2, an overview of knowledge representation and ontologies was given. This overview formed the first part of a wider investigation into the challenges of using NLP tools to develop an expert system in the energetic-materials domain. The review conducted in Chapter 2 narrowed the scope of this investigation to ontologies and ontology learning was identified as an area for further research. This chapter will provide the reader with an overview of ontology learning. Domain-specific challenges of ontology construction from text in the energetic-materials domain will be identified.

Chapter achievements:

- Comprehensive overview of ontology learning from text with a focus on tools that are applicable to small datasets.
- First-of-its-kind review into ontology construction in the domain of energetic materials.
- Identification of the challenges that ontology construction in the domain of energetic materials pose.

This chapter provides the reader with a survey of the field of ontology learning. First, a broad review of the field will be given. This review will cover the dominant approach to ontology learning (the Ontology Layer Cake) (Section 3.1). The development of language models has impacted the techniques used to approach ontology learning tasks; therefore, language models will be outlined in Section 3.2. A review of the techniques used to generate each component

of the Ontology Layer Cake will be given in Section 3.1. In Section 3.3, methods of evaluating ontologies will briefly be introduced, and in Section 3.5, existing ontology learning tools will be surveyed. Discussion of the challenges of implementing ontology learning techniques/tools in the energetic-materials domain will be discussed in Section 3.6 before summarising the Chapter's conclusions in Section 3.7.

3.1 The Ontology Layer Cake

Ontology learning is concerned with developing ontology construction methods that reduce the input of an ontology engineer. Ontology learning is an interdisciplinary field drawing on techniques from the fields of Natural Language Processing (NLP), Information Retrieval (IR), and logic-based computing [115].

The dominant approach to ontology learning from text is the Ontology Layer Cake [117]. The Ontology Layer Cake views the task of developing an ontology as a collection of subtasks, each corresponding to a 'layer in the cake' [115]. This layered approach to ontology construction starts with simple tasks such as identifying relevant domain terms. It then builds upon the previous layers of the cake to complete increasingly complex tasks such as generating axioms [118]. Figure 3.1 shows a visual representation of the Ontology Layer Cake.

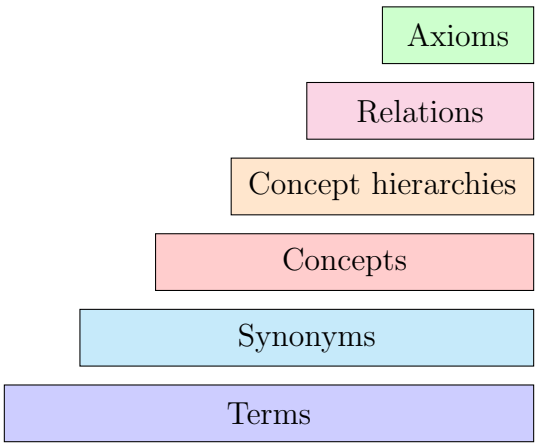


Figure 3.1: Adapted from Ref. [115]. The task of developing an ontology can be viewed as a collection of subtasks, each represented as a layer of the Ontology Layer Cake. These subtasks are completed in order starting at the bottom of the cake. Each subtask builds on those preceding it.

Extracting domain-relevant terms from the text is the first step in constructing an ontology

according to the Ontology Layer Cake. A term is a multi-word or single word that is a representation of some significant meaning and is of importance in a domain of interest [115]. Terms can be described as a linguistic realisation of a domain concept [118]. Extracting relevant terms is an essential task as the performance of all later tasks in the ontology later cake depend on accurate extraction and identification of relevant terms [85].

Following the extraction of terms, synonyms of these terms are identified and grouped [118]. The grouping of these synonyms is used to identify concepts [85].

Once concepts have been identified, the next step is to identify relations between concepts. Relating back to the definition of ontology given in Eq. (2.1), there are two types of relation: taxonomic, which is represented in Figure 3.1 as the Concept hierarchy component of the Layer Cake and non-taxonomic, referred to as Relations in Figure 3.1 [85].

Finally, axioms are found. In an ontology, axioms are used to define constraints of an ontology, and the domain assumptions [85].

3.2 Language models

The evolution of machine learning has significantly influenced approaches to the tasks that comprise the Ontology Layer Cake. One of these advances is the development of language models. Language models are AI tools that, generally speaking, model the probability of a given sequence of words occurring in a sentence [119]. These models can allow users to generate semantically meaningful numerical representations of words (word-vectors) [120]. These models tend to be trained in an unsupervised manner, meaning they learn patterns from unlabelled data.

In 2013, Google developed Word2Vec. Word2Vec is a predictive model consisting of a shallow neural network[1] that uses one of two architectures, Continuous Bag Of Words (CBOW) or Skip-gram, to generate numerical representations of words [120]. Both architectures start learning from randomly initialised vectors. The vectors are then updated depending on what was learnt from the training examples. The Skip-gram model uses an architecture with the input layer consisting of a one-hot encoded target word, a hidden layer and an output layer that predicts

[1] See Ref. [121] for an overview of artificial neural networks.

words surrounding the target word using a soft-max activation function. CBOW predicts the target words by taking an input of context words. The Skip-gram architecture tends to be preferred to CBOW as it gives the best results on semantic and syntactic tasks [120].

Global Vectors GloVe [122], is trained on global word-word co-occurrence statistics and aims to produce vector representations that minimise the reconstruction error between the model's predictions of global co-occurrence statistics and global co-occurrence statistics of the training corpus [123].

GloVe and Word2Vec learn on a word level. This means they cannot assign words not in the training vocabulary a useful vector representation. FastText solves this problem by extending CBOW to learn subword information in the form of character n-grams [124]. Using the learnt subwords, FastText can derive meaningful vector representations of out-of-vocabulary words [125].

Word2Vec and GloVe learn one single representation of a word regardless of the context in which it is applied. For example, if the algorithm was trained on a corpus that contained the following sentences: 'Darcy the dog is very energetic', 'Typical classes of energetic materials include propellants, explosives and fuels'. Then, one numerical representation of the word 'energetic' would be produced. These representation methods are therefore referred to as static representation methods.

Contextualised representations address this problem. ELMo (Embeddings from Language Models) is a contextualised language model that uses a bi-directional (meaning it learns from the preceding and proceeding word) LSTM architecture [126] to learn more context-dependent representations of words. ELMo learns at the character level, so can also develop out-of-vocabulary representations.

GTP-2 and BERT are transformer-based contextual representation models[2]. GTP-2 uses a uni-directional transformer-based architecture to learn contextual word representations [128]. BERT uses a multi-layer bi-directional transformer architecture [129]. This multi-layer architecture allows for the development of better quality contextualised embeddings. BERT is pretrained in an unsupervised manner for two tasks: (1) masked language model (a specific percentage of the tokens are replaced with a mask, and the model is then trained to predict the

[2]See Ref. [127] for an overview of transformers.

masked token) and (2) next-sentence prediction [130]. BERT tokenises words into subwords and learns at a subword level. It is, therefore, able to generate semantically meaningful numeric representations of out-of-vocabulary words.

GTP-2, BERT and ELMO can generate semantically meaningful word representations for out-of-vocabulary words and contextual embeddings. For instance, if we take the previous example, the word 'energetic' would have different numerical representations dependent on its context.

3.3 Ontology learning

This section reviews the techniques used to generate each component of the Ontology Layer Cake. The strengths and weaknesses of these methods will be stated. As mentioned previously, ontology learning is a field that draws upon techniques from a wide range of fields from NLP, data mining, machine learning (ML), IR & KR and reasoning. As a result, a wide range of techniques can be used to generate each component. Refs. [114, 85] classify these techniques into four categories: linguistic, statistical/ML, logic-based and hybrid. For simplicity, this classification will be used in this chapter.

Due to the vast array of methods available, most methods will be described only briefly; in instances where specific details of the method are relevant to the understanding of other techniques, greater detail will be provided. This review concentrates on statistical/linguistic approaches as these approaches do not require large quantities of data.

3.3.1 Concept extraction

Concept extraction consists of the first three subtasks or 'layers' of the Ontology Layer Cake (see Figure 3.1). Extracting concepts from text consists of generating terms, finding synonyms and concept formation. There are a variety of approaches to these subtasks. Here, some of the methods used will be outlined.

Generating Terms

The generation of terms can be divided into two main subtasks: term discovery and recognition. Term discovery, performed before term recognition, primarily consists of collecting a series of candidate terms. The purpose of term recognition is to determine whether or not a specific discovered term has a greater semantic relationship between a group of terms said to represent the domain than with the English language as a whole [131]. Term recognition determines if a term is part of a specific domain. Term recognition is not a straightforward task and is particularly difficult in the case of homographs, words which are spelt the same but have different meanings [132].

In order to select candidate terms, the text must be preprocessed so it is in a machine-understandable format. Preprocessing is the process of extracting and cleaning text data before being fed into an NLP algorithm. A brief summary of preprocessing will be given here; further details of preprocessing can be found in Chapter 4. The first preprocessing stage is to extract text from the source itself. Generally, following extraction of the text, the following preprocessing steps are completed: tokenisation of the text, removal of stopwords and Part-of-Speech (PoS) tagging. Tokenisation is the process of breaking down a string into a functional semantic unit for further processing. These units can be words, sentences or phrases [133]. Stop words are words which are not considered to have any semantic meaning, e.g. 'in', 'the' and 'to'. These words are considered noise and removed for the sake of computational efficiency and to not influence any frequency-based algorithms. PoS is the process of assigning a PoS label ('noun', 'verb', etc.) to word tokens. On completion of preprocessing the text, techniques to select candidate terms can be performed.

One standard linguistic method for generating candidate terms is to select nouns and nominal-word groups. These are word combinations in which the noun is the main word [134]. The theory behind this method is that terms tend to be nominal (nouns) [135]. Identifying nominal terms can be achieved using linguistic patterns such as those described in Ref. [136]. A shortfall of linguistic pattern-based approaches to term extraction is that it is time-consuming to predefine patterns for a domain [137]. Alternatively, PoS taggers can be used [115, 138]. PoS taggers often need to be trained to a specific domain [139], as different words can have different meanings in different domains. For instance, the term 'electric' can be a noun describing a system of electric wiring (and part of the physics domain), or it can refer to a feeling of excite-

ment as an adjective. Homographs can cause confusion as it is difficult to establish whether a particular term is relevant to a given domain. Training PoS taggers for a specific domain avoids this issue. One downside of this method is that verbs and adjectives can also be critical domain-specific terms.

Statistical methods for generating candidate terms can exploit the distributional properties of terms in the text. The vector-space model is a commonly used method for representing textual information in a way that allows for mathematical manipulation [140]. This can also be described as feature representation.

Traditional vector representations of words and documents can be produced from document-term matrices. A set of M documents is defined as $D = \{\ d_i \mid i \in \{1, 2, \ldots, M\}\}$, where d_i is the i^{th} document of the document set, with a vocabulary $V = \{w_j \mid j \in \{1, 2, \ldots, N\}\}$, where N is the number of words w_j in the vocabulary. An M document-term matrix $\bar{\mathsf{X}}$ is then produced from the set of documents and set of vocabulary as

$$\bar{\mathsf{X}} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \ldots & x_{1N} \\ x_{21} & x_{22} & x_{23} & \ldots & x_{2N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & x_{M3} & \ldots & x_{MN} \end{bmatrix}, \tag{3.1}$$

where x_{ij} is a metric denoting the relationship between document d_i and word w_j.

There are a range of distributional semantic metrics that can be used in the document-term matrix. The most simplistic is using boolean values,

$$x_{ij} = \begin{cases} 1, & \text{if } w_i \text{ in } d_j \\ 0, & \text{otherwise} \end{cases}. \tag{3.2}$$

Term Frequency - Inverse Document Frequency (TF-IDF) is a common metric used in NLP to measure how important a given word is to a document in a corpus [141]. The TF-IDF metric value is

$$x_{ij} = f_{ij} \log\left(\frac{M}{f_j}\right), \tag{3.3}$$

where f_{ij} is the number of occurrences of word w_j in document d_i and f_j is the number of

documents containing word w_j [142]. A higher value of x_{ij} indicates greater relevance of word w_j to document d_i [143].

This measure is commonly used in information retrieval [144]. However, it can be used for term extraction [137, 145]. Only terms beyond a threshold weighting, generally set as 0.9, can be used as terms [145]. The threshold weighting ensures only the most relevant terms are included. The TF-IDF gives a high score to terms that frequently occur in only a small number of documents. TF-IDF can be used not just to identify terms but to establish whether or not they are relevant to a particular domain. However, this method can be problematic when considering single-word terms.

The relevance of candidate terms can be established using metrics such as Domain Pertinence and Domain Consensus. Domain Pertinence is a measure used to compare the occurrence of a candidate term in the documents belonging to the target domain with its occurrence in some other domain corpus [146, 147]. Domain Pertinence is high if the candidate term is more frequent in the domain of interest than in the contrasting domain. One drawback of this measure is it can incorrectly identify a term that appears in the domain corpus with high frequency as a domain-specific term [148]. Domain Consensus attempts to ensure no non-domain specific words are extracted [149]. Domain Consensus may identify frequent terms as domain-specific terms [148, 147]. As a result, the approach taken in Ref. [147] combines these two metrics to obtain increased accuracy in assessing the relevance of terms to a domain. A shortfall of this method is that it is not designed for complex and multi-word terms [148].

A hybrid linguistic and statistical approach to generating terms is the C-Value/NC-value method [150]. This technique was developed to improve the extraction of multi-word terms [138]. C/NC-value method is used to generate multi-word terms from text automatically; it uses PoS taggers to generate nominal candidate terms. Candidate terms are then given two scores: C-value and NC-value. The C-value finds a group of valid terms in the corpus, whereas the NC-value considers the context of multi-word terms and tries to find longer strings which appear more frequently in the corpus [150]. In Ref. [151], the C-NC method was shown to be of high precision (90%). Further examples in the literature of this method being employed include that of Ref. [152], which cited a precision of 86%. However, a downfall of this technique is that it prioritises terms of higher length and consequently may ignore shorter terms of a higher importance [138].

The task of identifying terms often is presented as a feature-based supervised classification problem [153, 154]. A supervised learning algorithm learns from a labelled dataset. In the context of concept identification, this means important entities in the text would be labelled, e.g. the sentence 'RDX is a more energetic explosive than TNT' would be represented as the following set of labels [1,0,0,0,0,0,0,1] where '1' denotes that the word is a chemical term. The supervised classifier then learns from these labelled sentences and is applied to unlabelled data to identify other similar entities in the text [154]. Common approaches represent the text using TF-IDF or language models such as Word2Vec [155]. This approach is typically more complex than traditional classification tasks as entities can vary in length, location and context [153]. To adapt to this complexity, more complex approaches such as sequential learning classification [156], or boundary classification [157, 158] are used. These supervised approaches have one notable disadvantage - they rely on the existence of a large labelled dataset.

Synonym extraction and concept labelling

Once domain-relevant candidate terms have been identified, there are two remaining tasks: collecting and grouping synonyms to form concepts and the labelling of concepts.

One common approach to finding synonyms of terms is to use existing ontology databases, such as Wordnet, to look up different synonyms of extracted terms. This approach is commonly referred to as a dictionary look-up method [159]. Wordnet is an extensive lexical database of English words whereby nouns, verbs and adverbs are grouped into sets of synonyms, referred to as synsets [160]. The main problem with this method is that Wordnet is not domain-specific and may suggest synonyms that are not relevant. There are some domain-specific databases, such as the Unified Medical Language system [161]. However, no such resource exists for the energetic-materials domain. Homographs may also cause problems, and word-sense disambiguation algorithms may need to be used to determine the appropriate meaning of a word given its context. An additional issue with database look-up methods is that these databases lack word coverage and multi-word terms.

Most approaches for synonym extraction depend on Harris's distributional hypothesis: 'terms are similar in meaning to the extent to which they share syntactic contexts' [162].

Similarity-based clustering methods are widely used for concept extraction [163, 114]. Similarity-

based clustering can be defined as the organisation of unlabelled data into groups, called clusters, based on their similarity. Items in a cluster will be more similar than items in other groups. Clustering methods are applied to the vector-space model representation of text [140]. The similarity between items is calculated using a similarity or distance measure such as 'euclidean distance' or 'cosine' [164]. Clustering can be used to assign terms into groups for discovering concepts or constructing hierarchies (see Section 3.3.2) [165]. Clustering can hence be used as a means for identifying semantically similar words, and for labelling concepts, by finding the most centralised data point within a cluster. Clustering is an unsupervised learning approach.

In Refs. [165, 166] clustering is used for synonym discovery. Refs. [167, 168, 169] performed clustering using word vectors generated using Word2Vec for synonym identification achieving better performance than those approaches that used TF-IDF to generate word vectors. However, there are some drawbacks to clustering-based methods, namely data sparseness. There is also the problem of high dimensions, whereby a large corpus will typically require high-dimension vectors to represent the corpus [170]. In such situations, commonly used metrics used to obtain the similarity between terms, such as Euclidean distance measures, become less appropriate due to the 'curse of dimensionality'. As a result, the development of featureless metrics such as the normalised web distance have been developed [171].

A prominent statistical method for extracting synonyms is to perform co-occurrence analysis [172, 173, 114]. Co-occurrence analysis identifies words that regularly occur in the same context, assuming they are semantically similar [174]. There are a range of co-occurrence measures, including the dice coefficient, mutual information, and log-likelihood ratio [175, 176].

Information retrieval techniques for term indexing, such as latent semantic analysis and latent semantic indexing, apply dimensional analysis to reveal inherent connections between words [177]. These techniques have been used for concept extraction [114, 178, 179, 177]. Latent semantic indexing is based on Harris's distributional hypothesis; it is a vector retrieval model which models relationships between terms [180]. The algorithm is based on a term-document matrix, whereby rows represent terms and columns represent documents or sentences. The cell value of the matrix will contain a weighting of that term in the document/sentence. Examples of these weightings include frequency and TF-IDF (see Eq. (3.3)). Singular decomposition is then applied to the matrix to obtain a low-rank approximation of the matrix. This reduced

dimensional representation is referred to as feature-space representation. The reduced dimensions contain terms with a similar meaning and mitigate the problem of identifying synonyms. The reduced matrix similarity measures can be used to identify the most prominent terms in a corpus.

The development of term extraction techniques is more established than other techniques used for other parts of the Ontology Layer Cake. However, it is noted that when developing ontologies for specific domains, tools such as the aforementioned need to be adapted to the specific characteristics of that domain [87]. It is further noted that most of the techniques surveyed are statistical. As pointed out in Refs. [181, 163] statistical methods provide a probability associated with a string (series of words). There is no conceptual explanation of the result.

3.3.2 Hierarchies and relation extraction

Hierarchies and relations are the backbones of an ontology. Techniques for extracting hierarchical relations (taxonomic relations) from the text are more advanced and accurate than general relation-extraction (non-taxonomic) techniques.

Taxonomic relations

One of the earliest methods for extracting taxonomies from text is the linguistic-based approach of using lexico-syntactic patterns. Lexico-syntactic patterns are based on Hearst patterns, for instance, 'NP such as NP' or 'NP is an NP' (where NP corresponds to noun-phrase) [182]. These patterns frequently occur in English text, and the identification of a string containing such a pattern suggests the presence of a relation of interest. An advantage of this approach is that relations can be generated with little or no preencoded knowledge. In 1998, Hearst produced an algorithm for extraction of different types of lexico-syntactic patterns and validated them using Wordnet with an accuracy of 76% [182, 114]. However, generating these patterns manually is time costly. As a result, ML approaches have been used in recent years. For instance, work conducted in Ref. [183] used a logistic regression classifier on a training set of known hypernym pairs to learn dependency paths from parse trees and hence extract relationships automatically. Generally speaking, lexico-syntactic pattern-based approaches provide reasonably good precision, but low recall [115, 114, 184]. One reason for this is they require the

term and the hypernym to co-occur in the text within a fixed window; this is not always the case. A further disadvantage of this method is it can only derive taxonomies using single-word terms [185].

There are a range of clustering approaches to taxonomy extraction from a corpus. In this review, the classification of these methods is given in Ref. [164] (set-theoretic and similarity-based are used). Both approaches use vector-space representations of words or terms (see Section 3.3.1).

Similarity-based methods use similarity or distance metrics to determine whether or not they are semantically similar. Similarity-based clustering methods are briefly introduced in Section 3.3.1. Similarity metrics used for taxonomy extraction include co-occurrence [186], using similarity in a semantic lexicon [187], cosine and euclidean distance. Similarity-based approaches to clustering can be further classified into agglomerative and divisive clustering. An advantage of these methods is that a dendrogram (tree-like) structure is generated, making the algorithms easy to interpret. Agglomerative clustering builds a tree from the bottom up. The algorithms start with N clusters, where N is the total number of concepts, and group together the most similar clusters until a single cluster remains [114]. Divisive clustering, on the other hand, starts with one large cluster and divides it into smaller clusters incrementally until all items are in a separate cluster. Examples of taxonomy extraction using similarity-based clustering are given in Refs. [188, 189, 190, 164, 159]. Work conducted in Ref. [185] used a divisive clustering algorithm on word embeddings to extract taxonomies from corpora. This algorithm was found to outperform standard hierarchical clustering methods.

In contrast, set-theoretic methods such as Formal Concept Analysis (FCA) place items into clusters according to the inclusion relationship between their feature sets [164]. FCA is a widely used approach to generating concept hierarchies. This approach is based on the theory that objects are connected via their characteristics [152]. FCA generates an object (concept) - an attribute matrix is generated, allowing for analysis of the structure of the data and identification of dependants. Clusters of attributes and objects are found by identifying concepts with similar attributes. This results in the formation of a lattice representing concepts and attributes in a hierarchy. The size of this lattice can be considerable, resulting in exponential time complexity, which is a severe disadvantage of this method. An advantage of FCA over other clustering algorithms is it has a high degree of traceability since each cluster is given an intensional description [164].

Work conducted in Ref. [164] compared conceptual, divisive and agglomerative methods for learning taxonomies and found FCA to be the most effective. However, it was noted these methods were found to 'fall short of human achievement' [164].

Term subsumption is a method of deriving a term hierarchy [191]. When a concept frequently occurs in the context of another concept, a parent-child relationship can be created. However, this easy-to-implement method requires a vast corpus to be effective, which can be a severe disadvantage. A study conducted in Ref. [190] concluded hierarchical clustering methods are preferred for generating more complex taxonomies.

Supervised or semi-supervised methods of learning taxonomic relations generally consist of training a classifier to predict if two terms are a hypernym-hyponym pair. These methods require large quantities of training data consisting of hypernym-hyponym pairs [192]. These methods have also been seen to utilise word-embeddings [193, 194, 185]. In Ref. [195], a method of learning embeddings as features input into the classifier (supervised support vector machine) was used to generate taxonomic relationships. The embeddings were learnt using a dynamic weighting neural network; the embeddings represented hypernym-hyponym pairs as well as the contextual information between them. A comparison was conducted with other embedding approaches such as Word2Vec and term embeddings described in Ref. [196]. This method of constructing embeddings was found to outperform the aforementioned.

Non-taxonomic relations

Extraction of non-taxonomic relations is a significantly more challenging task than concept formation and taxonomic relation extraction. As a result, it is often neglected in the ontology construction process [197, 198, 199]. Approaches to non-taxonomic relation extraction can be classified as supervised or unsupervised.

One notable unsupervised approach to non-taxonomic relation extraction is outlined in Ref. [200]. The theory behind this approach is that non-taxonomic relations are expressed by verbs that relate to pairs of concepts, where concepts are assumed to be nouns. PoS tagging is applied to the corpus, and pairs of concepts that occur in the same sentence with verbs that relate them are identified. Candidate relationships are presented using triples: (noun, noun, verb). If either of the two nouns identified are not preexisting concepts in the ontology, then the candidate re-

lation is discarded [50]. Similar approaches have been outlined in Refs. [201, 198, 137]. These methods only focus on extracting relationships between a subject and object in a single sentence. This is a serious limitation [50]. In Ref. [50], an approach to extracting non-taxonomic relations to avoid this problem is presented. A probabilistic method is proposed and evaluated on texts in the tourism domain. Unfortunately, no comparative analysis of existing methods was given.

Similarly to that of taxonomic relation generation, supervised approaches to relation extraction are presented as a classification problem [202]. There are two primary types of supervised ML approaches applicable to taxonomic relation identification: feature and kernel-based approaches [202].

Kernel-based methods are based on kernel functions which define the inner product of two observed instances represented in some feature vector-space [203]. Kernal functions can be thought of as a similarity measure [202]. There are two types of kernel-based methods used for relation extraction: tree-based kernels and composite kernels. Tree-based kernels explore structured feature-space generated from parse trees [204] or dependency trees [205]. Tree-based kernel methods are found to outperform standard feature-based kernel methods. Composite kernel methods can integrate various tree-based/feature-based methods into one kernel [206]. It has been shown that composite kernel methods tend to outperform single syntactic tree-kernel methods [206].

Feature-based approaches rely on a training dataset of positive and negative examples of relations [202]. To train the algorithm, lexical, semantic and syntactic features are extracted from the examples and inputted into a classifier (for instance, support vector machine or maximum entropy) in the form of a vector of features. Upon completion of training, the algorithm will be able to classify the relation examples into a predefined set of relationship types. Approaches such as these are presented in Refs. [130, 207].

In Ref. [208], a feature-based approach relying on distant supervision is used for non-taxonomic relation extraction. Distant supervision is based on the concept that any sentence containing a pair of entities expressed in a preexisting database of relations may also express that relation in the same way. This is an extension of the approach outlined in Ref. [183, 208] for taxonomic relation extraction. In this case, the preexisting database used to provide weak supervision is Freebase. Freebase is a large semantic database consisting of over 166 million instances

of 7,300 relations [209]. Sentences containing relations found in Freebase have their lexical and syntactic features extracted and are used to train a logistic regression classifier [208]. This method has achieved precision of 67% [202]. This approach gives a key advantage over standard supervised approaches, as a database provides the labelling. This means the algorithm is less likely to suffer from issues of over-fitting and domain dependence [208]. However, since Freebase contains general relations only, domain-specific relations may not be detected. For extraction of domain-specific relations, a domain-specific database of taxonomic relations would have to exist; there is no such database for the energetic-materials domain. Furthermore, this approach focuses on binary relation extraction and does not identify relations across the text as a whole [202]. A similar approach that also uses an existing knowledge base is outlined in Ref. [210].

Ref. [211] proposes a methodology for extraction of complex relations in the biomedical domain. This approach first considers binary relations, and a graph is generated from the pairs of entities in the binary relations. An edge exists between two pairs of entities if a trained feature-based binary classifier decides the two entities are likely to be related. Following this, complex relations are generated by making tuples from selected maximal cliques in the graph [211][3]. An advantage of this method is it considers non-binary relation extraction.

In general, supervised approaches are cost, and time-intensive due to the need for comprehensive labelled data [202, 114]. Unfortunately, there are currently no existing labelled datasets available in the energetic-materials domain. As a result, there has been an increased interest in semi-supervised approaches to relation extraction.

Bootstrapping approaches for non-taxonomic relation extraction are an example of a semi-supervised method. They require a small number of seed instances of the relation type of interest. A bootstrapping algorithm will then extract similar relations to those given as seeds. DIPRE is the first bootstrapping approach to extracting relations [212]. Later approaches, such as Ref. [213] built on this approach. The performance of these approaches is highly dependent upon the quality of the seed relations used.

TextRunner is a relation extraction system presented in Ref. [214]. Relation extraction is presented as a classification problem. The Textrunner system consists of three components: a single pass extractor, a self-supervised classifier and synonym resolution. The single-pass

[3] A clique in a graph is a subset of vertices that every two distinct vertices are adjacent to. A maximal clique of a graph is one such that there is no extension by including one more adjacent vertex [211].

extractor tags each sentence with PoS tags and noun-phase chunks. For every pair of noun-phases subject to a series of constraints, a classifier is applied to determine whether or not to extract a relationship. This will be represented in a tuple of the form (e_i, r_j, e_k) whereby e_i, e_k are entities and r_j is the relation between them. The classier is trained using features such as PoS tags on parsed sentences automatically labelled as positive or negative tuples. The classier is hence able to determine if a given sequence of words is a relation.

3.3.3 Axioms

The performance of axiom learning tasks depends on the performance of previous tasks in the Ontology Layer Cake [114]. Although the generation of axioms is a particularly complicated task, like non-taxonomic relations, it is often neglected in the ontology construction process [115].

A common approach to axiom generation is Inductive Logic Programming (ILP). This method derives axioms from positive and negative examples of preexisting concepts and relations [112, 85]. ILP depends on expert oversight to ensure accurate axiom generation. This approach is shown in Ref. [215].

Logical inference can be used to infer axioms from existing relations [112]. For example, given the statements 'solid propellants are easy to store' and 'ammonium perchlorate composite propellant (APCP) is a solid propellant', then 'ammonium perchlorate composite propellant (APCP) is easy to store' can be inferred. As mentioned in Refs. [85, 112] there is a high possibility of introducing incorrect or conflicting relations. Therefore, only very basic relations can be generated using this method [112].

In Ref. [216], an unsupervised methodology for finding inference rules from text is proposed. First, dependency trees of a parsed corpus are generated. They then hypothesise that if two words are similar, a path on the dependency tree will link the same sets of words. A path represents a binary relationship, and inference rules are generated for a pair of similar paths.

3.4 Ontology evaluation

Ontology evaluation is performed to ensure a generated ontology represents the domain it is designed to represent accurately. This is a particularly important task in ontology learning as effective evaluation can allow future improvements to an ontology learning system to be made [217]. In addition, ontology evaluation must ensure quality and correctness [218, 219]. There are two approaches to evaluating ontology learning systems: to evaluate the ontology learning method and the resulting ontology. The more popular of the two approaches is to evaluate the resulting ontology. This is due to difficulties measuring the correctness of the learning procedure.

Ontology evaluation processes that assess the resulting ontology can further be divided into four distinct approaches as specified in Ref. [220].

- Gold standard, comparing an ontology to a gold-standard ontology, e.g. Ref. [221].
- Application-based, using a target ontology to perform a specific task and evaluating the specific results. For example, see Ref. [222].
- Data-driven or coverage analysis, comparing the ontology to a source of data such as a collection of documents in the domain. For example, see Ref. [223].
- Manual evaluation performed by an expert. For example, see Ref. [224].

Ontology evaluation approaches can also be considered in terms of the level of evaluation. Evaluating ontologies on different levels as opposed to the ontology as a whole is useful for automated evaluation [219]. The different levels include: lexical, taxonomic, non-taxonomic, syntactic, application and structure [220]. Table 3.1 summarises which approaches are suitable for which level of ontology evaluation.

The lexical level evaluates the conceptual layer of the ontology. Evaluation on this level is concerned with the quality of terms and definitions [225]. The taxonomic and non-taxonomic levels evaluate the correctness and consistency of relations in the ontology. Evaluation on the application layer assesses the performance of the ontology at a particular application that might be designed to do, for example, query answering. Evaluation on the syntactic level is concerned

Level	Evaluation Approach			
	Gold standard	Application based	Data-driven	Human
Lexical	✓	✓	✓	✓
Taxonomic	✓	✓	✓	✓
Non-taxonomic	✓	✓	✓	✓
Syntactic	✓			✓
Application		✓		✓
Structure				✓

Table 3.1: Adapted from Ref. [220]. Overview of ontology evaluation showing the layers of an ontology each method can evaluate.

with the formalism of the language used [220]. The structural level assesses the ability of the ontology to meet certain predefined design criteria [219]. Each evaluation approach will be briefly described in Sections 3.4.1 - 3.4.4.

3.4.1 Gold standard

The gold standard evaluation approach compares an ontology against a predefined benchmark or standard ontology that is assumed to be 'correct' [114, 217]. Gold-standard techniques tend to be concerned with completeness, conciseness and accuracy factors. The quality of the ontology is established by comparing it to an example. This is normally done on different levels, e.g. lexical - considering terms and concepts, or taxonomic - considering taxonomies, as illustrated in Ref. [72].

In ontology learning, the metrics precision (P), recall (R) and F-measure (F) are widely used to evaluate an ontology at a lexical level. These metrics are adapted and used in a wide range of NLP tasks. In this context these metrics are based on the following values:

- True positives (T_P) is the number of concepts found in the evaluated ontology and in the gold-standard ontology.
- True negatives (T_N) is the number of concepts found not in the gold-standard ontology and in the evaluated ontology.
- False positives (F_P) is the number of concepts found in the evaluated ontology and not in the gold-standard ontology.

- False negatives (F_N) is the number of concepts found in the gold-standard ontology and not in the evaluated ontology.

Recall

Recall, R, is the proportion of relevant concepts to those that were found in the gold standard ontology [112], defined as

$$R = \frac{T_P}{T_P + F_N}. \tag{3.4}$$

Precision

Precision, P, is the proportion of relevant concepts to those were found in the evaluated ontology [112]. It is defined as

$$P = \frac{T_P}{T_N + F_P}. \tag{3.5}$$

F-Measure

F-Measure, F, is the harmonic mean of precision and recall. It is a trade-off between the two quanities and a good measure of overall performance. It is expressed as

$$F = \frac{2PR}{P + R}. \tag{3.6}$$

P, R and F can be calculated automatically when compared with an ontology by using exact matches of strings. The problem with this method is it is unable to detect synonymy and polysemy. The efficiency and accuracy of this method are hence poor. P, R and F have values in the range of [0,1], where a greater value indicates better performance.

Other metrics for a gold-standard approach to ontology evaluation are outlined in Ref. [221]. The present work tends to focus on evaluation at the lexical level. Similar work is conducted in Ref. [220] on the evaluation at the taxonomic level.

Gold-standard methods are good at evaluating the accuracy of an ontology. They can also determine whether an ontology has sufficient coverage of a domain of interest. However, a limitation of this method is that the gold standard ontology itself needs to be evaluated to determine if it is, in fact, the gold standard. If the evaluation of the gold standard ontology is not done correctly, it can be difficult to determine if the source of the error is in the ontology

being evaluated or the gold standard ontology [218]. The accuracy of the evaluation method is determined by the accuracy and quality of the gold standard ontology.

3.4.2 Human evaluation

Human evaluation is when a group of humans assess how well an ontology meets a set of predefined requirements, this is done by asking domain experts to score candidate ontologies subject to these criteria. Examples of such criteria are given in Refs. [116, 226, 227]. The advantages of manual evaluation methods include that experts should be able to tell whether or not an ontology is a good reflection of a domain.

One limitation of this approach is it requires expert intervention to determine what is relevant and what is not. This is seen as somewhat subjective. Furthermore, manual evaluation may not be completely feasible in the case of a large-scale ontology. A further disadvantage of human methods of ontology evaluation is that they are highly expensive both in terms of money and time [114].

3.4.3 Application based

Application-based approaches to ontology evaluation evaluate the performance of an ontology concerning a specific task [220]. Examples of task-based approaches include document clustering and classification, word-sense disambiguation, information retrieval, question answering and coreference resolution. These approaches are generally the most efficient means of evaluating the adaptability of an ontology to a specific task. Moreover, they are useful for finding inconsistent concepts and evaluating the adaptability of a particular ontology to certain tasks. Examples of application-based approaches are outlined in Refs. [222, 228].

There are several disadvantages to application-based evaluation. Notably, if an ontology is not appropriate for a specific task, it is not easy to establish why this is the case. For instance, the ontology may be a small part of the application so that the impact may be indirect, and small [220], or it may be the case the ontology contains inconsistent concepts.

3.4.4 Data driven

Data-driven approaches use domain-specific knowledge resources, such as a collection of independent documents, to determine coverage of a particular domain [220]. This is similar to a gold standard evaluation in that it covers completeness, conciseness and accuracy. Data-driven approaches are outlined in Ref. [229]. One challenge with this approach is to find a good domain-specific corpus to perform the evaluation.

3.5 Ontology learning tools

In Section 3.3, an overview of the techniques used in the various stages of constructing an ontology was given. It is noted there are some preexisting off-the-shelf ontology learning tools that are capable of conducting several stages of the process in ontology construction. Therefore, this section will briefly survey existing ontology learning tools. In Section 3.5.7, a table summarising ontology learning systems and which components of the Ontology Layer Cake each system generates can be found.

3.5.1 ASIUM

Acquisition of Semantic knowledge Using ML methods (ASUIM) is one of the first ontology learning systems. ASUIM is a semi-automatic system aimed at learning semantic knowledge from text using linguistic and statistical techniques [230]. Sentence parsing and syntactic analysis are used to generate sub-categorisation frames of the following form
$<$ verb $><$ syntactic role | preposition : head noun $>$ these frames are then used for term extraction and concept formation [231]. Heuristic and conceptual clustering are then performed to extract taxonomic relations. Evaluation of the system as a whole has not been performed. However, an evaluation of the term-extraction part of the ASIUM system was performed, quoting a high precision value of 89% [85]. It is noted this system omits two layers of the Ontology Layer Cake from its framework: non-taxonomic relation identification and axiom generation.

3.5.2 TextStorm & Clouds

Together, TextStorm & Clouds make up a semi-automated ontology learning system [215]. The first component of the system, TextStorm, generates binary predicates from the text. Next, information from the database WordNet [232] is used to annotate the input text. Parsing and syntactic structure analysis is also performed. This information is then used to generate pairs of terms in the form of binary predicates. These predicates are either of the form verb(subject, object) or property(noun, modifier). For instance, the sentence 'The preliminary results were recorded in the lab' will generate the following predicates recorded(results, lab) and property(results, preliminary). Following this, the predictors combine the terms and relations in an existing ontology. Expert input is required most heavily here [112].

The second component of the system, Clouds, is used for axiom generation and uses an Inductive Logic Programming approach [112]. This is best explained via example, given the following list of predicates:

- eat(cat, mouse)
- is-a(cat, carnivore)

The following axiom might be generated: eat(cat, mouse) :- is-a(cat, carnivore)[85]. Of the systems evaluated, the TextStorm & Cloud system are the only ones that generates all components of the Ontology Layer Cake; however, they suffer from low accuracy. For example, the binary predicate extraction part of the system was found to have an average accuracy of 52% over 21 articles, based off domain experts' judgments [215].

3.5.3 HASTI

HASTI is an automatic ontology learning approach. It aims to construct dynamic ontologies using only a few primitive concepts and relations. The system is designed for Persian text, so it is not as relevant as other systems and will hence be described very briefly. The system learns concepts using Lexico-syntactic patterns and semantic templates. Taxonomic relations are obtained using both heuristic clustering and semantic templates. Logical inference is used for non-taxonomic relations, and very general axiom extraction [233]. Axioms are also generated

using Inductive Logical Programming. HASTI is one of the few ontology learning systems that produce axioms. However, it is noted the axioms produced are very general. Expert input is required for validation and evaluation of the generated ontology.

3.5.4 Text2Onto

The Text2Onto tool is an updated version of the tool TexttoOnto [87]. The main structure of the Text2Onto framework is centred around the Probabilistic Ontology Model (POM). This is a container for learning objects from various ontology learning algorithms. Each learnt structure is assigned a calculated probability, which can be used to decide if the object should be included in the ontology. The main advantage of such an approach is that, because learnt structures are represented at a meta-level using modelling primitives as opposed to a concrete knowledge representation language, they are in a representation that can be translated to an ontology language. This is unlike ASIUM [230], which depends on specific ontology models, often limiting them to one ontology representation language and making them difficult to transfer into other formalisms [159].

Furthermore, previous systems are not dynamic; they need to recreate the ontology if the existing corpus is changed. This is a substantial problem in the creation of the semantic web as web pages are constantly modified and changed. However, this issue is overcome in Text2Onto as it tracks document changes and calculates the corresponding probabilities needed for the POM before making the changes accordingly.

A brief overview of the algorithms fed into the POM will now be given. A series of different metrics are used to find the importance of terms extracted from the text. These include TF-IDF, entropy and C-value/NC-value method. These are normalised in a [0,1] interval and used as a corresponding probability in the POM. It is noted that Text2Onto only extracts single-word terms; as mentioned previously, this is a limitation.

Taxonomic relations between terms are obtained using the approach outlined in Ref. [234]. Multiple sources of evidence are used. In order to establish an optimal combination of multiple sources of evidence, the sources of evidence are represented as first-order features, and standard classifiers are trained. Different features were derived from WordNet, exploiting its hypernym structure, and Hearst-style patterns were used [234].

For deriving non-taxonomic relations, JAPE (Java Annotation Patterns Engine) rules were used. JAPE is a pattern-matching language consisting of a series of patterns. More details of the approach are specified in Ref. [234]. To ensure instances or entities in the corpus are assigned to the correct concepts, a vector-based similarity approach, Skewed Divergence, is used [235]. Instance and concept vectors are obtained from the corpus, and instance vectors closest to the concept vectors are assigned to such concepts.

Equivalence between terms and concepts was obtained by considering the extent to which two terms or concepts share similar contexts. Similarities were calculated based on contextual similarity using a variety of features, from simple word windows to linguistic features found using a shallow parser. This similarity is then said to represent the probability of equivalence of the terms of the concept.

It is noted this approach does not generate axioms. Furthermore, whilst the approach can aid an ontology engineer in designing an ontology, further expert input is needed to validate and evaluate the generated ontology. Finally, the described approach is not entirely automatic. The authors of the paper only quote the low recall and precision (17.38% precision and 29.95% recall on tourism-related texts) for only taxonomic relations [234].

3.5.5 CRCTOL

Concept-relation-concept tuple-based ontology learning (CRCTOL) was generated for mining ontologies automatically from domain-specific documents [236]. The first component in the CRCTOL architecture is the data importer. This converts various document formats to plain text [85]. In this process, structural information is discarded. After this, PoS tagging and parsing is performed.

Extraction of terms is performed by considering nouns and noun-phases identified using the PoS tags. A manually expert-generated domain lexicon is then used to identify those terms that are relevant to a specific domain. The domain relevance measure (see Section 3.5) is then used to identify the most relevant domain terms, forming an initial list of domain concepts. Word sense disambiguation is performed using a variant of the LESK algorithm [237]. This identifies the meaning of the term in the target domain. relation extraction is performed between single and multi-word terms. Taxonomic relations are extracted using Lexico-syntactic patterns.

Non-taxonomic relations are extracted by extracting tuples of the form (noun1, verb, noun2). Verbs, in this instance, are considered lexical realisations of the semantic relations between the concepts.

This system not only needs user intervention for developing a domain-specific lexicon but also for the validation and evaluation of the system. Furthermore, the system can only be represented in RDFS or OWL and only observes general concepts. The CRCTOL system was compared against an earlier version of Text2Onto, TexttoOnto, for term and relation extraction. The term extraction obtained an F-score of 99.5%. In addition, F-scores of 90.3% and 68.3% were obtained for relation extraction for simple and complex sentences, respectively [236]. It is noted these scores are higher than those obtained by the Text2Onto system [159].

3.5.6 OntoGain

OntoGain is a system for the unsupervised acquisition of ontologies from text [238]. The first step in this system is to perform preprocessing using PoS tagging and shallow parsing. Then, to extract terms, the C/NC-Value technique are used to generate relevant concepts. Next, hierarchical clustering and Formal Concept Analysis (FCA) are used to generate taxonomic relations. Finally, association-rule mining is used to generate non-taxonomic relations.

In Ref. [238] the OntoGain system was compared to Text2Onto in medicine and computer science domains. The OntoGain system achieved precision values in the range of 86.7–89.7%. In the comparison given in Ref. [238], there are no numerical values for the performance of Text2Onto. Evaluation of each component of the system was performed. FCA recorded low precision values between 44.2–47.1%. However, agglomerative clustering obtained values of 71.2–71.3%. Non-taxonomic relation extraction using association-rule mining achieved precision values between 71.8–72.9%. One advantage of the OntoGain system is it considers multi-word terms, whereas systems such as Text2Onto do not.

3.5.7 Comparison

An overview of various ontology learning systems has been given. A summary of the capabilities of each system with regards to which component of the Ontology Layer Cake they produce is

given in Table 3.1.

System	Component of Ontology Layer Cake				
	Terms	Concepts	Taxonomic relations	Non-taxonomic relations	Axioms
ASIUM	✓	✓	✓		
Text-Storm & Clouds	✓		✓	✓	✓
HASTI	✓	✓	✓	✓	✓
Text2Onto	✓	✓	✓	✓	
CRCTOL	✓	✓	✓	✓	
OntoGain	✓	✓	✓	✓	

Table 3.2: Summary of ontology learning systems and which components of the Ontology Layer Cake each system generates. If a particular system generates a part of the Ontology Layer Cake, it is marked with a '✓' in the corresponding cell of the table.

As mentioned throughout this chapter, the generation of non-taxonomic relations and axioms are complex tasks and are often neglected. This is reflected in Table 3.2: only two systems produce axioms, Text-Storm & Clouds and HASTI. The axioms generated by these systems are basic. Axiom generation is an evident weakness of these systems.

In Table 3.2, HASTI is the only system that produces all components of the Ontology Layer Cake. Hence, according to our definition in Chapter 2, it is the only system that produces a complete ontology. However, HASTI requires expert insight not only for validation and evaluation of the ontology but also for inputting a few initial concepts. It is hence not fully automatic. The same can be said for the other systems described in this section.

The majority of ontology learning systems are not dynamic; if new knowledge of the domain needs to be added to the ontology, then the ontology often needs to be rewritten from scratch. Text2Onto's framework allows for the ontology learning process to be dynamic, thereby negating this problem. Another advantage of the Text2Onto system is it facilitates transfer to any ontology language. This flexibility gives it an advantage over other systems such as ASIUM, TextStorm and Clouds.

From reviewing the literature, it is noted there is no commonly adopted measure for evaluating ontologies. Various approaches are used to develop ontologies, often focusing on individual components in ontology development processes. Only the OntoGain system was evaluated as a whole. Different domains and datasets are also used to evaluate ontology learning systems. The factors outlined in this paragraph make comparing these systems very difficult.

Furthermore, all of the systems described in Table 3.2 require human intervention. They are not yet fully-automated methods for generating ontologies from scratch. The evaluation methods and ontology validation require human input for the described systems in this section.

To conclude, popular ontology learning systems were surveyed. It was determined none of these systems are fully automatic. In addition, there is no commonly established method for evaluating ontologies, which makes comparing these systems difficult.

3.6 Applying ontology learning to the energetic-materials domain

Thus far in this chapter, techniques used to generate each stage of the Ontology Layer Cake and preexisting ontology learning tools have been discussed. In this section, the application of these tools and techniques to the domain of energetic materials will be discussed.

As mentioned in Chapter 1, there has been limited work on applying NLP techniques to the domain of energetic materials. Ref. [239] describes an approach to extracting chemical insights from corpora in the energetic domain. This unsupervised approach used Word2Vec embeddings to extract chemical-chemical and application-chemical relations from large corpora using cosine similarity metrics and clustering. Preliminary results presented in Ref. [239] indicate this approach can capture conceptually meaningful relationships between essential terms in the text.

A recent PhD thesis [60] introduced a system for ontology population from scientific articles, Hybrid Ontology-Learning Material Engineering System (HOLMES). The framework is designed with the pharmaceutical domain in mind. However, insights from this approach can be applied to energetic materials. The objectives of the HOLMES system include identifying ontological entities, and the relationships between them, and relating them to existing ontological classes and properties. Each of these tasks is considered a separate ML task. This system is unique as it incorporates information extraction methods from equations and diagrams. Formula extraction and graphical-imaging processing are also used to extract information from equations and images. These techniques will be briefly outlined.

The HOLMES system uses the following methodology for understanding equations. First, mathematical equations are identified in the corpus. Then, using optical character recognition algorithms, the individual characters in the equation are identified; this approach is taken as, in some PDF files, formulae may be represented as images. The algorithm used in this approach is described in Ref. [240]. The 2D layout of the equation is then extracted as a graph. The weights on the edges of the graph correspond to the likelihood the classes of those graphical symbols would occur in a spatial relationship. The meaning of the equation is found by searching for the maximum weight tree of this graph via dynamic programming. Variables in the equation are then cross-referenced with surrounding text to find their meaning. Finally, the formula is stored in a format that captures the variable definitions, the relevance of the equation, and possible assumptions [60].

To process chemical formulae, HOLMES uses the 2009 Optical Structure Recognition Application, OSRA [241]. This analyses the chemical structure and generates an output of a Simplified Molecular Input Line Entry System (SMILES) representation of the structure (essentially a single string representation) [242]. The SMILES representation is machine understandable and facilities further analysis.

Another issue highlighted in Ref. [60] is that scientific publications often convey a large amount of conceptual meaning per word. Domain-specific terms can be very complex, whereby parts of the term can be classified differently from the term itself. For example, the term ‘magnetic confinement fusion’ contains the words ‘magnetic’ and ‘fusion’ which are terms relevant to energetic materials. These terms can be referred to as nested terms and often need multiple labels in identifying and categorising terms in multi-domains. Multi-label classification techniques presented in Ref. [60] require further work. The authors suggested implementing a rule-based approach.

The HOLMES system is a prototype method. Critical weaknesses in the system include concept detection identification, which relied on entity recognition techniques [243]. In particular, it was found that concepts relating to processes and objects were difficult to extract. However, chemical and mathematical formulae extraction methods are used to provide insight into how to tackle ontology learning in the energetic-materials domain.

Analysis of text sources in the energetic-materials domain was conducted, and, using the above-surveyed literature, potential issues with transferring ontology learning techniques to the do-

main of energetic materials were identified. These issues include:

- *Equations* in the physical sciences domain contain valuable information. This information is not always found in the surrounding text.
- *Acronyms and abbreviations* are frequently used in energetic materials literature. Unfortunately, there are not always specific rules for expanding acronyms and abbreviations; as a result, they can have dual meanings. This could impact the term identification phase of the Ontology Layer Cake. As a result, some means of identifying and mapping term variants to each other are needed.
- *Greek symbols/math symbols* can have different meanings in different contexts. For instance, μ can be used to represent 10^{-6} when describing units of measurement, or it can be used to represent the coefficient of friction. This is entirely context-dependent. The meaning of a symbol in one paper can be different from the meaning of the same symbol in a different paper.
- *Diagrams, figures and graphs* contain valuable information. Arguably a good caption will summarise the necessary information in a figure. Hence, it should be ensured that figure captions are included in the extracted body of text.
- *Chemical symbols and equations* contain important scientific information. Often the information in these equations is not later explained in the text. Chemical equations and symbols should hence be made machine-readable.

It is observed that in scientific literature, chemical and mathematical formulae are expressed as figures instead of the easier-to-process ASCII representation. This further complicates the issue of chemical-formula and equation extraction.

One obvious limitation of applying the discussed techniques to the domain of energetic materials is the absence of domain-specific resources. A not insignificant proportion of the techniques discussed in Section 3.3 were supervised techniques which rely on domain-specific datasets. This success in utilising such approaches is best illustrated in the biomedical field, where ontology for knowledge representation is a well-established practice. A wide variety of domain-specific resources exist in this domain, for example, the GENIA corpus [244], a human-annotated

corpus of 670 biomedical journal abstracts taken from the MEDLINE database. As illustrated in Ref. [163], resources can be used to train algorithms for specific tasks in the ontology learning process. A specific example is given in Ref. [245] where term recognition is enhanced by identifying acronyms of specific biological terms. In the absence of such resources in the energetic-materials domain, ontology learning approaches are limited to unsupervised/linguistic/statistical approaches.

Another fundamental barrier to the successful implementation of an ontology learning system in the energetic-materials domain comes from the preprocessing of text resources. Energetic materials are an academic domain. As a result, the majority of the text resources on the subject come from journal articles and books. Extracting the relevant text from journal articles and books is not a simple task [246]. Whilst there is a growing trend among publishers to release HTML formats of journals, some publications do not yet do this producing the harder-to-extract text from PDFs. The presence of chemical names, figures and formulae can also complicate other preprocessing tasks such as tokenisation [247]. Therefore any ontology learning system in the energetic-materials domain will more than likely require a custom preprocessing pipeline that addresses some of these challenges.

3.7 Conclusion

This chapter provided an overview of the field of ontology learning, and challenges with ontology learning in the energetic-materials domain were identified. In Section 3.1, an overview of the dominant approach to ontology learning, the Ontology Layer Cake, was given. In Section 3.2, a significant development in NLP, language models, was introduced to the reader.

A review of the techniques used to generate each component of the Ontology Layer Cake was given in Section 3.3. These techniques can be statistical/machine-learning, logical, linguistic or hybrid approaches. However, the majority of surveyed techniques were statistical/linguistic due to the scope of this thesis being small-data techniques.

Section 3.3.1 evaluated the current state-of-the-art techniques in concept extraction. Correct identification of terms is a critical task as later tasks in constructing the ontology depend on it. Dictionary look-up methods were found not to be suitable for concept identification in specific

domains without the existence of domain-specific resources. Since no such resources exist for energetic materials, the use of these techniques has been excluded from any implementation in future plans. Statistical measures such as hierarchical clustering, latent semantic analysis and the N/NC method were found to perform better than the aforementioned linguistic techniques. The N/NC method, in particular, is a method that has demonstrated high precision and should hence be considered when developing an ontology learning framework.

In Section 3.3.2, techniques for taxonomic and non-taxonomic relation extraction are considered. One of the earliest methods of learning taxonomic relations from text is a linguistic technique utilising Heart-patterns. This method is of good precision but low recall and can only consider taxonomies that use terms consisting of single words. Supervised methods and term subsumption were discussed. Clustering for taxonomic relation derivation is a particularly promising area of research. Formal Concept Analysis (FCA) is a set-theoretic clustering approach found to be more effective than traditional clustering-based methods at deriving concept hierarchies. Recently the accuracy and precision of clustering methods have improved due to the use of word embeddings. Using word embeddings for clustering algorithms for taxonomic relation discovery should be considered an area worthy of further investigation.

Non-taxonomic relation discovery from text is a significantly more complex task than the aforementioned. Supervised and unsupervised approaches were described. Supervised approaches require the existence of domain-specific labelled datasets. There are currently no such resources for the domain of energetic materials; developing such a resource would be timeconsuming.

In Section 3.3.3, axiom generation was discussed. From reviewing the literature, it is clear axiom generation is a weakness in ontology learning. This is reflected in Section 3.5, which reviews popular ontology learning systems. Axiom generation was often a neglected step in these systems. Discussed techniques were only able to generate very basic axioms.

In Section 3.4, the four approaches to ontology evaluation were introduced; gold standard, application-based, data-driven and manual evaluation. A brief comparison was conducted; human evaluation, despite being time and labour-intensive, was the only identified evaluation metric capable of evaluating all levels in the Ontology Layer Cake.

In Section 3.5, a range of ontology learning systems were evaluated. This analysis highlighted that none of these systems are fully automatic, with the majority of systems requiring expert

insight for ontology verification and validation. In addition, there is no commonly established method for evaluating ontologies, which makes comparing these systems difficult.

Section 3.6 discussed the potential challenges of applying ontology learning to the energetic-materials domain. Specific challenges were discussed, such as extracting information from equations and figures. A recently developed ontology population from scientific articles, Hybrid Ontology Learning Material Engineering System (HOLMES) was introduced. Insights from this work, such as methods to extract information from formulae and figures directly applicable to the identified challenges with ontology learning in the energetic-materials domain, were discussed.

In addition, the most fundamental barrier to the successful implementation of an ontology learning system in the energetic-materials domain was preprocessing and extraction of text from text sources. Therefore, developing such a preprocessing pipeline for the energetic-materials domain will be explored further in Chapter 7.

Chapter 4

Comparing NLP tools' and experts' ability to summarise technical text

In Chapter 1, it was identified that in defence and security-related fields, NLP tools have been used to supplement human analysis instead of replacing human analysis. Therefore, examination of the boundary between using NLP tools to supplement or replace human analysis in the energetic-materials domain was identified as a key aim of this thesis. This chapter presents a preliminary study examining this aim. This preliminary study compares the ability of NLP tools in data-limited environments against domain experts to summarise technical information contained in a single document in the energetic-materials domain according to an inherently subjective concept, 'importance'. The desired outcome of this study is to identify a series of recommendations to guide future studies.

There has only been one prior study of this nature conducted in defence/security domains—the study conducted in Ref. [6] evaluated individual and intra-group human understanding of 6 volunteer participants of a short text in the domain of terrorism hostage-taking and compared this to the views of the author of the text and output generated by off-the-shelf NLP tools. The work in this chapter can be viewed as a follow-on study from Ref. [6]. One of the authors of this work acted in a supervisory role for the present study. Influences from Ref. [6] will be acknowledged throughout.

Chapter achievements:

- A review of existing summarisation techniques for the suitability of summarising a single document in a niche domain (energetic materials).
- Adapted methodology for comparing the ability of NLP tools, an expert and a group of technically trained readers to evaluate the importance of technical knowledge in a single document.
- Clear set of guidelines to inform future studies of this nature.

In Section 4.1, a literature review of text-summarisation techniques is given with the aim of identifying a series of suitable techniques for summarising a single document from the energetic-materials domain. In Section 4.2, an overview of the study methodology is given. In Section 4.3, results and their discussion are presented. In Section 4.4, conclusions and recommendations for future work are summarised.

4.1 A literature review of NLP-summarisation techniques

Automatic text summarisation aims to produce concise summaries of a document or collection of documents that convey the most important information in the text, akin to a human-generated summary [248]. An automatic text-summarisation pipeline typically consists of the following steps: preprocessing, a summarisation algorithm and evaluation of algorithm performance.

This section presents a literature review of the techniques used in all steps of an automatic single-document summarisation pipeline. This review aims to identify a series of suitable techniques for summarising a single document from the energetic-materials domain.

4.1.1 Preprocessing

In this section, an overview of preprocessing techniques will be given.

Briefly introduced in Section 3.3.1, preprocessing is the process of extracting and cleaning text data before being fed into an NLP algorithm[1].

[1]In this study, the document of interest is a single Microsoft Word document; therefore, techniques for extracting text will not be discussed in this review as it is easiest to convert the Word document to a text file.

Tokenisation is the process of breaking down a string into a functional semantic unit for further processing. These units can be words or sentences and are called tokens. Most programming languages have prebuilt tokenisers, e.g. Python's NLTK package contains a tokeniser [133].

Stopwords are words which are not considered to have any semantic meaning, e.g. 'in', 'the' and 'to'. These words are removed for the sake of computational efficiency and to not influence any frequency-based algorithms. There are various preexisting stopword lists, e.g. Python's NLTK package contains stopword lists for many languages, including English.

Part-of-Speech (PoS) tagging and parsing are two processes of obtaining syntactic information about a text. PoS tagging is the process of assigning a Part-of-Speech label (e.g. 'noun', 'verb') to each word in a sentence [249]. Labels can also include more specific tags, e.g. referring to tense. Parsing takes the PoS tag and places it into a parsing/syntactic tree to illustrate how the sentence joins together, see Ref. [250] for examples.

Stemming and lemmatising are a form of fundamental morphological analysis used to identify morphological variants of the words in the text and perform data reduction for the sake of computational efficiency.

Stemming is the process of matching various morphological forms of words to their stem and, therefore, to a root word (a word with no prefix or suffix). A root word is the smallest lexical unit of a word that contains all semantic content and cannot be broken down any further [251]. Stemming produces a stem as an output, a part of a word that does not necessarily have a meaning that can be used to form words through processes such as compounding or attaching affixes [251].

Lemmatizing (a sibling process of stemming) reduces words from their inflexion forms to a root word. Although unlike stemming, lemmatising produces a lemma, this is a linguistic compound that, by convention, is used to represent the lemme (a collection of words with the same meaning) [252]. Furthermore, unlike stemming algorithms, lemmatising algorithms use PoS tagging and consider the context of the word. As a result, lemmatising algorithms are significantly more complex than stemming algorithms [252].

The majority of literature reviewed used the more computationally efficient stemming as opposed to lemmatising as a preprocessing step in automatic summarisation algorithms[253, 254, 255]. Three main stemming algorithms are cited in the literature: the Lovins, Paice-Husk and

Porter stemming algorithms [251]. All three of these algorithms rely on affix removal to reduce inflected words to their stem. Comparative analysis between the stemmers undertaken by Paice concluded that although the Louvins stemmer is the most efficient at data reduction, it had the highest error rate. Stemming algorithms have two types of errors: over-stemming and under-stemming [256]. Over-stemming is the process by which two words of different stems are mapped to the same root. Under-stemming is the opposite, whereby two words of the same stem are not mapped to the same root. [252]

For text summarisation, we are concerned with finding the least error-prone algorithm for stemming. The Porter algorithm is identified in the literature as having the lowest error rate; it is also the most widely cited [252]. However, the performance of stemmers are domain dependent; therefore, a comparison should be conducted to determine the optimal stemmer for the energetic-materials domain.

4.1.2 Abstractive summarisation

There are two approaches to text summarisation; abstractive and extractive. Unlike extractive methods that lift sentences from the text verbatim, abstractive methods identify the main concepts in the text and use paraphrasing to generate concise summaries, similarly to a human [257, 258]. Abstractive methods typically require a more complex analysis of the input text than extractive methods. Generally, these methods produce better summaries with fewer incidents of redundancy. However, generating a good quality abstractive summary is difficult as this requires natural language generation technology [257, 259]. Consequently, abstractive summaries often suffer from grammatical errors that can confuse the meaning of the summary and struggle to analyse out-of-vocabulary words successfully.

Furthermore, approaches to abstractive summarisation are ML-based, which means they rely on large quantities of data to train the model [260]. A large corpus/dataset is not readily available for the domain of energetic materials. As a result, abstractive techniques were ruled out for inclusion in this study[2]. Given this, the remainder of the techniques reviewed in this section will be extractive summarisation methods.

[2]It is noted that pretrained models for abstractive summarisation are readily available in Python, e.g. [261, 262], but these methods produce small, succinct summaries that were not comparable in length to those produced by the participants in the study.

4.1.3 Frequency-based single-document summarisation

Early approaches to automatic summarisation, such as Luhn's [263], are centred around the theory that the frequency of a word in a document indicates the importance of the word [263]. Nenkova [264] investigated the validity of this theory by examining whether the probability of a word being included in a human-generated summary is related to the frequency of the word in the document being summarised. A set of thirty documents, each with four human-generated summaries, were examined. Results showed that words with a high frequency are very likely to appear in human summaries; the top five frequency words were 94.66% likely to occur somewhere in the set of the four human-generated summaries. However, words with low frequencies also appeared in the human-generated summaries. Whilst the results appear to support Luhn's theory, the frequency of a word is not always an accurate indicator of importance. Using frequency alone will not replicate all the content in a human-generated summary.

Nenkova developed SumBasic [264], a purely frequency-based summarisation algorithm. SumBasic uses the probability $p(w_i)$ of a word w_i occurring in a document to generate weightings $\alpha(s_j)$ of each sentence s_j. Then, $\alpha(s_j)$ is used to rank the sentences and obtain a summary. This algorithm is often used as a baseline for comparing other algorithms and is a good starting point for generating automatic summaries, it will therefore be explained in detail. The probability is given by

$$p(w_i) = \frac{c(w_i)}{N}, \tag{4.1}$$

where the number of counts of w_i is given by $c(w_i)$ and N is the total number of words in the document.

The weightings $\alpha(s_j)$ can be obtained by summing $p(w_i)$ for each word in a given sentence and taking an average,

$$\alpha(s_j) = \sum_{w_i \in s_j} \frac{p(w_i)}{|\{w_i | w_i \in s_j\}|}. \tag{4.2}$$

The sentence with the highest score, i.e. containing the most-probable words, is then picked and added to the summary. The algorithm then updates the probabilities to avoid a word appearing in the summary more than once hence mitigating repetition within the summary (a problem with earlier algorithms such as Luhn's [263]). For each word w_i in the selected

sentence, the word probability is updated from $P_{\text{old}}(w_i)$,

$$P_{\text{new}}(w_i) = P_{\text{old}}(w_i). \tag{4.3}$$

The process is repeated until a summary of a predefined size is generated [264].

Frequency-based approaches to text summarisation are simplistic and have the added advantage of being language-independent; however, they do not consider the context in which the word appears. This can lead to issues with homographs. In addition, preprocessing can significantly impact the performance of algorithms: stemming and lemmatising reduce the corpus to a selection of root words. Cases of over- and under-stemming will lead to words having a higher/lower frequency count than the 'true count', which could influence the output. Furthermore, frequency-based algorithms cannot identify when a concept is discussed using different words; this can lead to generated summaries containing redundant information.

4.1.4 Graph-based single document summmarisation

Another common subset of approaches to extractive document summarisation are graph-based approaches. Graph-based approaches structure text documents as graphs and exploit the graph's structure to extract important concepts [265]. TextRank is a popular single document automatic summarisation algorithm that employs graph-based analysis to generate a summary [266]. The TextRank algorithm can also be applied to keyword extraction, a technique useful in the generation of ontologies [267]. The algorithm is derived from Google's Page-Rank algorithm, used to rank search results for web-pages [268][3]. The TextRank algorithm represents a text document as a directed graph, $G = (V, E)$, where V is the set of vertices of the graph, where each vertex $V(s_i)$ represents a sentence s_i in the document and E is the set of edges of the graph, used to connect the vertices of the graph, $E \subset V \times V$ [266]. The edges of the graph represent the relations between the sentences. The set of vertices directed into vertex $V(s_i)$ is denoted $\text{in}(V(s_i))$.

The algorithm assigns each edge a weight λ_{ji}, reflecting the relationship (similarity) between

[3]TextRank is similar to the popular multi-document summarisation algorithm LexRank [269].

the two sentences s_i and s_j [270]. This weight is given by

$$\lambda_{ji} = \frac{\sum_{w \in i,j} C_w^i + C_w^j}{\log |s_i| + \log |s_j|}, \tag{4.4}$$

where C_w^j is the number of occurrences of the word w in the sentence j. The logarithmic length of each sentence normalises the weights to ensure long sentences are not unfairly given an advantage.

These weights are used to obtain the score of each sentence. The score α of each sentence s_i is calculated iteratively until convergence using

$$\alpha(v(s_i)) = (1-d) + d \sum_{v(s_j) \in \text{in}(v(s_i))} \frac{\lambda_{ji}}{\sum_{v(s_k) \in \text{in}(v(s_j))} w_{jk}} \lambda(v(s_j)), \tag{4.5}$$

where d is the damping factor set to the default value of 0.85 [266]. The sentences with the highest-ranking scores are used to comprise a summary [266].

This algorithm is popular since it is unsupervised and language-independent. It only depends on word occurrence and identifies document-wide connections. However, a fundamental limitation of this algorithm can be found in Eq. (4.5), which gives the weighting between two sentences as the number of common tokens in the sentences. As a result, sentences with no common words but discussing the same concepts will have a weighting of 0. This could result in a sentence having a lower weighting than expected and critical information being neglected from the summary as a result.

Recent work by Mallick et al., [271], adapted the traditional TextRank algorithm to address this issue. This approach considers a vector-space feature representation method (a numerical representation of the text) to compute sentence similarity. The text document, d, is a vector-space model comprised of sentences s_i represented as word vectors. A graph of the form $G = (V, E)$ is produced according to the procedure laid out above for TextRank. Unlike in the TextRank algorithm, words are not assumed to be of equal weighting. The modified TextRank approach uses the Term Frequency-Inverse Sentence Frequency (TF-ISF)[4] measures to find the relevance of the sentences according to the salience of the words they are composed of. This is incorporated into the similarity measure. The term frequency f_i^j is merely the count of a word

[4]This is a modification of TF-IDF introduced in Chapter 3.

w_i in a sentence s_j. The inverse sentence frequency ISF_i is the normalised number of sentences that a word w_i appears in, i.e.

$$ISF_i = \log \frac{N}{|\{s_j \in D : w_i \in s_j\}|}, \tag{4.6}$$

where N is the total number of sentences in the document and $|\{s_j \in D : w_i \in s_j\}|$ is the number of sentences that the word w_i occurs in. Given this, the TF-ISF measure is given by

$$\beta_i^j = f_i^j \times ISF(w_i), \tag{4.7}$$

where β_i^j (the weight of a word w_i in sentence s_j) can then be used to calculate the similarity between two sentences s_j and s_k, i.e.

$$\lambda(jk) = \frac{\sum_{i=1}^{m} \beta_i^j \beta_i^k}{\sqrt{\sum_{i=1}^{m} (\beta_i^k)^2} \sum_{i=1}^{m} (\beta_i^j)^2} \tag{4.8}$$

where $j, k = 1, \ldots, N$ and N is the total number of sentences in the document. This similarity metric considers the relevance of the sentences with reference to the importance of the words composing it. The graph is then reduced by removing edges with weights below a certain threshold value before the standard TextRank algorithm is performed.

This method is an improvement on the standard TextRank algorithm as it uses a more effective method of computing sentence weightings that considers the importance of the words composing the sentences [272]. This method also achieved improved performance over other representation methods such as TF-IDF (see Chapter 3, Eq. (3.3)) [271]. However, the method cannot consider complex semantic relationships; it cannot identify when a concept is discussed using different words. This can lead to the generated summaries containing redundant information and important information being missed from the summaries.

Work carried in Ref. [273] used the feature representation method Word2Vec to obtain semantic similarities between sentences (see Section 3.2 for a description of Word2Vec). This process of using a pretrained word representation model from a previous task on a new somewhat-related task is referred to as feature-representation-based transfer learning [261]. Transfer learning will be discussed in detail in Chapters 8 and 9. Results obtained using this approach were an improvement on previous graph-based approaches.

A graph-based approach by Parveen et al. [274] tackled the redundancy problem discussed in the previous two methods. They represented the text document as a bipartite graph with two sets of nodes: entities and sentences. Coherence scores were calculated using the degree of each node (number of edges from each node) to ensure the summary was non-redundant and coherent optimisation was performed. Meaningful sentences were picked using a graph-ranking algorithm to establish which sentences belong in the summary. However, this algorithm did not perform as well as the TextRank algorithms in capturing important information. This may be due to the very basic method of capturing semantic relationships.

4.1.5 Latent Semantic Analysis

Latent Semantic Analysis (LSA) is an unsupervised extractive text-summarisation method [275]. LSA assumes that if two words appear in similar contexts, they have analogous semantic interpretations, and if two contexts are alike, they must have words which are semantically related [270]. Using this assumption, LSA-based methods map the semantic relationships in a document and can group related sentences even if they do not contain the same words [276]. LSA can be divided into three phases: term-document matrix creation, singular value decomposition (SVD), and sentence selection [265].

Term-document matrix creation transforms a document, D, to a matrix. The sentences in a document are represented as a set, $S = \{s_1, s_2, \ldots\ldots\ldots, s_N\}$ and $W = \{w_1, w_2, \ldots\ldots\ldots, w_M\}$ is the set of M unique words in D. Given this representation, D can be represented as the matrix

$$\mathsf{A} = \begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_M \end{array} \begin{array}{c} \begin{array}{cccc} s_1 & s_2 & \ldots & s_N \end{array} \\ \begin{pmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,N} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{M,1} & a_{M,2} & \ldots & a_{M,N} \end{pmatrix} \end{array}, \tag{4.9}$$

which has dimensions $N \times M$. The entries a_{ij} of the matrix represent the importance of a word in a given sentence. There are a variety of different methods to calculate this, e.g. cosine similarity [277], log entropy [278], frequency or TF-ISF [279]. A comparison of different weighting functions can be found in Ref. [280].

After the creation of A, SVD is performed to model the relationships between the words and sentences. This is the process of decomposing A into three different matrices using

$$\mathsf{A} = \mathsf{U}\Sigma\mathsf{V}^{\mathrm{T}}, \tag{4.10}$$

where U is a unitary matrix of left singular vectors with dimensions $M \times N$ and V^{T} is is the transpose of the unitary matrix V of right-singular vectors, of dimensions $N \times N$ [279, 281], and Σ is a diagonal $N \times N$ matrix with diagonal elements known as singular values. These singular values are sorted in descending order.

It is of note that the preprocessing techniques can significantly impact the resulting matrices produced by SVD. For instance, stopword removal, stemming and lemmatising can significantly reduce the dimensions of A [279].

The dimensions of the outputted matrices can be reduced by deleting the smallest singular values from Σ, which eliminates the least important values from the calculation [282]. Determining the optimal number of dimensions is a problematic issue in SVD. The lower the number of dimensions, the broader the comparison of the concepts. Finding the optimal number of dimensions for LSA will increase computational efficiency and reduce noise or terms in the documents that do not contribute to the meaning [177]. However, there is no one-size-fits-all solution to this task. This process is often task- and content-dependent.

Assume the reduced dimensions of the matrices are U $(m \times r)$, Σ $(r \times r)$ and V^{T} $(r \times n)$. This process of reducing the dimensions can be equated to a mapping between the m-dimensional space of weighted-frequency term vectors and the r-dimensional singular vector space.

SVD decomposes the document into r linearly-independent base vectors; each of these base vectors represents a salient concept. The concepts are found by semantically clustering words and sentences by identifying important and recurring patterns in the text. If a word combination pattern is important and recurs, it will be represented by a left singular vector in matrix U. The magnitude of the corresponding singular value in matrix Σ represents the degree of importance of the concept.

A sentence containing a particular word combination pattern will be projected along the corresponding singular vector. The sentence with the best representation of this word combination

pattern will have the largest value of overlap with the vector. The cell values of the matrix V^T show the overlap between the sentences and the concept; the rows indicate the most important concepts, and the columns represent the sentences [283].

The final step of LSA selects the sentences for inclusion in the summary from the results of SVD. There are various approaches to doing this [284].

Gong et al. [285] used V^T to select the most important sentences for inclusion in the summary. By identifying the most important concepts, the sentence that best represents this concept is selected. This involves going down the rows of the V^T matrix; the sentence with the highest cell value for that row is identified and included in the summary. The column is then removed from the matrix, and the process is repeated on an iterative basis until a predefined number of sentences are collected. An overarching issue with this method is it assumes all concepts are of the same importance. This could lead to some of the selected concepts not being significant. Furthermore, this method assumes there is one important sentence per concept; it is possible not all of the important information on a concept can be represented in one sentence. This is particularly relevant when considering technical documents such as the one in this study.

Steinberger and Jezek [286] resolved this problem by finding the sentences with the highest combined weight across the important topics. Then, the length of each sentence vector in $\Sigma \cdot V$ was calculated, and the sentences with the greatest length were selected [286]. This is among the more popular approaches in LSA [284]. The aim of doing this is to find the sentences with the greatest combined weight across all important topics [286]. A shortcoming of this method is it relies on traditional weighting schemes to build the document representation.

Other approaches to this method include the topic approach by Ozsoy et al. [279]. This approach aims to classify the topics extracted from V^T as 'topics' or 'subtopics'. First, the main topics are identified by constructing a concept matrix. The strength values of each concept are used to determine the document's main topics. Next, the sentence with the highest score is chosen for each selected main topic. This selection is performed until a predefined number of sentences are collected.

Another approach by Ozsoy et al. [279] builds on Steinberger and Jezek's [286] approach of using a preprocessed V^T for sentence selection. This method is known as the 'cross approach'. V^T is preprocessed by setting all values in a row to zero if they are below a threshold value.

This removes any sentences that are not core sentences representing the topic as opposed to sentences that are related in some way. The length scores of the sentences were found, and the highest scoring sentences were used in the summary. This was found to outperform the other sentence-selection methods mentioned above.

Yeh et al. [282] used an approach referred to as LSA + TRM to document summarisation. This approach used AutoTag [287] to obtain a list of keywords in the sentences, any keywords that were not nouns were removed. The standard LSA approach outlined above was then conducted with the keywords being represented as w_i in the input matrix (Eq. (4.9). The entries of the input matrix a_{ij} were obtained using

$$a_{ij} = G_i L_{ij}, \tag{4.11}$$

where L_{ij} is the local weight of w_i in s_j and G_i is the global weight of w_i in D, demonstrating the saliency of w_i in the document.

L_{ij} is defined as $L_{ij} = \log(1 + c_{ij}/n_j)$ and $G_i = 1 - E_i$, where c_{ij} is the frequency of W_i occurring in S_j, n_j is the number of words in S_j and E_i is the normalised entropy of W_i which is defined as

$$E_i = -\frac{1}{\log(N)} \sum_{j=1}^{N} f_{ij} \log(f_{ij}), \tag{4.12}$$

where f_{ij} is defined in Eq. (3.3). It is noted this method of obtaining the coefficients a_{ij} is designed for multi-document summarisation and has been adapted by setting the number of documents to one. After generating the input matrix A, SVD is performed, and an alternative approach is used to extract the sentences for the summaries. A relationship map based on the document's semantic representation is then constructed; from this, important sentences are identified as those with a high number of connections [282]. Whilst using this method provides a visualisation of the summarisation process like with other LSA approaches, finding the optimal dimension reduction ratio is difficult and computational time is an issue [282]. A shortcoming of this method is it relies on traditional weighting schemes to build the document representation in the matrix A. As a result, this method suffers from some of the same limitations associated with those using cosine similarity and TF-ISF.

Recently, work undertaken by Al-Sabahi [288] used word embeddings to obtain the input matrix A as opposed to the traditional methods outlined above. Word2Vec was used to learn the vector

representation of words. The model is trained to reconstruct the linguistic contexts of words. As the model is trained on such a large corpus, this method has the advantage of bringing in semantic knowledge that is not available from the document. The study used pretrained word embeddings from Google's Word2Vec, trained on the Google News corpus, and embedding-based weighting schemes to calculate the values of the input matrix. However, since the corpus the embeddings were trained on was not similar to the documents being summarised, the study ignored words that were not in the pretrained word embeddings. Despite the documents being summarised being different from the domain of the documents used to train the embeddings, this method achieved higher performance scores than standard LSA approaches. This is another example of feature representation-based transfer learning.

4.1.6 Machine-learning-based single-document summarisation

In recent years, ML-based summarisation algorithms have become more popular [289]. This reflects both large-scale advancement within the field of ML in the last decade and the need for summarising large datasets in a computationally efficient manner [290].

Supervised learning techniques use labelled training data which, for every input document, has a corresponding human-generated extractive summary, i.e. the 'correct result' [291]. Examples of supervised learning techniques that have been applied to automatic summarisation include converting the summarisation problem to a sentence-level classification problem [292, 293]. As with the classification techniques discussed in Chapter 3, this requires a large manually annotated dataset. Every sentence in the dataset is labelled as 'in' or 'out' of the summary [265]. Other methods represent extractive summarisation as an optimisation problem [294, 293]. Likewise, a large labelled dataset is required [295]. Generating such a dataset is time consuming and challenging to do without the introduction of bias: if the corresponding summary for each document is human, who is to say it is the correct or ideal summary? Training supervised algorithms on small datasets can lead to overfitting, particularly when the dataset's number of input features (dimensions) is greater than the number of training samples [296], as is often the case with text data. Overfitting is when a trained model fits against its training data exactly and therefore fails to generalise well (perform well on unseen data) [297], which can lead to over-optimism about the algorithm's performance. Semi-supervised techniques use a small amount of the same type of labelled training data and a large amount of unlabelled training data.

Unsupervised techniques use unlabelled training data, a large dataset of input documents without the corresponding human-generated summaries. Unsupervised training models deduce inferences from the data by finding hidden patterns or structures. This contrasts with supervised approaches, which give a clear labelled response [291]. Unsupervised techniques applied to automatic summarisation include clustering [298], and those based on neural networks [299, 300].

Recently there has been an increase in supervised and unsupervised techniques relying on deep learning methods [301]. A review can be found in Ref. [300].

The present study is conducted in a data-limited environment using one document in the domain of energetic materials. Since no open-source dataset of documents and their summaries exists for the energetic-materials domain, supervised and semi-supervised methods are ruled out and not discussed further. Furthermore, data-intensive unsupervised methods are also ruled out. ML techniques that do not require large datasets are therefore considered. These techniques are limited to those using pretrained models available to be deployed in Python and clustering.

Clustering-based techniques were first introduced in Chapter 3. They are an ML technique that groups data (which has not been labelled, classified, or categorised) by similarity. There are three main types of clustering: hierarchical, diverse, and partition/Bayesian clustering [302].

Agglomerative clustering is an example of a hierarchical clustering approach that has been applied to single-document extractive summarisation. Each sentence is initially represented as an individual cluster; sentences are then clustered together based on the maximum pairwise cosine similarity of vector representations of sentences. Clusters are then bound together in a 'greedy' manner[5], and, as a result, a hierarchy of clusters is generated [303]. This is performed until a threshold value of dissimilarity between all clusters is achieved. In doing this, semantically similar sentences are grouped, for instance, in Ref. [304], TF-ISF (see Eq. (4.7)) is used to generate sentence vectors before performing agglomerative clustering to generate summaries.

El-Kilany and Saleh [305] performed Louvain clustering on dependency graphs of sentences to group coherent words into clusters. Using these clusters, words were scored based on their position in relation to other words and the importance of the word in the dependency graph of the sentence in which the word appears. The highest-scoring nouns were taken to be the keywords. The scores of these keywords were increased if a keyword appeared in the context

[5]Greedy algorithms solve problems by selecting the best option at a given point without considering the effect on later results.

of another keyword. The sentences were then scored depending on the number of keywords in a sentence and the associated scores of the keywords. Finally, the top-scoring sentences are selected. This algorithm assumes the generated keywords, and hence selected sentences, represent the most important topics so it can produce non-redundant summaries. Unlike the previous clustering approach, this approach captures the dependency relationships between sentences which can be useful for establishing coherence. However, selecting one sentence per topic can lead to important information being lost, as not all the salient information on a topic is necessarily contained in one sentence.

Other non-hierarchical clustering algorithms have also been used to generate automatic summarisation algorithms. Work undertaken by Jain et al./ [306] used a K-Means clustering algorithm for query-related summary generation [306]. A K-Means clustering algorithm separates data into clusters by minimising criteria referred to as inertia [307]. Considering a dataset of N items, $X = \{x_1, ...x_N\}$ being split into a predefined number of clusters, $C_1, ...C_K$, inertia is defined as the sum of the squared Euclidean distances between each sample x_i in cluster C_j and the mean of the samples in each cluster μ_j (centroids). The K-Means clustering algorithm hence aims to find the centroids that minimise the inertia

$$E(\mu_1, ...\mu_K) = \sum_{i=1}^{N} \sum_{j=1}^{K} I(x_i \in C_j) \min(||x_i - \mu_j||^2), \tag{4.13}$$

where $I(X) = 1$ if x_i is in cluster C_j and 0 otherwise [308]. In general, clustering algorithms produce non-redundant summaries, however they do not capture the context in which words appear which can lead to the true meaning of the text not being captured.

Ref. [301] describes a feature representation-based transfer-learning technique that has been applied to extractive summarisation using K-means clustering. First sentences are embedded using a pretrained model. The embeddings are then clustered using K-means clustering, and sentences closest to the centroid of each cluster are selected for inclusion in the model. This method achieved state-of-the-art performance for clustering lecture materials. The work done in this paper lead to the development of a Python package, 'bert-extractive-summarizer'. This package allows any pretrained language representation model to be loaded into the summarisation pipeline, notably BERT and GTP2 (see Section 3.2 for an overview). This approach achieved state-of-the-art performance over other methods.

4.1.7 Evaluation metrics

Thus far, this review has covered techniques used in two stages of developing an automatic text-summarisation pipeline: preprocessing and summarisation algorithms. This section will cover the final stage of an automatic text-summarisation pipeline, evaluation of algorithm performance.

There are two approaches to summary evaluation: extrinsic and intrinsic [309]. Extrinsic approaches evaluate the performance of automatic-summarisation algorithms by looking at how the summary performs at completing tasks such as information retrieval and query answering. In contrast, intrinsic evaluation looks at the summary by comparing it to human-generated reference summaries and assessing the quality and content of the produced summary. For this study, intrinsic content-based evaluation of extractive summarisation algorithms is of interest.

For summary evaluation, the metrics can be calculated from the definitions in Chapter 3, Eqs. (3.4)-(3.6), after adapting the definitions of T_P etc. in the following way:

- True positives (T_P) are the number of sentences machine-generated summary that are also in the reference summary.
- True negatives (T_N) are the number of sentences not in the machine-generated summary that are also not in the reference summary.
- False positives (F_P) are the number of sentences in the machine-generated summary that are not in the reference summary.
- False negatives (F_N) are the number of sentences in the reference summary that are not in the machine-generated summary.

There are issues with P, R and F as summary-evaluation metrics. Humans often cannot agree on what makes an ideal summary [6]. If P and R are used alone, it is possible two equally good summaries could be judged differently. One method of correcting this is to average the precision and recall scores for a collection of reference summaries.

In order to evaluate the content on a word level, metrics like the ones in the widely established Recall-Orientated Understudy for Gisting Evaluation (ROUGE) group of metrics can be used.

The metric $R_{\text{ROUGE-N}}$ gives a measure of overlap of number of N-grams (where 1-gram is one word) between the machine-generated summary and a gold-standard summary (denoted by G). This is a value between 0 and 1, where 1 means all N-grams in the gold-standard summary are present in the generated summary. $R_{\text{ROUGE-N}}$ is defined as

$$R_{\text{ROUGE-N}} = \frac{\sum_{S \in G} \sum_{\text{gram}_n \in S} C_{\text{match}}(\text{gram}_n)}{\sum_{S \in G} \sum_{\text{gram}_n \in S} C(\text{gram}_n)}, \tag{4.14}$$

where n is the length of the N-gram being calculated (gram_n) and $C_{\text{match}}(\text{gram}_n)$ is the maximum number of N-grams co-occurring in the reference summary and the machine-generated summary and $C(\text{gram}_n)$ is the number of N-grams in the reference summary [310]. The metric $P_{\text{ROUGE-N}}$ gives a measure of the number of N-grams in the generated summary that are also present in the gold-standard summary. $P_{\text{ROUGE-N}}$ is defined as

$$P_{\text{ROUGE-N}} = \frac{\sum_{G \in S} \sum_{\text{gram}_n \in G} C_{\text{match}}(\text{gram}_n)}{\sum_{G \in S} \sum_{\text{gram}_n \in G} C(\text{gram}_n)}, \tag{4.15}$$

$P_{\text{ROUGE-N}}$ is also between 0 and 1. If the number of N-grams in the reference and gold-standard summaries are not the same then these metrics will be biased. The $F_{\text{ROUGE-N}}$ metric is the harmonic mean of $R_{\text{ROUGE-N}}$ and $P_{\text{ROUGE-N}}$. This metric removes the aforementioned bias by accounting for statements of different lengths and can be interpreted as the balanced overlap of N-grams between the two summaries. The metric $F_{\text{ROUGE-N}}$ is adapted from Eq. (3.6) and is defined as

$$F_{\text{ROUGE-N}} = \frac{2 R_{\text{ROUGE-N}} P_{\text{ROUGE-N}}}{R_{\text{ROUGE-N}} + P_{\text{ROUGE-N}}}. \tag{4.16}$$

The value of $F_{\text{ROUGE-N}}$ ranges between 0 and 1, where a value of 1 denotes the highest overlap of N-Grams between the two summaries.

4.1.8 Conclusion

In this section, a review of the techniques used to develop an automatic single-document summarisation pipeline was conducted. The objective of this review was to identify a series of suitable techniques for summarising a single document in the field of energetic materials. The findings of this review will be discussed here.

In Section 4.1.1, an overview of preprocessing techniques was given. The NLTK Python package

was identified as a useful preprocessing tool. Stemming and lemmatising were introduced. A literature review established that stemming is used more frequently in automatic-summarisation pipelines. Three popular stemming algorithms were discussed: Lovins, Paice-Husk and Porter. The Porter stemming algorithm was identified in the literature as having the best performance and being the most widely cited. However, the performance of these algorithms is domain-dependent. Therefore, an in-domain comparison of various stemming algorithms should be conducted.

There are two main approaches to automatic summarisation: abstractive and extractive. In Section 4.1.2, abstractive methods were introduced. Abstractive approaches were immediately ruled out as they are ML-based. In the absence of an extensive labelled/unlabelled dataset in the energetic-materials domain, it was determined that it is unlikely these approaches would yield high performance. All remaining approaches reviewed were extractive methods. Generally speaking, extractive methods, while faster and more straightforward, tend to produce summaries very different from one that an expert would produce [257]. In addition, these summaries can include redundant statements; there can be a lack of semantic cohesion between sentences due to unexplained co-reference relationships, and important information is often spread across sentences.

In Section 4.1.3, frequency-based methods were reviewed. SumBasic is an example of a frequency-based extractive summarisation algorithm commonly used as a baseline against which other summarisation algorithms are compared. As illustrated by Nenkova [264], frequency alone is capable of replicating all the concepts in a human-generated summary. Frequency-based approaches do not take into account the context in which words appear, nor are they able to identify any semantic relationships in the text nor capture the true meaning of the text. There are also issues with homographs and redundancy in the summaries produced. Whilst these approaches are efficient and easily understandable, more comprehensive methods that employ semantic, syntactic and morphological analysis are needed to overcome the shortcomings. It is recommended that SumBasic be included in this study as a baseline for comparison to more complex methods.

In Section 4.1.4, graph-based methods for single-document summarisation were discussed. TextRank is a graph-based extractive summarisation algorithm that would be suitable for inclusion in this study as it does not rely on a large dataset. Algorithm performance can be improved

using distributional sentence vectors such as those based on TF-ISF weightings to compute the weightings between nodes. Transfer-learning techniques, for instance, using Word2Vec embeddings to compute the weightings between nodes on a graph, have also been found to improve performance. The TextRank algorithm will be used in this study.

In Section 4.1.5, Latent Semantic Analysis (LSA) was discussed; this is an effective method for extractive summarisation. In general, LSA approaches are an improvement on graph and frequency-based approaches to solving homograph issues. However, the context in which the words occur is not fully considered. It is an unsupervised approach suitable for data-limited environments. There are various approaches to selecting sentences for inclusion in a summary. The method that achieves the best performance out of those mentioned is the 'cross method'. The cross method will be used in this study.

In Section 4.1.6, ML approaches to extractive summarisation were examined. Supervised ML approaches typically perform better than the other methods described in this review. However, supervised methods require sizeable labelled training sets; since no such dataset exists in the energetic-materials domain, these methods were not considered further. Unsupervised methods such as clustering were also reviewed. The 'bert-extractive-summarizer' is an easily implementable Python package that uses pretrained language representation models to generate summaries of documents using K-means clustering. This approach will be explored in this study.

In Section 4.1.7, an overview of evaluation metrics was given. Metrics that evaluate sentences on a sentence level: precision P, recall R, and F-score F were introduced, as were metrics that evaluate on an N-gram level (the ROUGE metrics).

To conclude, this review aimed to identify techniques for developing a summarisation pipeline that can summarise a single document in the field of energetic materials. Various suitable techniques were identified, and areas of further research were established. This is summarised below:

- An in-domain comparison of the Lovins, Paice-Husk and Porter stemming algorithms should be conducted to identify the most appropriate stemmer for the domain of energetic materials.

- The following summarisation algorithms are suitable for this study:
  - SumBasic as a baseline measure of performance.
  - TextRank.
  - LSA using the cross method for sentence selection.
  - The clustering-based 'bert-extractive-summarizer'.
- In the cases of TextRank and LSA, modifications to traditional algorithms, such as the use of TF-ISF weightings and Word2Vec, should be explored further.

4.2 Methodology

The objective of the study in this chapter is to compare the ability of NLP tools in a data-limited environment to domain experts in the summarisation of technical information contained in a single document. In Section 4.1, a literature review was conducted to identify useful techniques that can be used to develop a suitable NLP-summarisation pipeline. In this chapter, an overview of the methodology of this study will be presented, and techniques identified in the previous Chapter will be used to define a summarisation pipeline.

The proposed methodology for this study follows a similar format to the study conducted in Ref. [6]. Here, an overview of the methodology will be presented. Specific details will be given in this section's sections. An overview of the methodology is given in Figure 4.1.

The first stage of this method is to preprocess the document that is the subject of this study, henceforth referred to as 'the text'. The text is an eight-page technical report providing an overview of the effect of electric and magnetic fields on propellant combustion and was selected to be the exemplary document of this study. Further details on the text can be found in Section 4.2.1. A description of the preprocessing methodology is detailed in Section 4.2.4. A group of scientifically proficient 'readers' and an expert were asked to complete a background-information questionnaire, this is described in Section 4.2.2.[6] Collecting information on the

[6] An expert practitioner in the field of energetics could be someone who has acquired expertise through studying the field academically or by obtaining practical experience, such as through working in the armed forces or in the manufacturing of energetic materials. It is therefore possible to have an expert in energetic materials who has limited education or literacy. In this study, an expert in the field is considered someone who has studied the field in an academic sense.

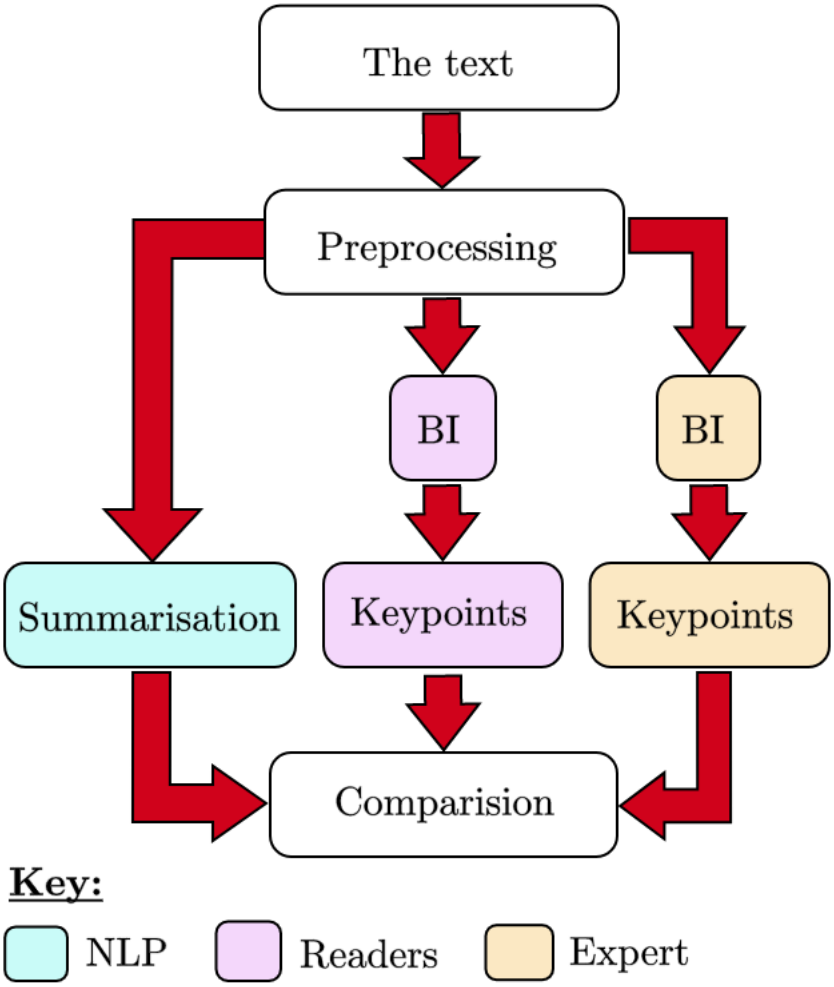


Figure 4.1: Tasks completed by the NLP are shown in turquoise, those completed by the readers in lilac and those completed by the expert in orange. First, the text was preprocessed. Then the readers and the expert were requested to fill in a background-information questionnaire (BI). Next, the NLP summarised the document, and the readers and expert completed the keypoints exercise. Finally, a comparison was conducted on the output of the summarisation/keypoints tasks.

reader's background was a suggestion given in Ref. [6] as understanding why different readers have different interpretations of the text is an area of interest. The background questionnaire given to readers can be found in Appendix A. Details of reader selection and the expert can be found in Sections 4.2.2 and 4.2.3. Collectively the readers and the expert are referred to as the 'participants'.

The participants are then asked to complete the keypoints task; details of the task are discussed in Section 4.2.5. The NLP is used to generate a summary of the document. A discussion of the NLP techniques used in this study is given in Section 4.2.6.

Comparative analysis will be conducted with the output of the NLP's summarisation task and output from the participant's keypoints task. Details of the methods used for comparative analysis are discussed in Section 4.2.7. Approval from Imperial College London's ethics committee was granted for this study.

4.2.1 Stimulus materials

An eight-page technical report providing an overview of the effect of electric and magnetic fields on propellant combustion was selected as the document used in this study. The document contains a review of scientific and technological developments on the effects of magnetic and electric fields on combustion. The paper's emphasis is on solid propellant combustion, but relevant papers describing the effect on liquid or gaseous combustion are also reviewed. This paper aims to establish potential routes of exploration for future defence-based endeavours.

The report was selected as it is on a well-defined neutral subject matter, reducing the number of additional biases or prejudices in the participants' interpretation of the text. Furthermore, the report's length is a manageable number of pages for human analysis. The report is of a technical nature but focuses in detail on a specific domain.

4.2.2 Reader selection

This study recruited seven readers on a voluntary basis. The inclusion criteria for the study was that readers must have a STEM background. The background-information questionnaire details the readers' current occupation, higher-education qualifications and work history. In terms of occupation, participants were asked to tick the relevant box of five categories: government policymaker, industry scientist, academic scientist, government scientist or other. Participants were also asked to detail the number of years spent in each profession.

4.2.3 The expert author

In this study, the domain expertise is limited to a single document; the most apparent expert on the information contained in the document is its author. As with the study conducted in

Ref. [6], it was decided the expert in this study would be the author of the document. The expert was recruited voluntarily.

In this case, the expert has over 17 years of experience working in commercial research areas and over 23 years of experience in government research in fields related to the energetic-materials domain. They, therefore, have an extensive knowledge base in the selected field of propellant combustion.

4.2.4 Preprocessing methodology

Section 4.1.1 gave a general overview of preprocessing techniques in NLP. This section will give an overview of how these techniques will be applied to develop the preprocessing stage of an automatic-summarisation pipeline. An overview of the preprocessing methodology used in this study is outlined in Figure 4.2. Figure 4.2 shows the first step in the preprocessing methodology

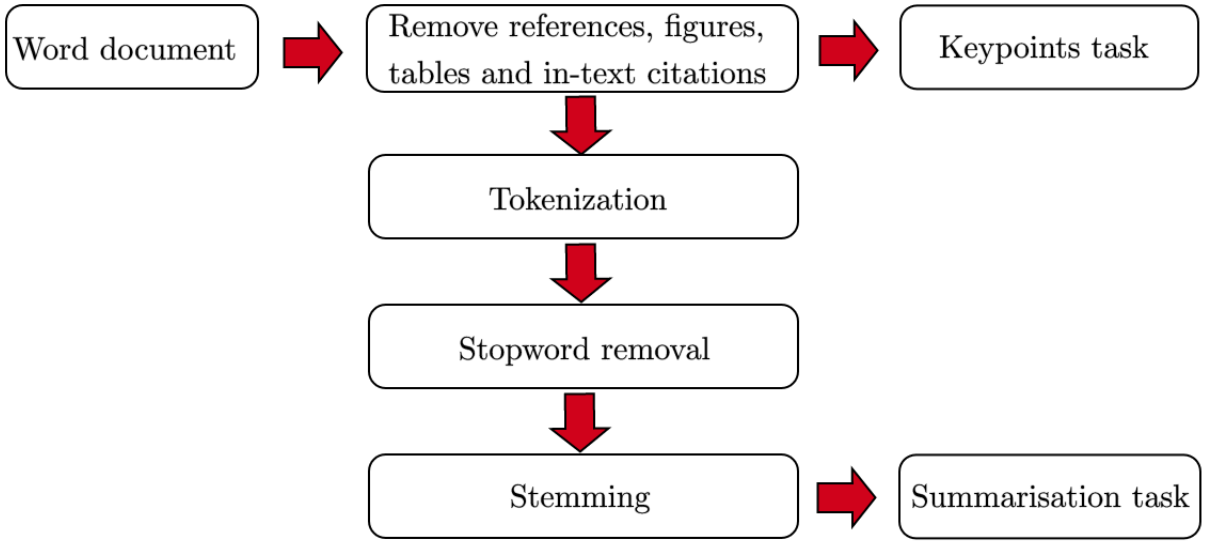


Figure 4.2: Overview of the methodology used to preprocess the document.

is to take the text in a Microsoft Word document form and generate a corpus that is equally understandable by machines and humans. Following recommendations from Ref. [6], references, titles and subtitles are removed from the text. References are removed from the text so as not to influence the reader by giving information where the author's identity of the referenced papers could be deduced (knowledge of the author introduces bias depending on whether participants view them positively or negatively). Titles and subtitles are also cut out not to direct the reader

to key pieces of information, as this could give them an unfair advantage over the automatic text summarisation algorithms.

In addition, images and their captions are deleted as NLP techniques do not apply to images. Including images would give the readers an unfair advantage over the NLP given the present study concerns only the ability of humans and NLP to analyse text alone. At this stage in the preprocessing, the text can be given to the participants to complete the keypoints task.

Following this, the issue of reformatting the text document so it is in a machine-interpretable format is tackled. Following on from this, tokenisation is performed. NLTK's tokenisation function was used to complete this step [311]. Next, all words are converted to lowercase, which is achieved using Python's 'lowercase()' function [312]. This step is performed to ensure the NLP recognises the lower- and upper-case words as being the same word.

Then, stopwords are removed using a custom stopword list. NLTK's preexisting stopword list was modified by adding the Python string list of punctuation [312], all numbers preceding a percentage sign and all numbers/functions of numbers. In addition, common words such as 'figure' and 'reference' are added to the stopword list, as were in-text citations.

Finally, stemming can be performed on the text. As mentioned in Section 4.1.1, the performance of a stemmer is domain-dependent. No comparison of stemming algorithms in the energetic material domain has been conducted in the literature. A recommendation from Section 4.1.8 was to compare the performance of Lovins, Paice-Husk and Porter stemming algorithms. Therefore, using four pages of the text, a comparison of stemmer performance was conducted. Every unique word in the stemmed text and the words it stemmed from were listed for each of the aforementioned stemmers. Incidents of over- and under-stemming were recorded and compared. The overall performance (percentage of correctly-stemmed words) of these stemmers is summarised in Table 4.1.

Stemming algorithm	Percentage of correctly-stemmed words
Lovins	92%
Paice-Husk	93%
Porter	95%

Table 4.1: Percentage of correctly-stemmed words in four pages of the text using the Lovins, Paice-Husk and Porter stemming algorithms.

Table 4.1 shows the best-performing stemming algorithm was the Porter stemmer with 95% of

words stemmed correctly. This stemmer was used to preprocess the text. After stemming is conducted, the NLP summarisation task can be performed.

4.2.5 Keypoints task

The keypoints task is based on the keypoints task in Ref. [6] differing only in the subject matter of the text being reviewed. The keypoints task asks participants to read and digest the text. After reading the stimulus material, they were asked to list the points they felt were most significant in the text. These points are henceforth referred to as 'keypoints'. The definition of importance was left to the participant to interpret. No constraints regarding the number of points or the word limit of a keypoint were given. The participants were also asked to identify the five most important keypoints in the text and then rank them in order of importance, with 1 being the least important and 5 being the most important. This information was elicited via a questionnaire; see Appendix B for a copy of the template given to participants.

4.2.6 NLP-summmarisation task

In Section 4.1.8, a series of recommendations were made for developing a summarisation pipeline for this study. Among these recommendations were that the following algorithms: SumBasic, LSA, TextRank and the clustering-based 'bert-extractive-summarizer', should be used as part of a summarisation pipeline. The performance of these algorithms will be compared and, if appropriate, different feature-representation methods will be explored to establish the best-performing summarisation algorithm. This subsection will provide an overview of the method used to generate the summarisation algorithms.

For SumBasic, LSA, and TextRank, preexisting implementations of these algorithms are available in Python packages such as Ref. [313]. However, one of the recommendations from Section 4.1.8 was to implement the 'cross method' for identifying the sentences to include in the summary of LSA. Furthermore, it is of interest to examine different feature-representation methods of these algorithms. It was determined that preexisting Python implementations were too rigid to do this. Therefore, custom implementations of the algorithms in Python were written. This not only allowed for the algorithms to be easily adapted to any modifications, but also allowed

for the text to be easily preprocessed using the methodology discussed in Section 4.2.4.

The 'bert-extractive-summarizer' approach allows for a range of different pretrained language models to be loaded into the summarisation architecture. It is generally advisable to use a pretrained model that has been trained in the target domain as it will then be able to capture the semantic nuances of the domain. Unfortunately, no open-source language models are trained in the energetic-materials domain. Therefore generic language models were considered. BERT [129] is a state-of-the-art transformer-based language model that can produce contextual representations of text and meaningful representations of out-of-vocabulary words. BERT is discussed in more detail in Section 3.2. The original pretrained BERT model, 'BERT-base', is a general language model trained on 3.3 billion words in a self-supervised fashion [314]. Sci-BERT is a pretrained model trained on a large corpus of scientific publications. It was developed to address the lack of large-scale labelled scientific data [315]. GTP-2 is a similar transformer-based language model; the base model is trained on the text dataset WebText, which contains 40GB of text data from 45 million web links [128]. Sci-BERT, 'BERT-base' and GTP-2 were implemented in the bert-extractive-summarizer architecture for this study. It is noted the language models used in this approach do not take an input of stemmed/tokenised text. Therefore the same level of preprocessing that was applied before the text was given to the participants is used for these models.

All algorithms were set to return 25 sentences (the average number of keypoints the participants returned in the keypoints task).

4.2.7 Evaluation

Intra-group comparison metrics

Ref. [6] introduces two metrics to evaluate how important the readers found the agreed points: $RAIS$ (reader-agreed important points scores) and $MRIS$ (mean reader importance scores). $RAIS$ is a measure of how important the readers *as a group* found an agreed point. $MRIS$ is a measure of how important, *on average*, the readers found an agreed point. Worked examples illustrating the difference between these two scores can be found in Appendix C. This section will describe the methodology used to calculate these scores in this study.

This questionnaire received a total of seven responses. The length of the responses varied from 5 to 23 keypoints. The total number of keypoints received was 98. The style of each keypoint varied from a 'bullet-point style' response to several sentences. In the latter case, each keypoint could consist of several subpoints. Therefore, to generate a measure of reader agreement, each point was divided into subpoints. The analysis resulted in a total of 135 subpoints. Subpoints conveying the same meaning were grouped. The number of conceptually unique subpoints was 91. These conceptually unique subpoints will henceforth be referred to as reader-agreed points. It is noted that it is possible for a reader to have more than one subpoint grouped under a reader-agreed point.

Identifying subpoints in keypoints and grouping them to form reader-agreed points is a very subjective process. For instance, the statements 'There has not been much work reported', and 'Electro-thermal guns using electric solid propellants showed promising results but have had no other further publications in the US' imply a lack of publications. However, it is not possible to determine they are making precisely the same point. For this reason, they are not grouped. To reduce bias, this process was repeated three times at six-month intervals.

As part of the keypoints task, the readers were asked to identify and assign the top 5 most-important keypoints a ranking, R, from 1-5, where 5 is the most-important. These rankings were converted to importance scores (IS) using

$$IS = 1 + R, \tag{4.17}$$

such that the keypoints, that were not ranked in the top 5 most-important key points, have a score of $IS = 1$; while the most-important have a score of $IS = 6$. Each keypoint now has an IS, and each subpoint is assigned the same IS as the keypoint it originates from. $RAIS$ was calculated for the reader-agreed points using

$$RAIS_j = \sum_{i \in J} IS_i, \tag{4.18}$$

where IS_i is the IS for subpoint $i \in J$ where J is the set of subpoints in agreement with reader-agreed point j. $MRIS$ was calculated using

$$MRIS_j = \frac{RAIS_j}{N_j}, \tag{4.19}$$

where N_j is the number of readers that had a subpoint in agreement with reader-agreed point j.

Inter-group comparison metrics

In Section 4.1.7, a metric to evaluate the overlap between two summaries at the sentence level is introduced. However, evaluating at a sentence level may result in the overlap of the two summaries being misrepresented. For example, a human-generated summary may contain two conceptual ideas in one sentence, whereas a machine-generated summary might express the same conceptual ideas over two sentences. The sentence overlap, however, would be 0 as the two summaries do not return the same sentences. Therefore Eq. (3.6) is adapted to evaluate at a conceptual level. To determine the conceptual overlap between two participants (denoted in this case by i and j), the harmonic mean of the content overlap, F_{ij}, is defined as

$$F_{C_{ij}} = \frac{2P_{C_{ij}}R_{C_{ij}}}{P_{C_{ij}} + R_{ij}} \tag{4.20}$$

where $P_{C_{ij}}$ is defined as

$$P_{C_{ij}} = \frac{A_{ij}}{N_i}, \tag{4.21}$$

where A_{ij} is the number of subpoints from participant i's summary found to be in agreement with that of participant j and N_i is the number of subpoints in participant i's summary. $R_{C_{ij}}$ is defined as

$$R_{C_{ij}} = \frac{A_{ji}}{N_j}. \tag{4.22}$$

The metric $F_{\text{ROUGE-1}}$ will also be used to indicate the overlap between summaries. $F_{\text{ROUGE-1}}$ is a modification of Eq. (4.16) that scores the unigram overlap of two summaries. Both summaries follow the same preprocessing methodology described in Section 4.2.4 to normalise the text before evaluating with ROUGE. There are a range of different implementations of ROUGE available in Python [316, 317, 318].

4.3 Results and discussion

In this section, the results of the study are presented. First, a comparison of NLP summarisation algorithms will be made to determine the best-performing algorithm in Section 4.3.1 . Secondly, an overview of the readers' background information will be presented (Section 4.3.2). Thirdly, a comparision of the two metrics, F_C and $F_{\text{ROUGE-1}}$, used to evaluate the quality of the generated summaries will be made in Section 4.3.3. Finally, a comparison between the expert, reader and best-performing NLP summarisation algorithm will be made in Section 4.3.4.

4.3.1 A comparison of text-summarisation algorithms

This section aims to determine the best-performing summarisation algorithm. First, the algorithms identified in the literature review in Section 4.1 are compared. Then a comparison of different feature-representation methods for the best-performing algorithm is made.

Identifying the best-performing NLP algorithm

The preprocessing methodology described in Section 4.2.4 was used to preprocess the text. Then, the four approaches to summarisation identified in the literature review in Section 4.1 as suitable for the task, SumBasic, LSA, TextRank and bert-extractive-summarizer, were used to generate summaries. Three different models were inputted in the bert-extractive-summarizer architecture, Sci-BERT, GTP-2 and BERT. These summaries were compared against the expert's keypoints using conceptual overlap F_C (Eq. (4.20)) and unigram overlap using the metric $F_{\text{ROUGE-1}}$. Table 4.2 displays the results of this comparison.

Algorithm	$F_{\text{ROUGE-1}}$	F_C
SumBasic	0.37	0.36
LSA	0.35	0.31
TextRank	0.39	0.33
BERT-summariser (Sci-BERT)	0.29	0.27
BERT-summariser (GTP-2)	0.34	0.31
BERT-summariser (BERT)	0.30	0.28

Table 4.2: Evaluation of summaries generated by SumBasic, LSA, TextRank and bert-extractive-summarizer with Sci-BERT, BERT and GTP-2 against the expert's keypoints using the metrics $F_{\text{ROUGE-1}}$ and F_C.

The results of Table 4.2 show that basic statistical algorithms outperformed the state-of-the-art BERT-summariser. $F_{\text{ROUGE-1}}$ values of 0.37, 0.35 and 0.39 were obtained for SumBasic, LSA and TextRank respectively. These values are higher than the $F_{\text{ROUGE-1}}$ obtained for all three bert-extractive-summarizer implementations examined. The values of F_C for SumBasic, LSA and TextRank also outperformed those generated for the bert-extractive-summarizer implementations. There are a range of possible reasons for this, which will be explained below.

Firstly, SumBasic, LSA and TextRank are statistical algorithms. It was observed that all three algorithms returned a high number of sentences with high percentages of keywords. This resulted in summary-style sentences being returned, such as 'magnetic control of combustion falls into four main areas: (1) direct control of the burning at the surface of a solid propellant, (2) influence on the manufacturing process by forcing alignment of metallic needle-like particles within the propellant, (3) influence of the combustion flow patterns within a chamber or nozzle, (4) upstream polarisation of the fuel system.'. These summary-style sentences portrayed not only key technical concepts but had a high proportion of key terms and phrases that also appeared in the expert's summary. This resulted in higher $F_{\text{ROUGE-1}}$ and F_C.

Secondly, whilst bert-extractive-summarizer uses state-of-the-art sentence embeddings, the models used to generate these embeddings were trained on out-of-domain text. As a result, these embeddings may not have been able to capture domain-specific semantic relationships and consequently may be less semantically meaningful in this domain. In addition, the method of generating the embeddings from the pretrained models was not tailored to this domain. This may have resulted in lower algorithm performance.

Three different pretrained models were implemented as part of the bert-extractive-summarizer architecture. It was expected the Sci-BERT implementation would perform the best as it was trained on scientific text, whereas the other two examined implementations, GTP-2 and BERT, are trained on a general corpus. This, however, was not the case as Table 4.2 shows the Sci-BERT implementation was outperformed by the GTP-2 and BERT implementations for both $F_{\text{ROUGE-1}}$ and F_C. This may be because the style in which the text was written was closer to the text used to train the GTP-2 and BERT models than the Sci-BERT models. Alternatively, the method used to generate the embeddings from the pretrained models may have been better suited to the GTP-2 and BERT models. Future work should explore different approaches to generating embeddings from the models, for instance using S-BERT [319].

Table 4.2 shows F_C was lower than $F_{\text{ROUGE-1}}$ for all examined algorithms. This illustrates that summaries containing the same words and consequently having a high unigram overlap may not be portraying the same conceptual points and, as a result, have a lower conceptual overlap.

In addition, some sentences returned by the algorithms were returned out of context, causing them to lose the original meaning they expressed in the text. For instance, the sentence 'Again inconsistent results have been obtained.' appears in the text as a summary statement about electric control of combustion, specifically direct control of the burning at a surface of a solid propellant. Its meaning is spread over two sentences. The summary returned by the bert-extractive-summarizer with GTP-2 this sentence appears after the following sentences 'For a typical example, the outer electrode is a 3.2mm diameter aluminium tube, and the concentric inner electrode is a 1.2mm diameter molybdenum rod. The former method has been applied only at pressures suitable for rocket propellant applications'. In this context, it is unclear what the meaning of 'Again inconsistent results have been obtained.' is. This issue of selecting sentences that contained words requiring context also contributed to a lack of flow in the summaries. This issue was observed to a greater extent in summaries generated using bert-extractive-summarizer. In Ref. [301] a similar issue is identified as the bert-extractive-summarizer identifying sentences that contain words that require further context e.g. 'this', 'those', 'these', and 'also'.

The TextRank algorithm outperformed all other algorithms with both conceptual overlap and uni-gram overlap, achieving a $F_{\text{ROUGE-1}}$ of 0.39 and F_C of 0.33. These obtained values are in line with state-of-the-art results found in the literature [320] and results in the original TextRank paper [266]. The F_C score can be interpreted as the two summaries agreeing on 33% of the concepts.

Overall, the returned summaries showed a lack of cohesion. Sentences appeared in seemingly random order and did not flow. This is a typical limitation of extractive-summarisation techniques. The expert's summary contained concepts summarised in a single sentence that appeared over several sentences in the text. The summarisation algorithms were unable to identify such concepts and return them accordingly. The bert-extractive-summarizer summaries were found to flow less. This is because the summaries generated using LSA, SumBasic and TextRank returned sentences in the order in which they appeared in the text. It is noted that this is not in agreement with the results in Ref. [301] which compared TextRank to the bert-

extractive-summarizer on a Health Information Exchange lecture series and found the cohesion bert-extractive-summarizer summaries to be more cohesive.

The preprocessing methodology also resulted in errors. The sentence tokenizer was confused by the presence of some abbreviations and chemical names, which resulted in some sentences being incorrectly tokenized. For instance, the presence of the word 'i.e.' caused the sentence tokenizer to interpret this as the end of a sentence mistakenly. This would have had a significant impact on TextRank, LSA and SumBasic, which use frequency-based measures on a sentence-by-sentence level. Incorrect identification of a sentence boundary will result in incorrect values for these statistics. The bert-extractive-summarizer has an in-built tokenizer and was not affected similarly. Future work on energetic-materials text may require the development of a custom preprocessing pipeline.

The best-performing algorithm of those examined was TextRank. Following the recommendations in Section 4.1, different modifications to the original TextRank algorithm should be examined. This will be explored in the next section.

A comparison of TextRank modifications

Here 'TextRank cosine TF-ISF' and 'TextRank cosine Word2Vec', modified versions of the original TextRank algorithm, 'TextRank Basic Jacquard' (Eqs. (4.4) and (4.5)), will be compared.

TextRank cosine TF-ISF and TextRank cosine Word2Vec use TF-ISF (Eq. (4.7)) and a pretrained Word2Vec model, respectively, to generate sentence vectors. The weights on the TextRank graph are then calculated using the cosine of the relevant sentence vectors (Eq. (4.8)). There are a range of pretrained models available in Python, the most notable being those trained on the Google News corpus [120]. However, no open-source Word2Vec models trained on text in the energetic-materials domain exist. Therefore, for this study, embeddings trained using the Skip-gram algorithm on 3 billion tokens from English Wikipedia are used [321]. Since the model is trained on the entire English Wikipedia, it would have encountered some references to energetic materials. Sentence embeddings are generated by averaging the word vectors for each word in the sentence (excluding stopwords). However, there is some dispute over the effectiveness of this method in the literature. For instance, in Ref. [322], embeddings generated in this manner were not found to be semantically meaningful; in contrast, in Ref. [323], this

method was found to be effective.

Using out-of-domain embeddings meant some of the words in the text were not in the training corpus. This means they do not have word embeddings, e.g. 'ambipolar', 'micro-engine', 'carboxylated' and 'dioctyl'. These words are referred to as 'out-of-vocabulary'. In addition, in-text equations and units were also out-of-vocabulary. A complete list can be found in Appendix D. A large proportion of these out-of-vocabulary words had hyphens. Therefore hyphenated out-of-vocabulary words were split into individual tokens. If the remaining words were out-of-vocabulary, they were removed from the text. This problem could have been avoided if the model had been trained on domain-specific text.

The performance of TextRank cosine TF-ISF and TextRank cosine Word2Vec was evaluated against the expert's keypoints using $F_{\text{ROUGE-1}}$ and F. Results are displayed in Table 4.3 alongside the performance of TextRank Basic Jacquard.

Algorithm	$F_{\text{ROUGE-1}}$	F_C
TextRank Basic Jacquard	0.39	0.33
TextRank Word2Vec cosine	0.41	0.39
TextRank cosine TF-ISF	0.41	0.39

Table 4.3: Evaluation of 'TextRank Basic Jacquard', 'TextRank cosine TF-ISF' and 'TextRank Word2Vec cosine' against the expert's keypoints using $F_{\text{ROUGE-1}}$ and F.

Table 4.3 shows that, as with the results in Table 4.2, all values of $F_{\text{ROUGE-1}}$ are less than F_C. This is further evidence supporting the conclusion that returning sentences with a high unigram overlap will not result in an equally high conceptual overlap.

Both modified implementations performed equally well obtaining $F_{\text{ROUGE-1}}$ of 0.41 and F_C of 0.39, outperforming 'TextRank Basic Jacquard' which obtained $F_{\text{ROUGE-1}}$ of 0.39 and F_C of 0.33. This F_C value can be interpreted as the NLP and expert agreeing on 39% of concepts. This shows the summaries generated by the modified versions of the TextRank algorithm had a greater word and conceptual overlap with the expert's keypoints. It can therefore be concluded the two modifications of the TextRank algorithms outperformed the original implementation. This agrees with the results found in Refs. [288, 282]. It is noted the generated summaries suffered from the lack-of-cohesion issues described in the previous section.

Despite being two different methods of feature representation, the TextRank cosine TF-ISF and TextRank Word2Vec cosine algorithms returned identical sentences and hence yielded identical

values $F_{\text{ROUGE-1}}$ and F_C. The algorithms perform equally well at extractive summarisation. In this case, using transfer learning methods did not improve performance over traditional statistical methods of feature representation. There are other examples in the literature of traditional statistical feature-representation methods outperforming state-of-the-art language models for small/niche datasets [324, 325].

A significant limitation of the 'TextRank Word2Vec cosine' method was the use of a general pretrained model. This led to some of the words in the text being out-of-vocabulary. The out-of-vocabulary words are hence removed from the text. This led to poorer quality sentence vectors. Using a Word2Vec model trained on energetic-materials text would not only lead to more accurate semantic representations of the words but led to fewer out-of-vocabulary words, thus improving performance. The method of generating sentence embeddings from the Word2Vec model was to average over the word embeddings of the words in each sentence. Future work should investigate other methods of generating sentence embeddings from Word2Vec models, such as using TF-ISF weightings in the averaging process.

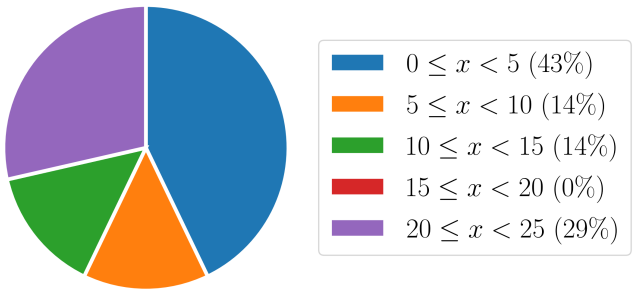
To conclude, TextRank outperformed SumBasic, LSA and the state-of-the-art bert-extractive-summarizer architecture. Furthermore, two modifications of TextRank, 'TextRank cosine TF-ISF' and 'TextRank Word2Vec cosine', outperformed the original version of TextRank. Despite different methods of feature representation, these methods returned identical sentences and hence yielded identical word and conceptual overlap with the expert's keypoints. Since the modifications produced identical summaries, the best-performing algorithm will henceforth be referred to as 'modified TextRank'. This summary will be used to compare with the readers' and expert's keypoints.

4.3.2 Background information of the readers

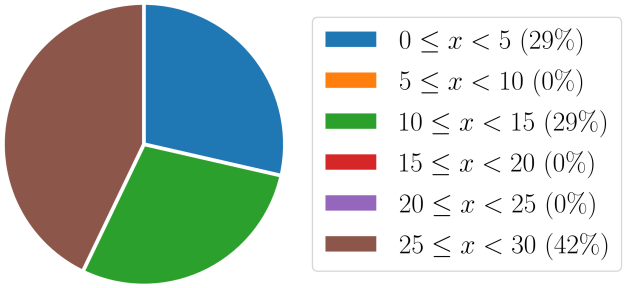
The participants in this study were asked to complete a background-information questionnaire (Appendix A). This questionnaire collected details on the participants' experience levels and expertise in the topic of the text (effect of electric and magnetic fields on propellant combustion). This section will present an overview of the readers' responses to the background information. This information will be used in later analysis (Section 4.3.3) to understand how a reader's background influences their response to the keypoints task.

The readers were requested to specify the number of years of experience they have in academic, government and commercial research as well as government policy. None of the readers had experience in government policy. Only two readers had experience in government research, one totalling 24 years and the other totalling 0.25 years. Figures 4.3 (a) and (d) display the proportion of readers by the number of years of experience they have in academic/commercial research. The total number of years of experience of each reader was totalled. Figure 4.3 (b) displays the proportion of readers by their total years of experience.

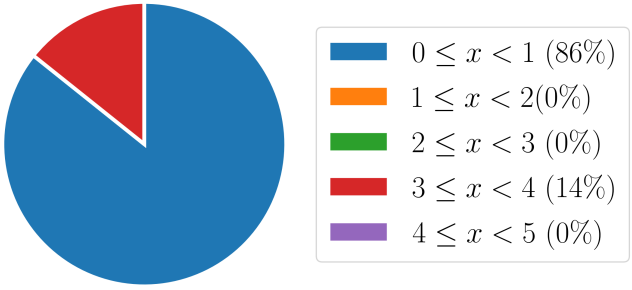
The readers were also asked to assign themselves an expertise rating on a 1-5 scale, where a score of 5 would consider them an expert on the effect of electric and magnetic fields on propellant combustion. The results are displayed in Figure 4.3 (d).



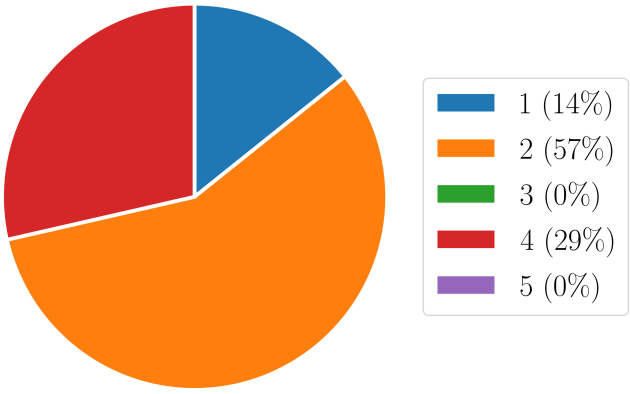
(a) Proportion of readers by the number of years of experience in academic research.



(b) Proportion of readers by the total number of years experience.



(c) Proportion of readers by the number of years experience in commercial research.



(d) Proportion of readers by their expertise rating.

Figure 4.3: Proportion of readers by: numbers of years of experience in academic research (a) or commercial research (c), number of years total experience (b) and expertise rating (d).

Figure 4.3 (a) shows the majority of the readers are not in the early stages of their careers, with over 67% having over 5 years of experience in academic research. This is further illustrated in Figure 4.3 (b), which shows a sizeable proportion of the participants (43%) having a total of 20-25 years of experience. Most of the recruited participants have limited experience in commercial

research, with only one participant having more than 3 years of experience. The readers were also asked to specify any higher-education qualifications. All of the readers in this study were either studying for or had completed a PhD. From this analysis, it is concluded most readers are experienced academics. These results indicate the readers are experienced and, therefore, fully equipped to complete the keypoints task. However, this is a limited selection of readers. Future work should aim to attract a more significant number of readers from more varied backgrounds. This will allow for better comparisons of how a reader's background influences their response to the keypoints task.

Figure 4.3 (d) shows the majority of readers (71%) rated themselves as having little expertise (1-2) on the topic of the effect of electric and magnetic fields on propellant combustion. Two of the participants rated themselves as having an expertise level of 4. To examine whether those participants rated themselves the highest were the most experienced, the Pearsons correlation between the total number of years of experience and the self-assigned expertise rating was calculated. However, no statistically significant correlation was obtained. This indicates there was no linear relationship between the self-assigned expertise level and the total number of years of experience. This may be due to the sample size being too small to view any significant correlation. Alternatively, this could be due to the inherent subjectivity in asking participants to assign themselves an expertise rating or because the total number of years of experience may not correspond to relevant experience.

4.3.3 Inter-group and intra-group comparison

Reader intra-group agreement: NLP vs Expert

In Section 4.2.7, two metrics, $RAIS$ (Reader Agreed Importance scores) and $MRIS$ (Mean Reader Importance Scores), were introduced to measure how important the readers as a group found an agreed point and how important, on average, the readers found an agreed point[7]. These are measures of group importance (inter-group importance). In this section, analysis using $RAIS$ and $MRIS$ will be conducted.

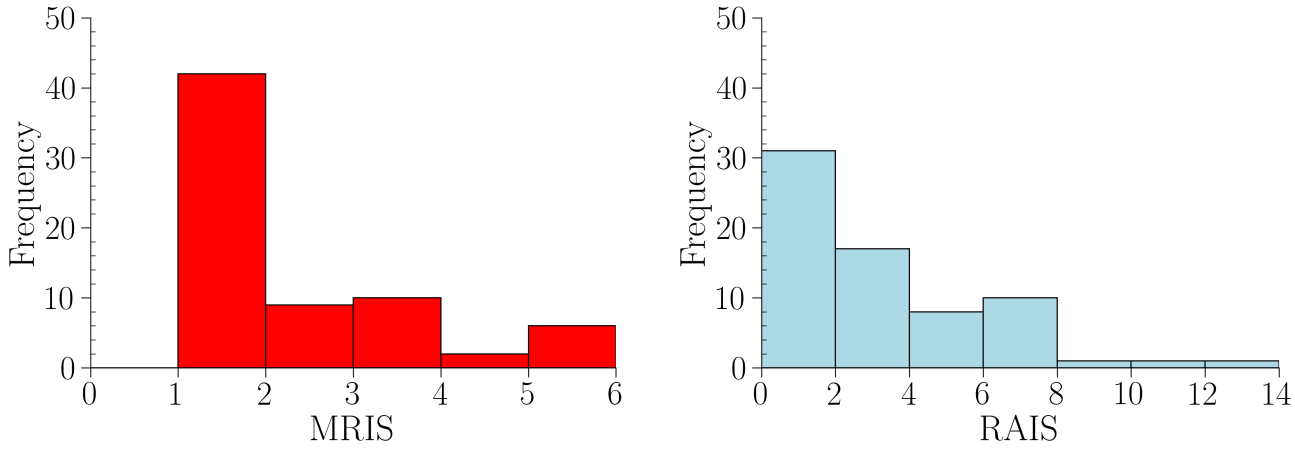
The method presented in Ref. [6] divided reader-agreed points (RAPs) into two sets of datasets:

[7]See worked examples demonstrating the difference between these two scores in Appendix C.

(1) those the NLP found important and not important and (2) those the expert found important and not important. T-tests were then performed on the associated $RAIS$ or $MRIS$ scores for each set of datasets. The purpose of this analysis was to determine if the reader-agreed points were viewed as more important by the readers if they agreed with the points made by the NLP (using the first dataset (1))/the expert (using the second dataset (2)). This analysis is adapted to the data in this study.

Conducting a T-test requires the data to meet the following conditions: measured values on a scale, a simple random sample, normally distributed, appropriate sample size, and homogeneity of variance [326].

$RAIS$ and $MRIS$ scores are a scale of measurement, and they are assumed to be simple random samples[8]. The normality of the data is examined by plotting the distribution of the $RAIS$ and $MRIS$ in Figure 4.4.



(a) Distribution of $MRIS$ for the reader-agreed points. (b) Distribution of $RAIS$ for the reader-agreed points.

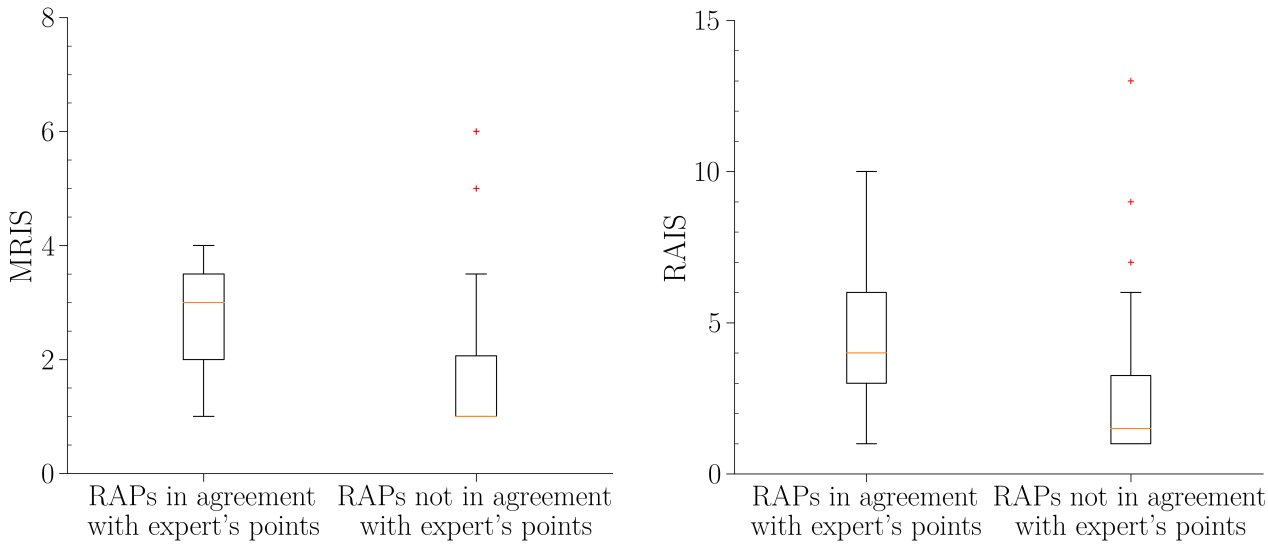
Figure 4.4: Distribution of $MRIS$ (a) and $RAIS$ (b) for the reader-agreed points (RAPs).

Figures 4.4 (a) and (b) show the distributions of $RAIS$ and $MRIS$ are heavily skewed to lower values of $RAIS$ and $MRIS$. This indicates the datasets as a whole are non-normal and non-parametric analysis may need to be utilised. The reason for the skew is many readers' keypoints were not found to agree with any other readers' keypoints. As a result, these points had low associated $RAIS$ and $MRIS$ scores. Similarly, in Ref. [6], a large conceptual variety was found in the readers' keypoints. In Ref. [327] a comparison of what novices and experts found to be

[8]The question of whether or not these scores are a simple random sample is complex, this implies the scores are an unbiased representation of the general population, but given the way the scores are computed (a biased method), this is not entirely true.

important in physics texts also found considerable variation in responses. As pointed out in Ref. [6] from a psychological perspective, this is unsurprising as experts bring to any task their own expectations drawing from their experiences which will influence their interpretations of the task. These varying approaches to completing the task will result in varied responses.

First intra-group agreement between the readers and the expert is examined. In order to determine the correct statistical test, the distributions of $RAIS$ and $MRIS$ for statements in agreement or not in agreement with the expert are examined individually. A total of 9 reader-agreed points were found to agree with the expert, and 60 were not in agreement with the expert. The distribution of these datasets is plotted using boxplots in Figure 4.5.



(a) Distribution of $MRIS$ for the reader-agreed points in agreement/not in agreement with the expert.

(b) Distribution of $RAIS$ for the reader-agreed points in agreement/not in agreement with the expert.

Figure 4.5: Distribution of $MRIS$ (a) and $RAIS$ (b) for the reader-agreed points (RAPs) in agreement with the expert and not in agreement with the expert. Red crosses indicate outliers and orange lines indicate median scores.

Figures 4.5 (a) and (b) indicate the distribution of $MRIS$ and $RAIS$ scores not in agreement with the expert are heavily skewed towards lower values, whereas the distribution of $MRIS$ and $RAIS$ scores in agreement with the author are considerably less skewed. Of those points that were found to agree with the expert, 4 out of 9 were general statements that described the overall theme of the text; an example is 'Both magnetic and electric fields may affect the combustion process'. Unsurprisingly, general statements tended to agree with at least one other reader generating high $MRIS$ and $RAIS$ scores, thus resulting in less skew for the distributions for RAPs in agreement with the expert.

In addition, Figures 4.5 (a) and (b) show some outliers where the group of readers found a RAP to be important and thus the RAP had a high $MRIS/RAIS$ but the expert did not find the RAP important. These outlying RAPs had no consistent attributes. Overall, the median $MRIS$ and $RAIS$ of RAPs in agreement with the expert, which had values of 3 and 4 respectively, were higher than RAPs not in agreement with the expert, values of 1 and 1.5 respectively. This preliminary analysis indicates the intra-group agreement is higher for RAPs in agreement with the expert. Further statistical analysis is needed to validate this.

From Figure 4.5 (a) and (b), the $MRIS$ and $RAIS$ datasets for RAPs in agreement with the expert appear normally distributed, whereas the datasets for RAPs not in agreement with the expert appear non-normal. The normality was validated using the Shapiro-Wilk test [328].

Firstly the $RAIS$ scores are examined, p-values of 0.84 and 5e-11 were obtained for $RAIS$ of RAPs in agreement with the expert and RAPs not in agreement with the expert respectively. From this, it is concluded that at a 95% significance level, since $0.84 > 0.05$ and $5 \times 10^{-11} < 0.05$, $RAIS$ for RAPs in agreement with the expert are not statistically different from the normal distribution, whereas those $RAIS$ of RAPs not in agreement with the expert are statistically different from the normal distribution and are described as non-normal. The dataset of $RAIS$ of RAPs in agreement with the expert had a sample size of 9. At such low sample sizes, tests of normality, such are the Shapiro-Wilk test are inaccurate [329]. Therefore non-parametric tests should be used.

An unpaired Wilcoxon rank sum test with continuity correction (see Ref. [330] for further details) was performed[9]. This test was selected as it does not assume normality of the data. An unpaired Wilcoxon rank sum test test compares the medians between the two datasets. The obtained p-value of the test is 0.011, which is less than the significance level of 0.05. Therefore it can be concluded the median $RAIS$ of RAPs in agreement with the expert is significantly different from the median $RAIS$ of RAPs not in agreement with the expert. The median difference between a sample from $RAIS$ statements in agreement with the expert and a sample from $RAIS$ statements not in agreement with the expert in location was determined to be 2. The size effect was measured using Vargha and Delaney's A (see Ref. [331] for details). An estimate of 0.75 was obtained, implying a large difference between the two datasets. This

[9] See Appendix E for an overview of the decision-making process used to determine the appropriate statistical test.

value can be interpreted as 75% of $RAIS$ scores being higher for RAPs in agreement with the expert than RAPs not in agreement with the expert. From this, it is concluded there is more agreement between readers for RAPs that the expert also finds important than for RAPs the expert does not find important.

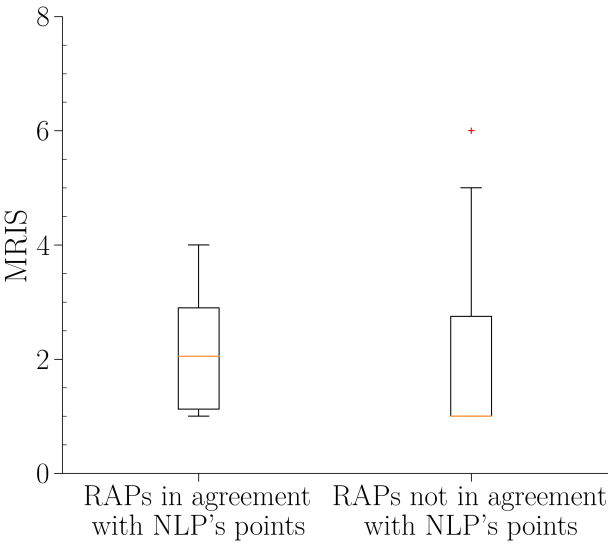
Secondly, the $MRIS$ scores are examined. The Shapiro-Wilk test is performed to assess the normality of the data. P-values of 0.20 and 5×10^{-11} were obtained for the dataset of $MRIS$ for RAPs in agreement with the expert and RAPs not in agreement with the expert, respectively. Since $0.20 > 0.05$ and $5 \times 10^{-11} < 0.05$, it is concluded the distribution of $MRIS$ for RAPs in agreement with the expert is not significantly different from the normal distribution but the dataset of $MRIS$ for RAPs not in agreement with the expert are not normally distributed. As discussed above, the dataset of $MRIS$ of RAPs in agreement with the expert has a sample size of 9. At such low sample sizes, normality tests are inaccurate [329]; therefore non-parametric tests should be used.

An unpaired Wilcoxon rank sum test with continuity correction was performed on the $MRIS$ scores. The p-value of the test was 0.013, and this is less than the significance level of 0.05. Therefore the median $MRIS$ of RAPs in agreement with the expert is significantly different from the median $MRIS$ of RAPs not in agreement with the expert. The median difference between a sample from $MRIS$ of RAPs in agreement with the expert and a sample from $MRIS$ of RAPs not in agreement with the expert was 1. The size effect was measured using Vargha and Delaney's A. An estimate of 0.73 was obtained, again implying a significant difference between the two populations. It is therefore concluded that, on average, the readers find RAPs in agreement with the expert more important than those not in agreement with the expert.

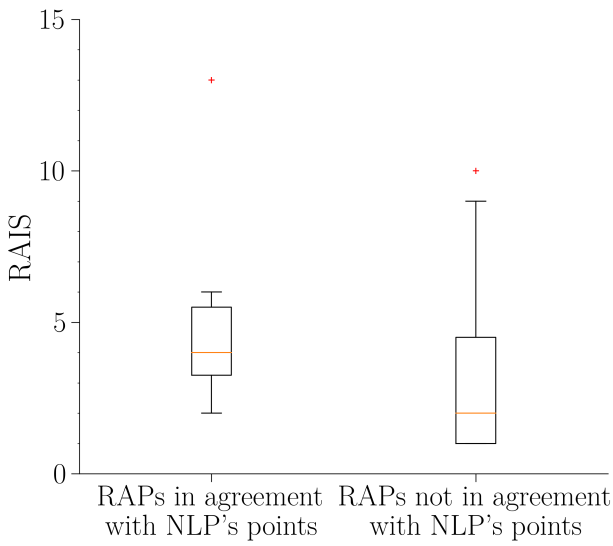
Thus far, it has been established that as a group, the readers find RAPs in agreement with the expert more important and, on average, are rated more important than those not in agreement with the expert.

The same analysis is applied to the datasets of RAPs in agreement/not in agreement with the NLP. A total of 6 RAPs were in agreement with the NLP, and 63 were not. The distribution of these datasets is plotted in Figure 4.6.

Figures 4.6 (a) and (b) show the distribution of $MRIS$ and $RAIS$ scores of the RAPs not in agreement with the NLP are more heavily skewed towards lower values than the scores for RAPs



(a) Distribution of $MRIS$ for the reader-agreed points in agreement and not in agreement with the NLP.



(b) Distribution of $RAIS$ for the reader-agreed points in agreement and not in agreement with the NLP.

Figure 4.6: Distribution of $MRIS$ (a) and $RAIS$ (b) for the reader-agreed points (RAPs) for those in agreement with the NLP and not in agreement with the NLP. Red crosses indicate outliers and orange lines indicate the median scores.

in agreement with the NLP. Of those points that were found to agree with the expert, 3 out of 6 were either general statements about the topic of the text or statements stating a summary conclusion, for instance, 'Electrothermal guns using electric solid propellants showed promising results but have had no other further publications in the US'. As mentioned in Section 4.3.1, the NLP returned conclusion-style statements frequently as they tend to contain a high frequency of keywords. As mentioned above, this RAP style also tends to have an agreement with at least one other participant.

The median $MRIS$ and $RAIS$ score of the RAPs in agreement with the NLP (2 and 4 respectively) was higher than RAPs not in agreement with the expert (1 and 2 respectively). This initial analysis indicates that group agreement is higher for RAPs in agreement with the NLP. However, further statistical analysis is needed to validate this. Figures 4.6 (a) and (b) show the distribution of $RAIS$ and $MRIS$ for RAPs that agree with the NLP to be normal, whereas the distributions of $RAIS$ and $MRIS$ for statements not in agreement with the NLP do not. These hypotheses are validated using Shapiro-Wilk tests. Consider the $RAIS$ scores first, p-values of 0.04 and 2×10^{-10} were obtained for RAPs in agreement with the expert and not in agreement with the NLP, respectively.

Given the non-normal nature of the datasets of the $RAIS$ scores, an unpaired Wilcoxon rank sum test with continuity correction was performed. The p-value of the test was 0.02, which is less than the significance level of 0.05. Therefore the median $RAIS$ of RAPs in agreement with the NLP is significantly different from the median $RAIS$ of RAPs not in agreement with the NLP. The median difference between a sample from $RAIS$ RAPs in agreement with the NLP and a sample from $RAIS$ RAPs not in agreement with the NLP was determined to be 2. The size effect was measured using Vargha and Delaney's A. An estimate of 0.77 was obtained again, implying a significant difference between the two populations. It is therefore concluded that as a group, readers find RAPs in agreement with the NLP more important than those not in agreement with the NLP.

Next, the $MRIS$ scores are examined. P-values of 0.40 and 2×10^{-10} were obtained for RAPs in agreement with the NLP and not in agreement with the NLP, respectively. Given the low sample size of the $MRIS$ scores for RAPs in agreement with the NLP, an unpaired Wilcoxon rank sum test with continuity correction was performed. The p-value of the test was 0.33, at a 0.05 significance level. This implies there is insufficient evidence to reject the null hypothesis. It, therefore, cannot be conclusively determined whether, on average, participants find RAPs in agreement with the NLP more important than those not in agreement with the NLP[10].

Inter-group agreement: readers vs expert vs NLP

Previously in this section, reader intra-group analysis was conducted. Here the readers will be considered individuals and randomly assigned a numeric label. The inter-group agreement will be examined for the participants (the readers and the expert) and the NLP using two metrics, conceptual overlap (F_C) and unigram overlap ($F_{\text{ROUGE-1}}$).

A matrix containing the F_C scores between the keypoints/summary for every possible pairing of individuals in the group and the NLP is displayed in Figure 4.7. Figure 4.7 shows the highest F_C for 4 of the 7 readers was with the expert. The highest F_C was a value of 0.61 between Readers 7 and 5. Their keypoints were not just similar conceptually but were also similar in the style in which they were written; both readers wrote lengthy, detail-orientated responses. The readers had similar numbers of years of experience in academic research, suggesting they have

[10] A statistical test cannot be conducted between the RAPs found important by the NLP and expert as some but not all samples are paired, and the two datasets are of different sample sizes.

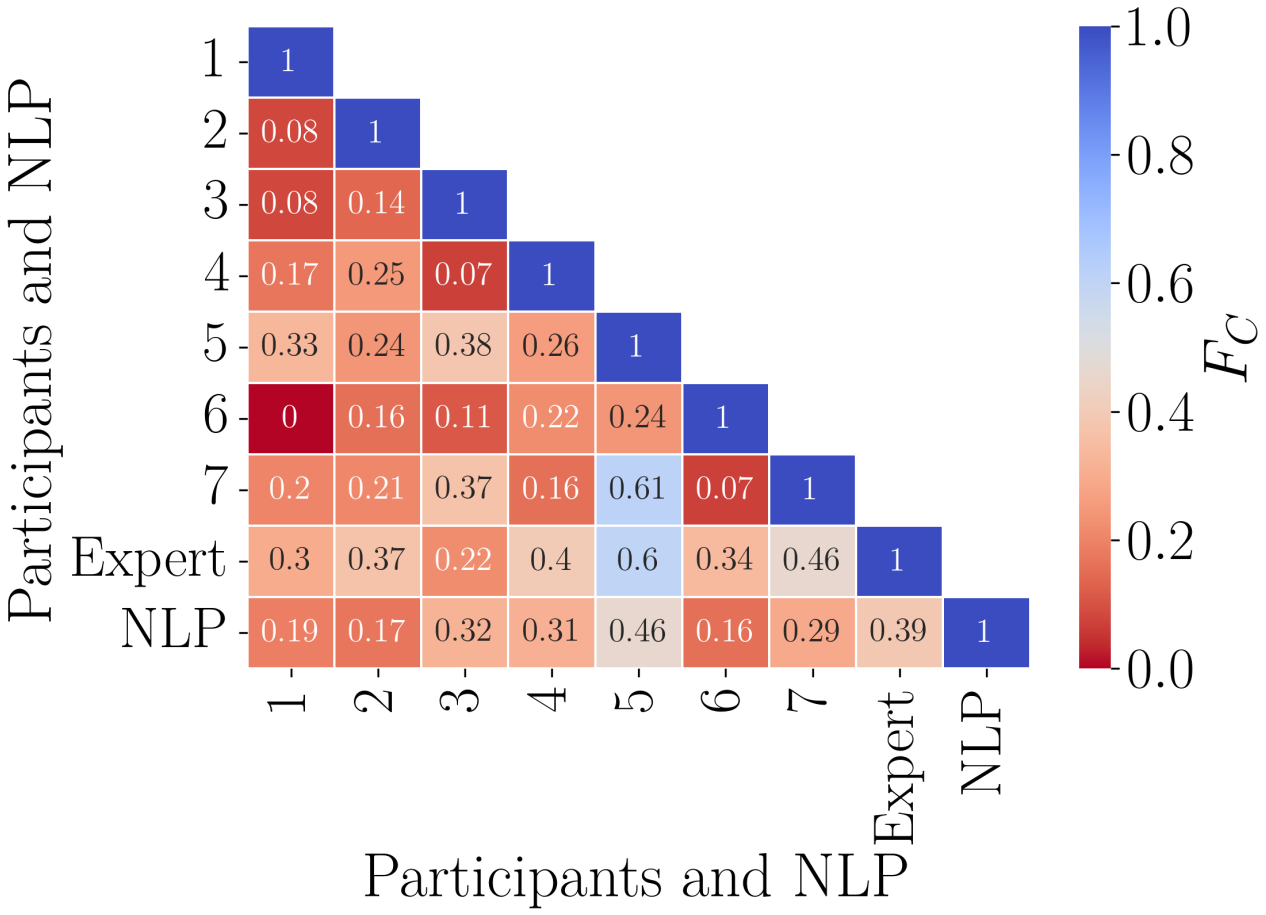


Figure 4.7: Matrix displaying the F_C overlap of the keypoints/summaries for every possible pairing of individuals in the group and the NLP.

similar backgrounds, which may be why they analysed the text similarly. Reader 5 also obtained high F_C with Reader 1 (0.33) and Reader 3 (0.38). These readers had either their self-assigned expertise level in common or similar levels of experience. This initial analysis implies the collected background information may be useful in predicting the agreement between readers.

The NLP exhibits the highest F_C with the expert obtaining a value of 0.39. This is unsurprising as in Section 4.3.1, a range of NLP tools were investigated, and the best methodology was chosen based on F_C and $F_{\text{ROUGE-1}}$ with the expert. Some readers' keypoints had no conceptual overlap. This reflects the extent of the conceptual variation in the readers' keypoints mentioned earlier in this section.

The average F_C among the readers ($\langle F_C \rangle$) was calculated for each of the participants/NLP. The results are displayed in Figure 4.8.

Figure 4.8 shows that, on average, the readers had an F_C with the expert of 0.39. This is higher

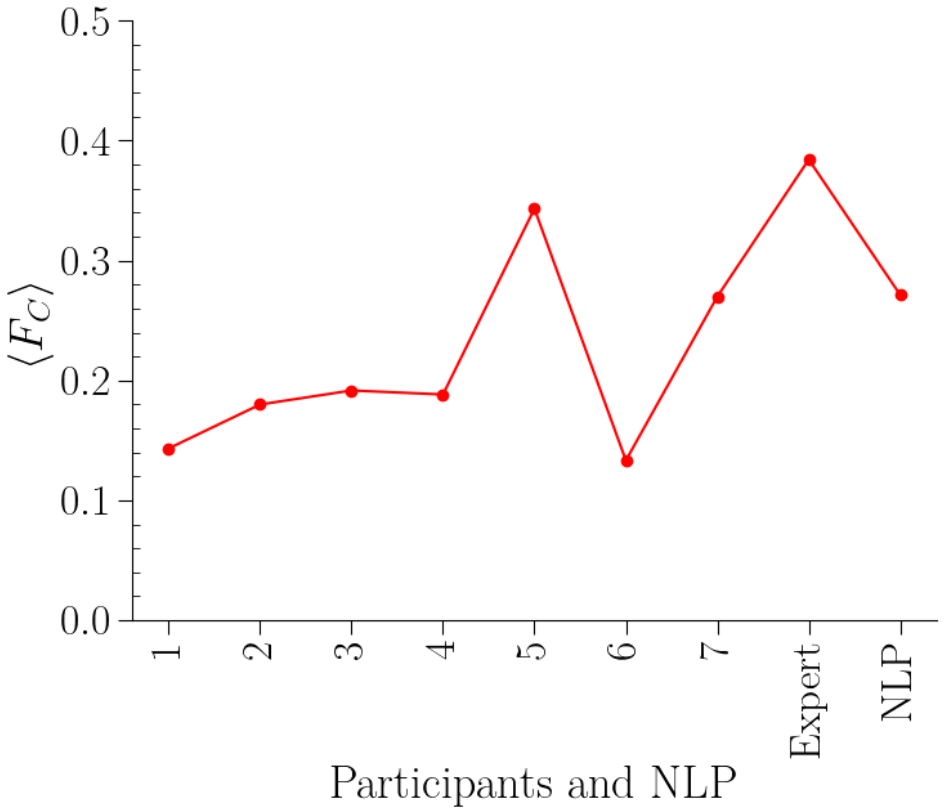


Figure 4.8: Average F_C among the readers ($\langle F_C \rangle$) for each participant and the NLP.

than the average F_C with NLP, which was 0.27. From this, it is possible to conclude that, on average, the readers have a higher agreement with the expert's keypoints than the NLP's summary. The readers often returned summary-style keypoints that contained concepts expressed in the text over several sentences. The NLP chosen for this comparison was modified TextRank. This is an extractive summarisation algorithm and thus cannot return such summary-style keypoints. It is unsurprising then that the participants exhibited higher agreement with the expert than the NLP.

The $\langle F_C \rangle$ was higher for the expert than for any other participant/NLP. Hence, on average, the keypoints generated by the readers have more significant conceptual overlap with the experts' keypoints than any other readeror the NLP. This supports the conclusion from earlier that the readers view concepts, that are also viewed as important by the author, as important by the author. Readers 1 and 6 obtained notably low values of $\langle F_C \rangle$ indicating low levels of conceptual agreement with the other readers. Reader 6 was one of the most experienced in this study, with all of their experience spent exclusively in academia. It may be that this experience meant the

reader interpreted the task very differently from the others.

The $F_{\text{ROUGE-1}}$ scores between the keypoints/summary for every possible pairing of individuals in the group and the NLP is displayed in Figure 4.9. Figure 4.9 shows a significantly narrower

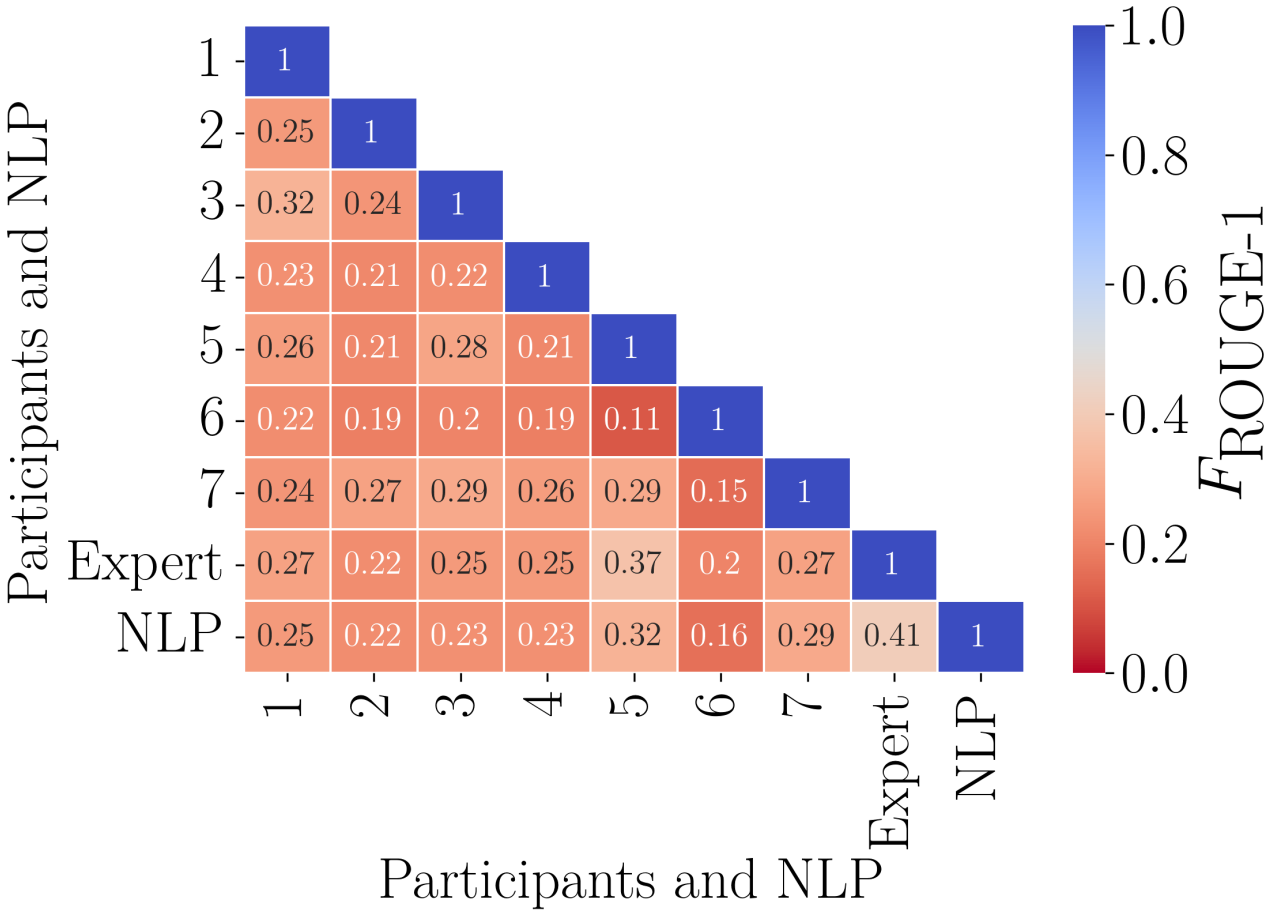


Figure 4.9: $F_{\text{ROUGE-1}}$ overlap of the keypoints/summaries for every possible pairing of individuals in the group containing participants and the NLP.

range of $F_{\text{ROUGE-1}}$ scores than the range of F_C scores displayed in Figure 4.7. Pairings of readers that had low F_C in Figure 4.7 are seen in Figure 4.9 to have high overlap. For example, Readers 1 and 6 have an F_C of 0 and an $F_{\text{ROUGE-1}}$ of 0.22. This supports the conclusion in Section 4.3.1 that a high/low conceptual overlap does not necessarily mean a high/low unigram overlap. The relationship between the two metrics is explored further in Section 4.3.4.

The highest $F_{\text{ROUGE-1}}$ in Figure 4.9 is between the expert and the NLP. As mentioned previously, this may be because the NLP method was optimised using the $F_{\text{ROUGE-1}}$ metric. In addition, the expert is the author of the text, and since the NLP lifts sentences directly from the text, it is unsurprising that $F_{\text{ROUGE-1}}$ is high as the style and word choices are likely to be very similar because the NLP's summary and the expert's keypoints have the same author.

Figure 4.9 shows the expert and Reader 5 had high $F_{\text{ROUGE-1}}$ of 0.37 as well as an F_C of 0.60. Reader 5 was the most experienced, and they also assigned themselves a high expertise rating of 4. Overall, Reader 5's background was the most similar to the experts. This may explain the high overlap between the produced keypoints.

The average $F_{\text{ROUGE-1}}$ among the readers ($\langle F_{\text{ROUGE-1}} \rangle$) was calculated for each of the participants/NLP. The results are displayed in Figure 4.10.

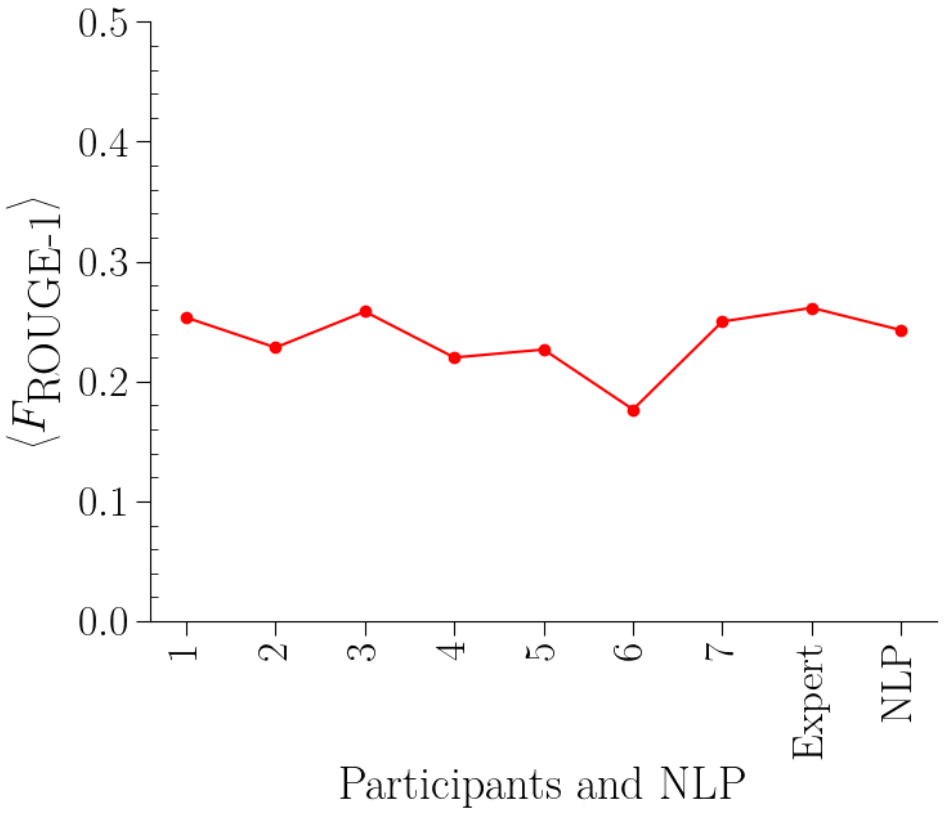


Figure 4.10: Average $F_{\text{ROUGE-1}}$ among the readers ($\langle F_{\text{ROUGE-1}} \rangle$) for each participant and the NLP.

Figure 4.10 shows minimal variation of $\langle F_{\text{ROUGE-1}} \rangle$ for each participant and the NLP. As observed in Figure 4.8, there is a noticeable global minima for Reader 6. This illustrates that, on average, the participants agreed the least with this participant.

The analysis so far has indicated there may be a relationship between the agreement of the readers' $F_{\text{ROUGE-1}}$ and F_C and background factors such as expertise rating. Therefore, the relationship between the difference in expertise level of the readers and $F_C/F_{\text{ROUGE-1}}$ was examined. The difference in expertise level is an ordinal scale; therefore, Kendall's Tau, which

ranges from -1 to 1, was used to establish the strength of the relationship between the two variables. The p-value is the statistical significance of the results. P-values above 0.05 are not statistically significant at a 95% confidence level. For F_C, a value of 0.15 was obtained with a p-value of 0.40. For F_C, a value of 0.24 was obtained with a p-value of 0.18. Both p-values are more significant than 0.05; there exists no statistically significant relationship between the difference of self-assigned expertise ratings and the unigram/conceptual overlap of the generated summaries.

The relationship between the difference in the total number of years of experience and F_C for each pair of readers was examined by calculating Pearson's Correlation, as the total number of years of experience is a continuous variable. A p-value of 0.26 was obtained, indicating no statistically significant correlation. The exact process was repeated using $F_{\text{ROUGE-1}}$. Again, no statistically significant correlation was obtained. This method was applied to the difference in the total number of years of academic experience. Again, no statistically significant correlations were observed.

Other background factors, such as the number of years of commercial/government research, again exhibited no statistically significant correlation. As seen in Section 4.3.2, there is only slight variation in the number of years of commercial/government research among the readers. The lack of statistically significant correlations may be due to the small sample size or the lack of variation in the readers' backgrounds. This suggests future studies should try to recruit a larger, more diverse group of readers.

4.3.4 Conceptual vs unigram overlap

In the previous section it was recommended that the relationship between conceptual overlap F_C and unigram overlap $F_{\text{ROUGE-1}}$ is examined further. In this section, the relationship between the two metrics, F_C and $F_{\text{ROUGE-1}}$, is examined. For every comparison between summaries made in this study, values of F_C and $F_{\text{ROUGE-1}}$ were calculated. These values are plotted in Figure 4.11. A regression line was fitted to this data, and the shaded region on the plots shows bootstrapped 95% confidence intervals generated for the regression line.

The plotted regression line indicates a positive linear relationship between F and $F_{\text{ROUGE-1}}$. This is in agreement with the results found in Tables 4.3 and 4.2. The 95% confidence intervals

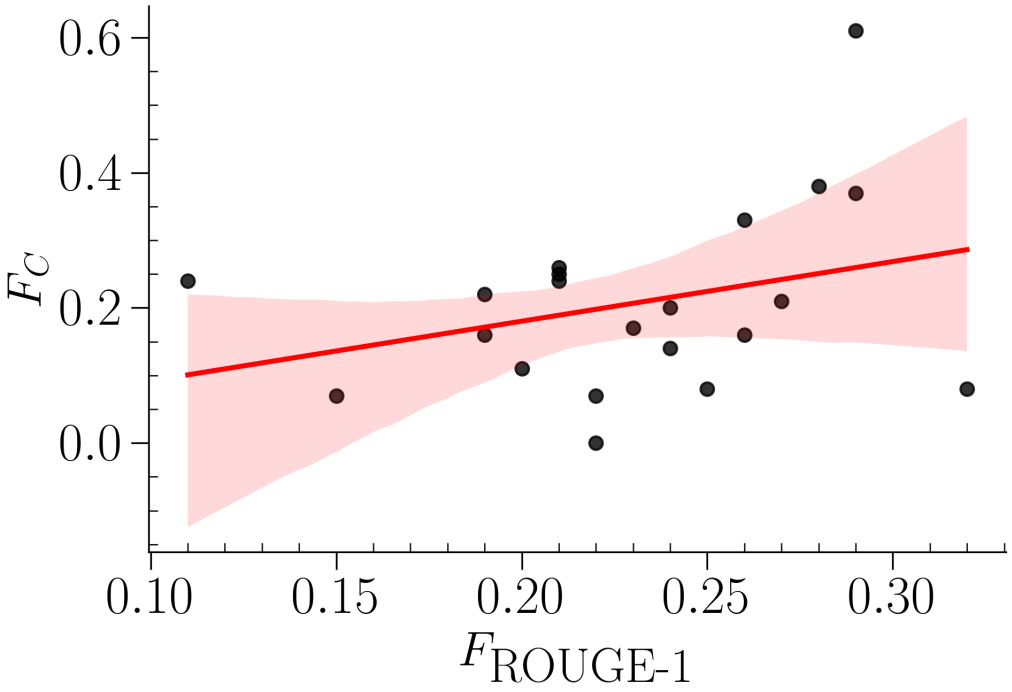


Figure 4.11: F versus $F_{\text{ROUGE-1}}$ for overlap between participants, expert and NLP summaries. A regression line is fitted to the plot and 95% confidence intervals are bootstrapped, represented by the red-shaded region of the graph.

are very large (approximately 0.35), particularly for values of $F_{\text{ROUGE-1}}$ less than 0.20 and greater than 0.3 as the majority of the data is clustered around $F_{\text{ROUGE}-1} = 0.25$. This is because the majority of the data is clustered around $F_{\text{ROUGE-1}}$ values of 0.25, resulting in few data points in the aforementioned regions; hence there are large estimated uncertainties if F was to be estimated from $F_{\text{ROUGE-1}}$ in these regions.

The strength of this relationship was measured using Pearson's correlation. A value of 0.50 was obtained at a significance level of 0.001. According to Ref. [332] this can be interpreted as a moderate-strength linear relationship significant to a confidence level of 99.9%.

Both F_C and $F_{\text{ROUGE-1}}$ are imperfect metrics. Conceptual evaluation F is an inherently biased process, and calculating unigram overlap, as discussed in Section 4.1.7, does not establish if two summaries are discussing the same concept. F and $F_{\text{ROUGE-1}}$ having a positive linear relationship implies these metrics are in agreement to an extent and are generally useful in assessing summaries. This result agrees with other studies that compared ROUGE metrics to human judgement, for instance, Ref. [310]. Whilst useful for analysis to an extent these metrics are imperfect for the purposes of this study. For future studies, the metrics need to be closely considered and outputs aligned to allow cross comparison.

4.4 Conclusion

This chapter presented a preliminary study investigating the boundary between using NLP tools to either supplement or replace human analysis. The ability of NLP tools and domain experts to summarise technical information in a single document in the energetic-materials domain according to a subjective concept (importance) was compared.

In order to identify suitable NLP techniques for use in this study, in Section 4.1, a literature review of the techniques used in all steps of an automatic document-summarisation pipeline was reviewed. The focus of this review was on techniques suitable for data-limited environments. The review covered preprocessing techniques in Section 4.1.1. There, the NLTK Python package was identified as a useful preprocessing tool, and three popular stemming algorithms (Lovins, Paice-Husk and Porter) were discussed. It was recommended to conduct an in-domain comparison of the stemming algorithms. In Sections 4.1.2 - 4.1.6, an overview of summarisation algorithms was given. ML-based abstractive approaches were ruled out due to the absence of readily available large corpora/datasets in the energetic-materials domain. The analysis concluded the following extractive algorithms should be investigated: the frequency-based approach SumBasic, graph-based TextRank, Latent Semantic Analysis (LSA) using the cross method for sentence selection and the state-of-the-art clustering-based summariser 'bert-extractive-summarizer'. The review found that in the case of LSA and TextRank, modifications to the traditional algorithms, such as the use of TF-ISF weightings and Word2Vec, will increase performance. It was recommended to explore such modifications further. In Section 4.1.7, metrics for evaluation were introduced.

In Section 4.2, a detailed overview of the methodology was given. In particular, in Section 4.2.1, a description of the document given to the NLP, readers and expert to analyse the text was given. An overview of the first step in the methodology, preprocessing, was given in Section 4.2.4. A brief comparison of the performance of the Lovins, Paice-Husk and Porter stemming algorithms on the text was conducted. The Porter stemmer performed the best, stemming 95% of words correctly. Section 4.2.2 described the inclusion criteria for readers in this study. An overview of the background-information questionnaire presented to the participants was given. Section 4.2.3 contains a description of the expert. Section 4.2.5 gave an overview of the keypoints task. Here it was described how the expert and readers (collectively referred to as

participants) were asked to list the points they felt were significant in the text. Section 4.2.6 described an overview of the NLP methods used to summarise the text. In Section 4.2.7, two metrics were introduced for intra-group comparison: $RAIS$ (Reader Agreed Important Points Score) and $MRIS$ (Reader Importance Score). These are measures of how important the readers as a group found an agreed point and how important, on average, the readers found an agreed point, respectively. The method for grouping keypoints responses together to form reader-agreed points (RAPs) was described. Methods for inter-group comparison, the conceptual overlap F_C and unigram overlap $F_{\text{ROUGE-1}}$ between two keypoints/summaries, were also stated.

In Section 4.3, the results and discussion are presented. Section 4.3.1 gave an overview of the results of the NLP summarisation algorithms. SumBasic, LSA, TextRank and the bert-extractive-summarizer architecture with (Sci-BERT, GTP-2 and BERT) were compared against the expert's keypoints using $F_{\text{ROUGE-1}}$ and F_C. The best-performing algorithm was TextRank which obtained an F_C and $F_{\text{ROUGE-1}}$ of 0.33 and 0.39, respectively. SumBasic, LSA and TextRank outperformed the state-of-the-art bert-extractive-summariser. This was partly because the embeddings generated from the Sci-BERT, GTP-2 and BERT models were not semantically meaningful in the domain. For future work, it was recommended to explore alternative methods of generating embeddings, for instance, using the S-BERT architecture [301].

Following this, two different modifications of TextRank were explored,'TextRank cosine TF-ISF' and 'TextRank Word2Vec cosine'. These modifications outperformed the original version of TextRank. Despite different methods of feature representation, these methods returned identical sentences and hence yielded identical word and conceptual overlap with the experts' keypoints. However, the 'TextRank Word2Vec cosine' had two significant limitations. Firstly the Word2Vec model used was a general pretrained model. This led to some of the words in 'the text' being out-of-vocabulary, resulting in poor quality sentence embeddings. Secondly, the method of generating sentence embeddings from the Word2Vec model was to average over the word embeddings of the words in each sentence. The effectiveness of this method is disputed in the literature. These limitations highlight that transfer-learning techniques will not enhance the performance of an algorithm in all instances. Future work should investigate other methods of generating sentence embeddings from Word2Vec models, such as using TF-ISF weightings in the averaging process.

Overall, the summaries generated in this comparison lifted sentences directly from the text and could not capture complex concepts given over several sentences. They often lacked cohesion and flow. It was further noted the preprocessing methodology used in this study resulted in errors due to the presence of chemical names and abbreviations. This supports the conclusion in Chapter 3 that a custom preprocessing methodology for the energetic-materials domain is needed.

Section 4.3.2 presents the background information of the 7 recruited readers. The analysis at this stage showed that most readers were not in the early stages of their careers and had substantial academic experience. All readers in this study had either completed or were studying for a PhD. The results indicated that, whilst the readers were fully equipped to complete the keypoints task, recruited readers are from a narrow range of backgrounds. Future studies should endeavour to recruit a more significant number of readers from more varied backgrounds.

In Section 4.3.3, a comparison between the keypoints generated by the expert, the participants and the NLP was conducted. To examine reader intra-group agreement, RAPs were divided into two sets of datasets (1) those the NLP found important and not important and (2) those the expert found important and not important. For each of these sets, unpaired Wilcoxon rank sum tests were conducted on the $RAIS$ and $MRIS$ scores. The results determined the RAPs in agreement with the expert were rated higher on average and found more important by the participants. Similarly, the readers found RAPs in agreement with the NLP more important than those not in agreement with the NLP. Statistical analysis was unable to conclude that the readers rated the RAPs higher on average. Closer inspection of the points that the NLP/expert found important revealed that a significant proportion were either very general statements on the topic of the text or summary-style sentences that state the conclusions of the text. Therefore high reader agreement would be expected.

Analysis of the distribution of $RAIS$ and $MRIS$ showed a heavily skewed distribution that indicated the majority of keypoints were not in agreement with any other readers' keypoints. This is further illustrated in the inter-group analysis, which found considerable variation in the conceptual overlap of summaries; some readers had no conceptual overlap whatsoever. This variation of responses was found in a similar study in Ref. [6]. As pointed out by this study, this variation is unsurprising from a psychological perspective as experts will draw on their experiences when approaching a task, resulting in varied responses.

Inter-group analysis was completed by calculating the F_C and $F_{\text{ROUGE-1}}$ between every possible combination of the readers/experts/NLP keypoints/summaries. On average, the readers exhibited higher conceptual overlap with the expert than with the NLP/other readers. The variation in $F_{\text{ROUGE-1}}$ was significantly less than in F_C. In both cases, the highest overlap with the NLP was the expert. This was unsurprising since the selection of NLP techniques was based on overlap with the expert's summary.

The relationship between different background factors of the readers and the associated F_C and $F_{\text{ROUGE-1}}$ was examined. No relationships were detected. This could be because the sample size is too small or the participants are not diverse enough. This is further evidence to support the conclusion that future studies should try to recruit a larger, more diverse group of participants.

The relationship between the two inter-group comparison metrics, $F_{\text{ROUGE-1}}$ and F_C, was examined in more detail in Section 4.3.2. F_C is an inherently biased metric, and calculating $F_{\text{ROUGE-1}}$, as discussed in Section 4.1.7, does not establish if two summaries are discussing the same concept. In Section 4.3.3, it was observed that pairings of participants could have a high F_C and low $F_{\text{ROUGE-1}}$; the converse was also true. Pearson's correlation was measured between the scores for all readers and a statistically significant value of 0.50 was obtained. It was therefore concluded that there was a moderate-strength relationship between F_C and low $F_{\text{ROUGE-1}}$ indicating the metrics are helpful to some degree.

Overall this study could be improved. The methods of analysing intra-group agreement of the keypoints relied on generating RAPs, the process of grouping together these points is subjective, resulting in biased analysis. Furthermore, inter-group comparison was conducted using metrics that are imperfect for the purposes of this study. There is an inherent conflict in the analysis methods used because the NLP and the participants are essentially performing different tasks. The NLP identifies important sentences in the text, whereas the participants generate succinct summaries of the conceptual ideas in the text in their own words. It is therefore recommended that future studies reframe the keypoints task to bridge the gap between the summarisation task and keypoints task. In addition, this study did not elicit the reasons why the participants found specific points in the text important. Future studies should aim to elicit why participants find specific points in the text important. The agreement among the participant responses should be examined.

The desired outcome from this study was a series of recommendations to guide future studies.

The four key recommendations are summarised below.

- Recruit a larger more diverse group of participants.
- Redefine the keypoints task.
- Elicit why the participants find specific points in the text important.
- Examine participant agreement of the reasons why the participants find specific points in the text important.

Chapter 4 appendices

A Background-information questionnaire

BACKGROUND INFORMATION QUESTIONNAIRE

Participant N.O:

In our pilot study we discovered that there were some quite substantial differences between readers in what they thought was important in a text. We hypothesised that this may be because of differences in people's technical background or professional work experience. For the current study then, we need to record some basic information about your background. This information is not used to judge your answers in any way, but to see which other readers your views are most similar to.

This information will be kept in strict confidence and will not be used to identify you in any report or publication. If you are not happy to fill in any section of this questionnaire please just leave it blank.

1. Knowledge of subject matter:

Please rate the extent to that you agree/disagree with the following statement, "I consider myself an expert in the topic of the effect of electric and magnetic fields on propellent combustion."

Using the scale below, please tick the most appropriate box.

- ☐ 1 = Strongly Disagree
- ☐ 2 = Disagree
- ☐ 3 = Neutral
- ☐ 4 = Agree
- ☐ 5 = Strongly Agree

2. Education:

Please list any higher education qualifications specifying the subject matter and level, on the dotted lines below:

……
……
……
……
……
……

3. Occupation History:

Version: 2. 6/03/2019

Current Occupation, please tick the appropriate box.

- ☐ Academic research/teaching
- ☐ Government research
- ☐ Commercial research
- ☐ Government policy making
- ☐ Other, please specify on the dotted line below.
  ……

Please specify the number of years of experience in the following fields by writing the number of year's experience on the dotted line:

- Academic research/teaching.
  Number of years' experience ……..
- Government research.
  Number of years' experience ……..
- Commercial research.
  Number of years' experience ……..
- Government policy making
  Number of year's experience ……..
- Other, please specify on the dotted line below.
  ……
  Number of years' experience ……..

B Keypoints task questionnaire

KEY POINTS TEMPLATE

Once you have read the report, please fill in the template below to explain briefly in a sentence or two what you thought were the key points made in the text. You can have as many or as few key points as you wish, it is your opinion that counts. We have left room for 25 key points, but you can add table lines for more, or delete (or not fill them in) if you have less. Once you have listed your key points, please rank order the top five points you think were made according to their importance. The importance should be judged by your reading of the text, not your previous experience. For example, it is not the newness of the idea to you personally that is important, just whether it was an important "take away" from your reading of the chapter.

Key point	Summary	Rank Order of importance
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11.		
12		
13		
14		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		

Please indicate your participant number here. My Participant Number is…….

Please return your completed key points to Sinead as soon as possible on

sinead.obrien14@imperial.ac.uk

Thank you very much for your time in completing this template.

Ver 2.0 07/01/2019

C RAIS vs MRIS

Below are some worked examples demonstrating how it is possible for two sets of keypoints to have the same RAIS=6 but possibly different MRIS values.

Example 1: Many readers return related points of low importance.

Reader-agreed point β		
Subpoint in agreement with reader-agreed point	Reader	IS
$\beta.F$	1	1
$\beta.G$	4	1
$\beta.H$	7	1
$\beta.I$	5	1
$\beta.J$	10	1
$\beta.K$	11	1

For the reader-agreed point β the RAIS=1+1+1+1+1+1=6 (Eq. (4.18)) but the MRIS=6/6=1 (Eq. (4.19)).

Example 2: One reader mentions a specific point several times in their keypoint summary.

Reader-agreed point γ		
Subpoint in agreement with reader-agreed point	Reader	IS
$\gamma.L$	1	1
$\gamma.M$	4	2
$\gamma.N$	1	1
$\gamma.0$	1	1
$\gamma.P$	1	1

For the reader-agreed point γ the RAIS=6 but the MRIS=6/2=3.

Example 3: A reader-agreed point is found to be very important by one reader only.

Reader-agreed point δ		
Subpoint in agreement with reader agreed point	Reader	IS
$\delta.Q$	1	6

For the reader-agreed point δ the RAIS=6 and the MRIS=6/1=6.

Example 4: A reader-agreed point is found to be moderately important by several readers.

Reader-agreed point η		
Subpoint in agreement with reader-agreed point	Reader	IS
$\eta.R$	3	2
$\eta.S$	5	2
$\eta.T$	2	2

For the reader-agreed point η the RAIS=6 and the MRIS=6/3=2.

D Out-of-vocabulary words

Words not in pretrained embeddings:
ambipolar, a/m, b=μh, μ, micro-engine, carboxylated, dioctyl, adipate, magnetorheological, jpl-126, lp-33, para-quinone, dioxime, guanadine, vorozhtov, hydrogen-oxygen, ion-producing, thermo-chemical, oxidisers, wt., hydroxyl-terminated, htpb, non-aluminized, non-burning, vyatskii, hres, polymethylmethacrylate, kg/m2-s, microthrusters, hipep, microthruster, electrothermal-chemical, pressure-time.

E Statistical tests

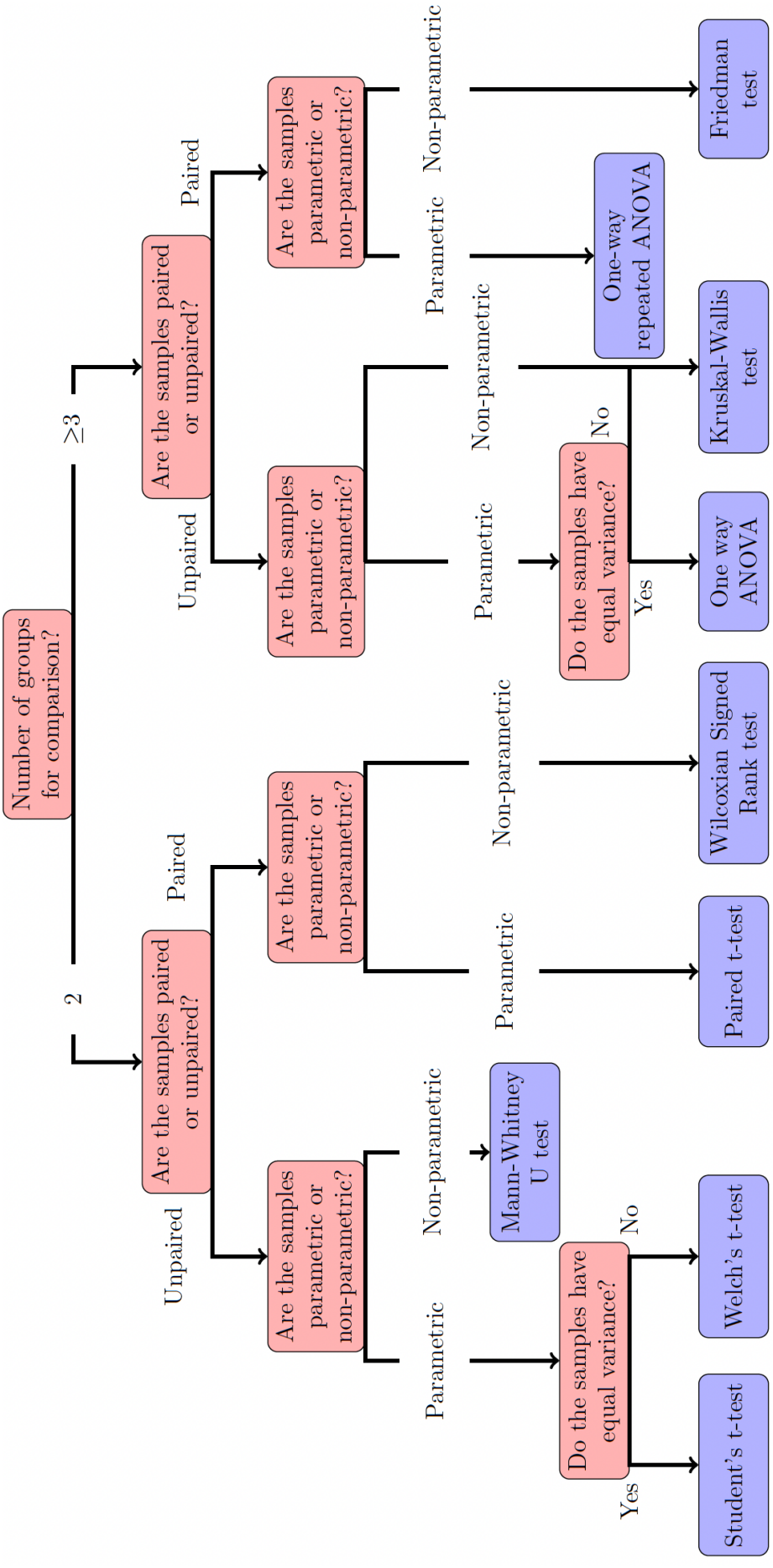


Figure 4.12: Overview of the decision-making process for determining the appropriate statistical test. Statistical tests are shown in blue. It is noted that the Mann-Whitney U test is also referred to as the Wilcoxon Rank Sum test or the Mann-Whitney Wilcoxon test.

Chapter 5

Elicitation of knowledge from a defence expert

In Chapter 4, a preliminary study investigating the boundary between using NLP tools to either supplement or replace human analysis in the domain of energetic materials was conducted. The ability of NLP tools in data-limited environments and domain experts to summarise technical information in a single document in the energetic-materials domain according to a subjective concept (importance) was compared. The outcome of this study was a clear set of recommendations to guide future studies of this nature. One of the recommendations was to elicit why the participants found specific points in the text important. To do this, the expert's reasons must first be elicited. This chapter forms the first part (explicitly focusing on the aforementioned recommendation) of a two-chapter follow-up study to Chapter 4 applying these recommendations. Therefore, this chapter aims to elicit the expert's complex content schema[1] in a confined domain. Specifically, this work focuses on eliciting how the expert evaluates the importance of the different points in the technical document. The content of this chapter largely follows that of my published work in the proceedings of the 2020 Spring International Conference on Defence Technology [1].

Chapter achievements:

- Overview of techniques for elicitation of knowledge.

[1] A schema is a knowledge structure describing how an individual perceives a particular aspect of the world.

- Development of novel methodology for elicitation of expert knowledge.
- First-of-its-kind open-source investigation into elicitation of knowledge in the domain of energetic materials.

In this chapter, a brief overview of elicitation of knowledge techniques will be given in Section 5.1. Then, Section 5.2 will provide an overview of the methodology to capture the expert's understanding of the text. Next, in Section 5.3, the results will be presented, and an analysis of the results will be presented in Section 5.4. Finally, in Section 5.5, results are summarised, and suggestions for further work are made.

5.1 Elicitation of knowledge

One of the challenges in expert-knowledge elicitation has been to access an expert's conceptualisation of their subject area without the knowledge engineer imposing their own view of the field via the elicitation process. In order to achieve this, a variety of open-ended elicitation techniques have been employed. One of the earliest versions was the repertory grid technique, derived from Kelly's personal construct theory [333].

This tool, while popular with knowledge engineers in the early 1990s [334], has subsequently been criticised as a method of knowledge acquisition because it assumes people organise their knowledge along linear dimensions into bi-polar categories [335].

To address the problems of the repertory grid, Canter et al. [336] developed the multiple-sorting method. This well-established method allows a participant to sort elements into categories of their own devising as opposed to using constructs given to them by the interviewer. When categories are presented to the interviewee this can subject the interviewee to bias, since asking to sort elements into categories implies a lot about their expected response. The multiple-sorting method allows the interviewer to elicit the meanings and explanations associated with an individual's categories as well as understanding the organisation of the elements within the categories [337].

No limitations are placed on the sorting process, i.e. the number of categories is undefined. This allows the interviewee to truly conduct the sorting in a way that mirrors their thought

process and allows the researcher to obtain insight into the underlying frameworks by which someone organises their own knowledge [338].

Recently, the multiple-sorting method proposed by Canter et al has been used in conjunction with in-depth interview methods [339]. The in-depth interview methods allow for participants to talk in detail through the reasoning behind their sorting. In combining the in-depth interview approach with the sorting task, more evidence is gathered to establish the validity of the conclusions and results elicited from the interview [340, 341]. In addition, the reasons behind the sorting may be revealed in more detail when done in conjunction with the interview methodology. By allowing the interviewee to talk through the elements separately, they are encouraged to think in detail about each of the elements. This results in a sorting process that has been more considered and hence more likely to be evidenced.

An expert would be expected to have a complex and layered construct system of the domain they are an expert in. Through years of experience in a certain domain, their own construct system would be expected to be refined in a way beyond surface features [342]. As a result, eliciting knowledge from an expert would be expected to be more complex and intensive than eliciting knowledge from a novice, and can therefore be time consuming and expensive [343].

5.2 Capturing expert understanding

Details of the stimulus material and the expert can be found in Sections 4.2.1 and 4.2.3. The process of eliciting the expert's views on the importance of the concepts in this document is a two-stage procedure. Phase one is the keypoints task described in Section 4.2.5. Phase two is a three-part sit-down interview.

On completion of the keypoints task by the expert, an interview with the expert was conducted in order to elicit a more in-depth examination of how the expert views the importance of the keypoints they generated in the keypoints task. Given the subjective nature of the concept of importance, exploring the topic in this in-depth manner was fundamental to generating an understanding of the expert's content schema.

The interview was divided into three main tasks: 1) exploring the importance of each of the expert's keypoints, 2) ranking the keypoints and 3) a sorting task.

Before the interview, each of the expert's keypoints were printed onto identical cards. The interview was conducted in a formal setting around a table. The cards were also labelled A-Q in order to make the recording process easier. Figure 5.1 shows an example of a keypoints card.

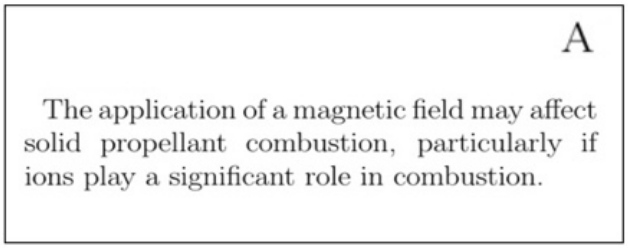


Figure 5.1: Example of a keypoints card. Points were labelled A-Q in the top right-hand corner to aid the recording process.

In the first stage of the interview, the expert was presented with a keypoints card and asked to explain why they thought that particular point to be important. The card was then removed from the table and placed into a pile and the expert was presented with another card and the process was repeated until all of the keypoints had been talked through. The purpose of this activity was to familiarise the expert with the material and encourage them to consider why each of the points were important individually.

On completion of this task, five cards, labelled 1 to 5, were laid out upon the table. The interviewer informed the expert that each of these cards represented five distinct groups, the card labelled 1 being the least important and the card labelled 5 being the most important. The expert was asked to place each of the shuffled keypoints cards into one of the five groups. The aim of this activity was to extract how important the expert views each of the keypoints in the context of the other cards. Using printed cards on a table allowed the expert to visualise the process as they performed it. This task will henceforth be referred to as the 'rating of keypoints'. Figure 5.2 illustrates the set-up.

The final part of the interview was the multiple-sorting task. The method used in this part was an adaptation of Canter et al's. multiple-sorting procedure. The following adaptation from Canter et al. [336] was read to the participant:

"You have indicated that the points you identified might be important for different reasons. Therefore we would like you to think about the points that you generated [hand cards over] and to sort them into groups in such a way that all of the points in any group are similar to each other in terms of the reason why they are important, and different from the points in the other

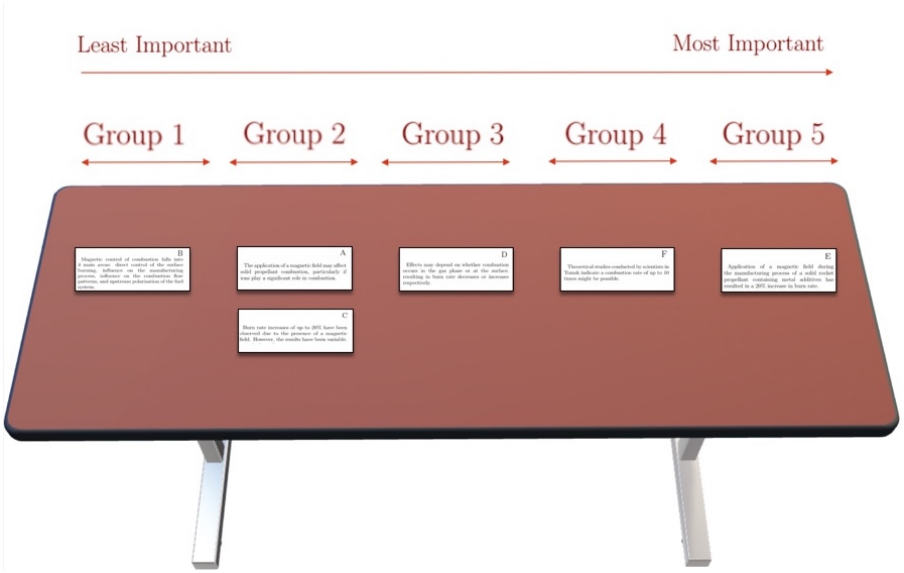


Figure 5.2: Set-up for the rating of the keypoints. Keypoints are placed into one of five categories according to their relative importance.

groups. You can put the points into as many groups as you like and put as many points into each group as you like - it's your views that count. Once you have carried out the sorting, I would like you to tell me the reasons why you put the points into each group and what the points have in common."

The expert was asked to form their own categories of importance, thereby avoiding imposing any preconceptions from the researcher. The number of categories and the number of points in each category was also left to the expert. Figure 5.3 illustrates this set-up.

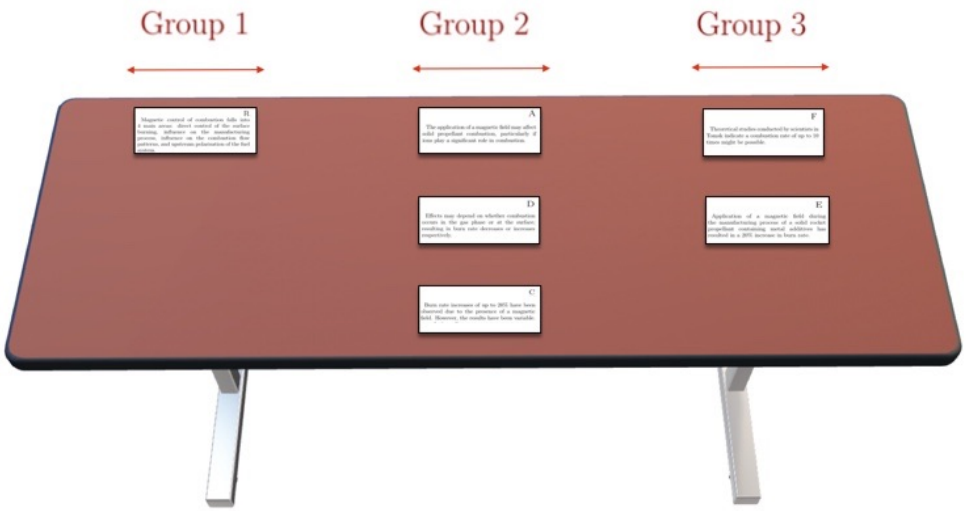


Figure 5.3: Sorting task configuration. Keypoints are sorted into categories.

Once the expert had sorted the points into categories they were asked to explain the reasons

for their classifications. The interview ended with an open-ended question: is there anything else we haven't talked about you think I should know? This gives the expert the opportunity to verbalise anything else not said that they feel is relevant.

5.3 Results

5.3.1 Keypoints task

The results from the expert's keypoints task are shown in Table 5.1. These are the same keypoints used in analysis in Chapter 4. The first column indicates the keypoint the expert found to be important. The second column assigns a label to each of the points which will be used in later analysis. As mentioned in Section 4.2.5, no word limit was given to the required length of each point nor to the total number of points.

Keypoint	Label
The application of a magnetic field may affect solid propellant combustion, particularly if ions play a significant role in combustion.	A
Magnetic control of combustion falls into 4 main areas: direct control of the surface burning, influence on the manufacturing process, influence on the combustion flow patterns, and upstream polarisation of the fuel system.	B
Burn rate increases of up to 20% have been observed due to the presence of a magnetic field. However, the results have been variable.	C
Effects may depend on whether combustion occurs in the gas phase or at the surface, resulting in burn rate decreases or increases respectively.	D
Application of a magnetic field during the manufacturing process of a solid rocket propellant containing metal additives has resulted in a 20% increase in burn rate.	E
Theoretical studies conducted by scientists in Tomsk indicate a combustion rate of up to 10 times might be possible.	F
Ionising additives to seed the propellant are necessary to improve the specific conductance to obtain greater effects.	G
The application of an electric field may affect solid propellant combustion.	H
Electric control of combustion falls into 2 main areas: direct control of the burning surface, and ignition and control of the ablation and combustion of electrically conductive solid propellant.	I
Work conducted at Imperial College by Mayo et al in the 1960s resulted in burn rate increases of up to 200 times and burn rate decreases of up to 10 times when applying voltages up to 30kV.	J
A typical rocket propellant containing ammonium perchlorate, aluminium (with and without) and HTPB with iron added showed a decrease in burn rate in the presence of an electric field, regardless of polarity and the decrease was approximately linear with applied voltage.	K
A combined theoretical and experimental study indicated that burn rate increases in the presence of an electric field were due to current heating of the unburnt propellant whereas burn rate decreases were due to flame electrons inhibiting combustion.	L
The PMMA flame is known to contain surplus positive particles. In the presence of an electric field, these can be driven towards the burning surface thereby increasing the combustion rate. Up to 30% increases were observed using voltages up to 35kV and electric fields of up to 266V/m.	M
Electric solid propellants (ESPs) are a relatively new development and have been used in microthrusters. More recently they have been used in electrothermal-chemical (ETC) guns.	N
In an ESP/ETC gun, electrodes are an integral part of the charge.	O
Burn rate enhancements of a solid propellant, similar to JA2 but with metal additives to increase the electrical conductivity, in an ESP/ETC gun have been achieved using as little as 500V. Multiple discharges were possible, achieving a flattened pressure-time profile.	P
Apart from one paper in 2015, no other papers on ESP/ETC have been found.	Q

Table 5.1: Expert's keypoints, assigned importance rating and a label assigned to each keypoint.

As illustrated by Table 5.1, the expert produced a total of 17 points of varying word length.

5.3.2 Interview

As mentioned in Section 5.2, the interview consisted of 3 main components. The first of these asked the expert to explain why each of the keypoints (detailed in Table 5.1) are important. No specific definition of importance was given to the expert, as the purpose of this interview is to elicit how the author evaluates the importance of different concepts in the document.

From a transcript of the interview, a summary of why the expert found each point to be important was produced. Details of the results are given in Section 5.4.

An important note about point N: in the context of the interview, the expert discussed the point as being part of a wider experiment talked about in the document. This is not something that may be understood from viewing the point on its own. The expert rated the point in this wider context. The point will henceforth be considered in this context.

The results of the rating task are displayed in Table 5.2.

1	2	3	4	5
A	B	E	J	N
F	C	G	L	P
	D	K	M	
	Q	H	O	
		I		

Table 5.2: Keypoints sorted into five categories according to importance, 1 being the least important and 5 being the most important.

Once the sorting was completed, the expert was asked to explain the reasoning behind their sorting. The expert described the most-important points (placed in Group 5) as those more application-orientated. These points looked specifically at how enhancing the combustion rate through the use of electromagnetic fields can be applied in a real application as opposed to considering enhancement of combustion in a theoretical or specialised environment. The point P was said to be the most important as this actually demonstrated how enhancing the combustion process through the use of electric and magnetic can be effectively applied. The expert also remarked that the points they considered the least important were those concerning magnetic

fields. The reasoning given to this was they felt that studies in this field were not as consistent or reproducible in an application.

The expert further noted points such as F were placed in lower-importance categories as they concerned theoretical studies conducted under specialised circumstances, which would not be directly possible to replicate in an application. The author remarked that studies that were experimental were seen as higher value that non-experimental studies.

The expert was then asked to complete the open-ended sorting task. The expert sorted the keypoints into four distinct categories of their own devising. These categories are defined as follows:

- Group α is defined as keypoints that are ESP-gun related.
- Group β is defined as results which quantify the effect of electric fields on the increase in burn rate.
- Group γ is defined as quantifying the effects in terms of the increase and decrease in burn rates seen for magnetic fields.
- Group δ is defined as keypoints that describe the physics and chemistry of the processes occurring.

The results of this sort are displayed in Table 5.3.

Group α	Group β	Group γ	Group δ
O	H	A	B
Q	J	E	I
N	K	F	D
P	M	C	L
		G	

Table 5.3: Keypoints sorted into four distinct categories generated by the expert.

The expert had no comments in response to the open-ended question asked at the end of the interview.

5.4 Analysis

5.4.1 Collective observations

As mentioned in Section 5.2, once the sorting was completed the expert was asked to explain the reasoning behind their sorting. It was clear the expert viewed the most-important category as the most application-orientated and the least important as that viewed to be of less interest in terms of developing an application. It is clear there is a link between how important the expert views the topic to be and its applicability.

In the process of talking through the keypoints, it was revealed the expert used past knowledge to direct their analysis. For example, when discussing why point F was important, the expert explained that from previous work they were aware of good-quality theoretical work by scientists in Tomsk within the domain of enhancement of combustion rate using electromagnetic fields [344, 345]. It is for this reason the expert looked at the work completed by these scientists for this report. A similar point was made when discussing point J and the work of Weinberg and Mayo [346]. Similar remarks were made in a study in Ref. [347] which remarked that, when evaluating academic texts, the physicists would often use the reputation of the author to guide text selection.

In addition, the expert described the approach to writing the document. It was described how the expert divided the literature on the topic into categories and subcategories based on the physical categories and the applications within the physical categories. The categories were magnetic and electric fields and the subcategories were direct control of the surface burning; influence on the manufacturing process; influence on the combustion flow patterns; upstream polarisation of the fuel system and direct control of the burning surface; and ignition and control of the ablation; and combustion of electrically conductive solid propellant. It is of note that in Section 5.3.2, the categories devised by the expert in the sorting task were labelled in a similar manner: ‘quantification of the effect of electric fields on the increases in burn rate’ and ‘quantification of the effect of magnetic fields on burn rate’. It is significant that, of all the subcategories of literature mentioned in the text, only one specific subcategory was used in the experts own categorisation in this interview, ‘ETC-gun’ related. This perhaps further backs up the idea that ETC guns were considered to be the most-important application of this work.

It is of interest that the sorting task uncovered the way the expert structured the process of writing the document.

5.4.2 Independent reasoning categories

The first stage of the analysis draws upon the open-ended descriptions of the keypoints the expert generated in the interview with the researcher. The transcript of the interview was analysed in order to generate a series of conceptual categories that represent the range of explanations given by the expert.

These categories will be referred to as 'independent reasoning categories'. These categories are not necessarily mutually exclusive since it is possible for a point to be in more than one category. The results of this analysis are shown in Table 5.4.

The first category of importance identifies those keypoints that were said to be important because they were a key piece of theoretical information vital to understanding the scientific processes in the document. 'Theoretical importance' applied to points A, G, H and O. In the interview process, the expert explained the theoretical value of these points.

The expert explained that when the literature was surveyed they divided the reviewed documents up into two separate categories according to physical concept: papers discussing the effect of magnetic fields and papers discussing the effect of electric fields. Within these two categories, the literature was further divided up into four and two subcategories respectively. Points B and I detail these subcategories explicitly. The expert explained these points were important because they describe an overview of the topics covered in the paper. These points were therefore categorised as being a 'Domain overview'.

Transcript analysis determined that keypoints N and P should be grouped together as they demonstrate a direct use of increasing the combustion rate with an electric field in an application: ETC guns. They therefore formed the category: 'Demonstrated use in an application'.

Point E details how burn rate increases were achieved by treating rocket fuel propellants with magnetic fields in the manufacturing stage. The expert explained this point is important because it illustrates how burn rate increases can be achieved using a method that is practical in a commercial environment. A similar point was made when describing why point K was

Category	Definition	Keypoints
Theoretical importance	A key piece of theoretical information vital to understanding the scientific processes in the document.	A, D, G, H, L, O
Domain overview	This is important because it provides an overview of what is talked about in the document.	B, I
Demonstrated use in an application	This is important as it is a validated example of the work being used in an application.	N, P
Practicality of methodology	The methodology described has applications.	E, K
Results show clear quantification of effects	The results shown demonstrate a clear quantification of the effects.	F, J, K, M
Advancement of the field	This is important because it is key to understanding how the field has evolved and/or refined itself.	C, Q
Novelty	This is important because it has not been done before..	M

Table 5.4: Generalised categories of importance, the corresponding definition of each category and the points sorted into each category.

important. The expert specifically highlighted that the methodology described in this point was polarity independent. The fact that the method used to achieve an increase in burn rate was polarity independent could make it more feasible to be implemented in a practical application. As a result, points E and K were added to the 'Practicality of methodology' category.

When discussing point K, the expert gave another reason why point K was important. Point K states the decrease in the burn-rate was linear with applied voltage; the expert said this clear quantification of the effects observed was important as it would be easier to implement in a practical environment. Having a clear quantification of the observed effects would allow for prediction of performance enhancement in an application. For instance, if the work was implemented in a tank gun system one would be able to predict increases in the velocity, range and hence lethality of the gun. Similar reasons were given to explaining the importance of points J and F. These points were hence grouped together into a category 'Results show clear quantification of effects'.

It is noted the expert made a clear distinction that the results for F were obtained through theoretical analysis only. At various points throughout the interview, the expert made several

references stating that theoretical work was thought to be less important that experimental. Given this, point F could arguably be in its own subcategory within the larger category of 'Results show clear quantification of effects'.

Point M was placed into a category labelled 'Novelty'. Point M details a method of achieving the desired control of burn rate in an electric field through the use of positive (as opposed to negative) species. The expert found this to be significant as it was able to achieve the desired physical effect in a new and distinct way from the other methods described in the paper.

Point C was placed into the category of 'advancement of the field'. The expert explained this point was important because it was one of the first studies in the area and although the results were not necessarily well understood, they did show some burn rate increase. It was highlighted this result formed the basis of future work.

Point Q was also placed in this category. Point Q describes how only one paper on ESP/ETC guns was found. The expert used their expert understanding of the field to explain why this is of significance. From experience the expert knew that if publication was halted it could mean either adverse results may have been obtained leading to a cessation of the work, or that work might have been very successful in which case further work became classified.

In Section 5.3.2, the expert generated four distinct categories of importance. These categories will be explored in more detail by considering the independent-reasoning categories assigned to each of the keypoints in these expert-generated categories. Table 5.5 summarises the categories generated by the expert alongside the reasoning categories assigned to each point in each category.

The starting point of this analysis was to examine Group δ. This group was specified as points that were important because they described the physics or chemistry of the processes occurring. Using the independent-reasoning categories, it is clear this group is subsequently divided into two further separate categories: Theoretical importance and Domain overview.

Aside from Group δ there seems to be no relationship between the groups and the reasons of importance between them. This implies the expert has generated the Groups α, β and γ purely on the overall physical effect being described by the points as opposed to anything else.

Group α	Group β	Group γ	Group δ
Theoretical importance	Theoretical importance	Theoretical importance	Domain overview
Advancement of the field	Results show clear quantification of effects	Practicality of methodology	Domain overview
Demonstrated use in an application	Practicality of Methodology and Results show clear quantification of effects	Results show clear quantification of effects	Theoretical importance
Demonstrated use in an application	Novelty	Advancement of the field	Theoretical importance
		Theoretical importance	

Table 5.5: Categories generated by the expert with the points replaced by their corresponding reasoning categories.

5.4.3 Rating comparison with summary ratings

A direct comparison of the expert's keypoint ratings with the output of the rating task conducted in the interview was performed. Table 5.6 displays a comparison of the five highest ranking keypoints in the keypoints exercise with the corresponding category the expert placed them in during the interview.

Keypoint	Rating	Importance category
Electric control of combustion falls into 2 main areas: direct control of the burning surface, and ignition and control of the ablation and combustion of electrically conductive solid propellant.	5	3
Magnetic control of combustion falls into 4 main areas: direct control of the surface burning, influence on the manufacturing process, influence on the combustion flow patterns, and upstream polarisation of the fuel system.	4	2
Work conducted at Imperial College by Mayo et al in the 1960s resulted in burn rate increases of up to 200 times and burn rate decreases of up to 10 times when applying voltages up to 30kV.	3	2
Burn rate enhancements of a solid propellant, similar to JA2 but with metal additives to increase the electrical conductivity, in an ESP/ETC gun have been achieved using as little as 500V. Multiple discharges were possible, achieving a flattened pressure-time profile.	2	5 (Most important)
The PMMA flame is known to contain surplus positive particles. In the presence of an electric field, these can be driven towards the burning surface thereby increasing the combustion rate. Up to 30% increases were observed using voltages up to 35kV and electric fields of up to 266V/m.	1	4

Table 5.6: Expert's top-rated points alongside the importance category assigned to each point in the interview task.

The rating task is compared to ranking completed at the time of the keypoints task. It is clear from the table that several of the highly rated points are highly rated in the keypoints task. This indicates consistency in the expert's evaluation.

Only two of the five top-rated keypoints appear in the top two most-important categories in the rating task completed in the interview. The top two rated points in the keypoints task (B and I) were said to be important because they were helpful to understanding the document as they surveyed the literature reviewed (see Table 5.4). It is possible the process of talking through the points individually in terms of their importance may have encouraged the author to rethink his evaluation of importance of these two points. This may also explain why only one of the two points in the most-important category in Table 5.2 appeared in the top 5 most-important points in the keypoints task.

5.5 Conclusion

The aim of the work completed in this chapter was to elicit the author's complex content schema in a confined domain. The work specifically focused on eliciting how the expert evaluates the importance of the different points in the technical document.

The expert's views on the importance of the concepts in this document were elicited in two phases. In the first phase, the expert was asked to summarise an eight-page document on the effects of electromagnetic fields on propellant combustion. Completion of this task generated a series of 'keypoints'. The expert was asked to rank the five most-important points. Phase two of the methodology was a sit-down interview with the expert. This interview was comprised of three parts: asking the expert to talk through why each of the keypoints were important; asking the expert to sort the keypoints into categories according to how important they are; and then asking the expert to generate categories of why the points are important.

The techniques used for expert elicitation proved highly successful in relation to this domain of knowledge. Not only were the procedures able to extract the underlying categories through which the expert structured their understanding of the field, but the results indicated reliability in the content of knowledge extracted through different methods.

For example, the different extraction techniques employed consistently revealed the prominence of applications in the way the expert defined the importance of the material contained in the document. Analysis in Sections 5.4.1 and 5.4.3 showed the expert viewed the most-important category, α, as the one has the most demonstrated use in an application. The most-important points in the rating task were found to be part of the same category.

In addition, in Section 5.2 it was shown there were various factors considered by the expert when assigning importance in terms of use in an application: the practicality of the methodology; whether or not use has been demonstrated in an application; and if the work in question has been able to quantify the effects in a way that would be useful for applications.

It is clear from this work the expert has an advanced and complex content schema in this domain. Whilst the sorting appeared to be picking up on basic surface features, the more in-depth interview revealed a complex and detailed level of understanding typically associated with experts. There were several points which used clear indications of expert knowledge, the

first being the identification of what features in a methodology make it suitable for a practical application. During the interview the author noted low voltages, for example, were a feature necessary to achieve adaptation to an application. A further example of this is shown in the fact the author kept mentioning that having a clear quantification of the effects observed was important as this would lead to use in an appropriate application. This expert insight is not something that would necessarily be known to a novice.

There has been much work on the comparison between how novices and experts interpret academic text [327, 348, 349]. Chapter 6 will apply the recommendations in Chapter 4 to develop a new questionnaire to be distributed to a broader range of participants, including novices. This questionnaire contains an updated keypoints task and a keypoints rating exercise that asks participants to assign keypoints a category of importance elicited from this study.

Chapter 6

Comparing NLP tools', experts' and novices' analysis of technical text

Chapters 4 and 5 presented a preliminary study and the first part of a follow-up study investigating the boundary between using NLP tools to either supplement or replace human analysis in the energetic-materials domain. These studies focus on the information in a single document, referred to as 'the text'. This chapter presents the second part of the follow-up study. Recommendations from Chapter 4 and insights from Chapter 5 will be applied to address the objectives of this chapter: (1) conduct a comparison of the ability of NLP tools, novices and experts to identify important points in the text, and (2) assess the agreement amongst experts and novices on how they evaluate the importance of technical information in the text.

Chapter achievements:

- Overview of inter-rater reliability metrics.
- Novel methodology for comparing the ability of NLP tools, experts and novices to identify important points in a single document.
- Adapted methodology for comparing how experts and novices evaluate the importance of technical information.

In Section 6.1, the reader will be introduced to inter-rater reliability metrics, a useful tool in psychology for assessing agreement. In Section 6.2, an overview of the methodology used in

this study will be given. In Section 6.3, results will be presented and discussed. In Section 6.4, the conclusion to this chapter and recommendations for further work will be stated.

6.1 Inter-rater reliability

One of the objectives of this chapter is to assess the agreement amongst experts and novices on how they evaluate the importance of technical information in the text. Therefore, this section will present an overview of agreement metrics.

Understanding the extent to which a group of participants agree is a common problem in psychology. Inter-rater reliability (IRR) metrics are a useful tool that measure how well raters can consistently distinguish between items on a measurement scale, examining the trend in the ratings as opposed to the absolute value [350]. Inter-rater agreement (IRA) measures the absolute differences in the ratings amongst raters. Despite the terms being used interchangeably on a measurement-scale, IRR and IRA do not refer to the same measure [351]. However, on a nominal scale IRR and IRA can be used interchangeably. In this thesis, IRR metrics will be interpreted as the extent to which participants agree.

Cohen's kappa, κ, is a measure of the observed agreement between two raters of a binary-categorical variable. This metric accounts for the probability of chance agreement [352, 353]. This is one of the most simplistic and widely cited agreement metrics. Cohen's kappa can be better understood through a contingency table. Considering the most simplistic example where raters are asked to rate items on a binary scale (Outcome A and Outcome B), Table 6.1 represents the possible outcomes for two raters (Rater 1 and Rater 2) on n samples [354]. Table 6.1 shows the frequencies of a particular combination of outcomes. For instance, $n_{1_A 2_A}$ is the number of outcomes where Rater 1 and Rater 2 both rated a sample with Outcome A.

		Rater 1		Total
		Outcome A	Outcome B	
Rater 2	Outcome A	$n_{1_A 2_A}$	$n_{1_B 2_A}$	R_{2A}
	Outcome B	$n_{1_A 2_B}$	$n_{1_B 2_B}$	R_{2B}
		R_{1A}	R_{1B}	n

Table 6.1: Contingency table for two raters: 1 and 2 on a binary scale (Outcome A and Outcome B).

Using the contingency table, 6.1, the percentage agreement observed is defined as

$$p_o = \frac{n_{1_A 2_A} + n_{1_B 2_B}}{n}. \tag{6.1}$$

The hypothetical probability of a chance agreement between two raters is defined as

$$p_c = \frac{R_{1A} R_{2A} + R_{1B} R_{2B}}{n^2}. \tag{6.2}$$

Hence,

$$\kappa = \frac{p_o - p_c}{1 - p_c}, \tag{6.3}$$

where κ ranges from -1 to 1, with 1 denoting perfect agreement, -1 denoting perfect disagreement and 0 denoting an agreement no better than pure chance [352, 354]. The literature contains a range of interpretations of κ. Landis and Koch's interpretation is one popular approach. This approach, referred to as the Landis and Koch benchmark scale, is summarised in Table 6.2. Other popular interpretations of κ can be found in Refs. [356, 357]. The disparity between

κ	Interpretation
$\kappa \leq 0$	poor
$0 < \kappa \leq 0.2$	slight
$0.2 < \kappa \leq 0.4$	fair
$0.4 < \kappa \leq 0.6$	moderate
$0.6 < \kappa \leq 0.8$	substantial
$0.8 < \kappa \leq 1$	almost perfect

Table 6.2: Interpretation of Cohen's kappa according to the Landis and Koch benchmark scale [355, 353].

approaches to the interpretation of κ can be problematic, as highlighted in Ref. [358]. Fleiss's kappa [359] is an extension of κ designed to include two or more categories/raters.

Krippendorff's alpha, α_k is a highly flexible IRR metric that can deal with missing data, any number of raters, categorical/ordinal/continuous data, and large/small sample sizes [360]. Calculating α_k is complex and based on disagreement between raters, i.e.

$$\alpha_k = 1 - \frac{D_o}{D_e}, \tag{6.4}$$

where D_o is the observed disagreement and D_e is the disagreement by chance. Further details on the calculation of α_k can be found in Ref. [361]. As with κ there are a range of different

interpretations of α_k; for instance, see Refs.[355, 362]. Krippendorff [363] suggests the following interpretation of α_k: $\alpha_k \geq 0.80$ for conclusive statements to be made, $\alpha_k \geq 0.67$ for tentative statements and $\alpha_k < 0.67$ for no agreement.

Despite the popularity of κ and α_k, these metrics have major limitations. Ref. [358] provides a summary of these limitations. The two most common pitfalls of κ and α_k are the kappa paradoxes where a high agreement is observed, but low kappa values are obtained. The paradoxes are summmarised as follows:

- **Bias: the frequency with which the raters choose a particular category.**
  'The first paradox of κ is that if p_c (percentage chance) agreement is large, the correction process can convert a relatively high value of p_o into a relatively low value of κ' [364, 365].
  This can be observed by comparing the distribution of R_{1A} R_{1B} and R_{2A} R_{2B}. If the distributions are the same, the marginals are symmetric, and if they are opposite then the marginals are asymmetric.

- **Prevalence: the probability a user classifies an object A or B.**
  'The second paradox occurs when unbalanced marginal totals produce higher values of κ than more balanced totals' [364, 365].
  This can be observed by calculating the proportion of total samples in R_{1A} and R_{2A}. If this equals 0.5 then the marginals are balanced.

To summarise, when the values of R_{1A}, R_{2A}, R_{1B} and R_{2B} (the marginals) are unbalanced it is better to use another statistic such as Gwet's AC_1. Gwet's AC_1, denoted as $\hat{\kappa}_G$, was developed to overcome these paradoxes [366]. $\hat{\kappa}_G$ is defined as

$$\hat{\kappa}_G = \frac{p_o - p_c}{1 - p_c}, \tag{6.5}$$

where

$$p_c = 2\hat{\pi}_1(1 - \hat{\pi}_1), \tag{6.6}$$

and

$$\hat{\pi}_1 = \frac{R_{1A} + R_{2B}}{2N}. \tag{6.7}$$

Whilst $\hat{\kappa}_G$ is argued to be paradox proof, some studies have shown it is still affected by unbalanced marginals to a lesser extent [358, 367].

The metric $\hat{\kappa}_G$ can be interpreted using standard benchmark scales such as the Landis and Koch scale in Table 6.2 [366]. In Ref. [366], Gwet provides an alternative method of how to interpret values of $\hat{\kappa}_G$: $\hat{\kappa}_G$ is a point estimate with a probability distribution and a standard error. This distribution and error is influenced by the number of raters and rating categories in the study. Given this, Gwet proposed the following statistical approach for obtaining benchmarks for interpreting $\hat{\kappa}_G$:

1. Select a benchmark scale. Using the calculated $\hat{\kappa}_G$ and its standard error, calculate the probability for $\hat{\kappa}_G$ to fall into each of the benchmark intervals. These probabilities can be referred to as interval membership probabilities [368].

2. Calculate the cumulative probability, starting from the highest benchmark level, i.e. find the running total of the interval membership probabilities [368].

3. Find the first benchmark interval for which the cumulative probability is higher than some prespecified value, e.g. 95% [368].

In summary, Cohen's kappa is a simplistic metric that can be used to measure observed agreement between two raters on a binary scale. Fleiss's kappa is an extension of Cohen's kappa to two or more categories/raters. Krippendorff's alpha is a highly flexible metric that, unlike the metrics mentioned above, can deal with missing data, categorical, ordinal and continuous data and is effective for small/large sample sizes. However, all three of these metrics suffer from the 'kappa paradoxes' meaning, in some cases, a low kappa value can be returned for high agreement. Gwet's $AC1$ is another flexible coefficient developed to overcome these limitations. IRR metrics are usually interpreted using benchmark scales such as the Landis and Koch scale.

6.2 Methodology

As stated in the introduction to this chapter, the objectives of this study were to (1) conduct a comparison of the ability of NLP tools, novices, and experts to identify important points in the text and (2) assess the agreement amongst experts and novices on how they evaluate the

importance of technical information in the text. The text is an eight-page technical report on the effect of electric and magnetic fields on propellant combustion, further details on the text can be found in Section 4.2.1. To achieve the objectives mentioned above, a group of readers were recruited for a follow-up study and asked to fill out a three-part online questionnaire. The same questionnaire was also presented to the author of the text (the expert). Details of the expert can be found in Section 4.2.3. As in previous chapters, the readers and expert will be collectively referred to as 'participants'. Approval from Imperial College's ethics committee was granted for this study.

A total of 25 readers were recruited voluntarily. Following recommendations from Chapters 4 and 5 to recruit a more diverse group of readers for the follow-up study, 2 of the 25 that were recruited were considered 'novice readers'. The inclusion criteria for the 'novice readers' was they must not have a background in STEM. The remainder of the readers had the inclusion criteria of a STEM background. The suitability of a particular candidate for this study was established via the first part of the online questionnaire, a set of background-information questions. These background-information questions collected information on current occupation, higher-education qualifications and work history. Participants who filled out the questionnaire had the option to remain anonymous[1]. The questionnaire was an online version of the background-information questionnaire used in the previous study (see Chapter 4 Appendix A).

The second part of the questionnaire addresses objective (1). Chapter 4 compares NLP tools and domain experts on their ability to summarise information in a technical document. As described in Section 4.4, the Chapter 4 study could be improved by bridging the gap between the tasks performed by NLP and the participants. As a result, in this study, the participants were asked to perform **exactly** the same task as the NLP: identifying the important sentences in the text. Figure 6.1 is a flow chart illustrating the methodology used to address objective (1).

The first step in this methodology was to preprocess the text following the methodology described in Section 4.2.4. This generates two forms of text, one suitable for further NLP processing and one suitable for human processing. Next, the participants were asked to download the text, read and digest the information, before highlighting the most-important sentences in the

[1]Participants were given the option to remain anonymous in order to attract more participants from defence and security-related fields.

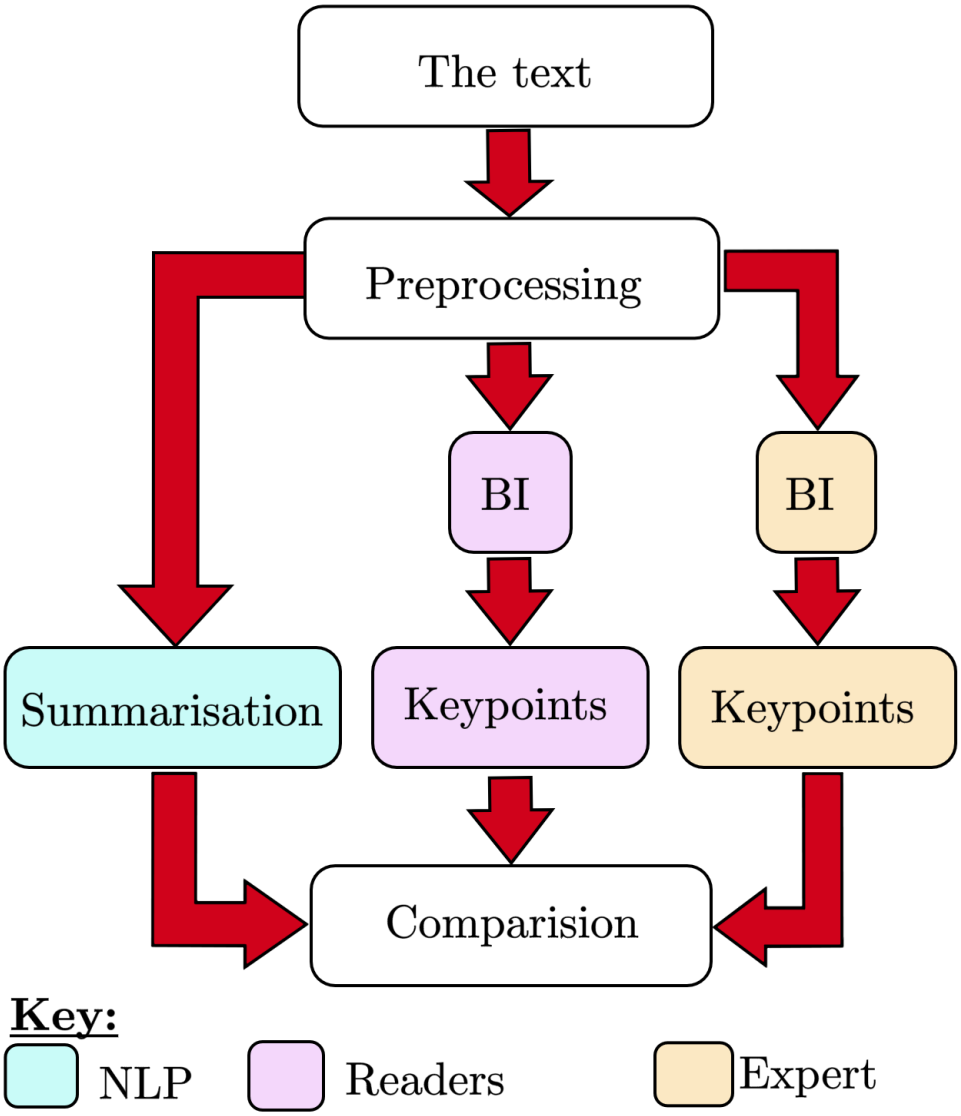


Figure 6.1: Tasks completed by the NLP/participants/expert are shown in turquoise/lilac/orange respectively. First, the preprocessing methodology from Chapter 4 is applied to the text. The participants are asked to complete a three-part online questionnaire. Only the first two parts of the questionnaire are relevant to this method. Part 1 asks the participants to answer background-information questions. Part 2 is a summarisation exercise. Next, the best-performing NLP summarisation algorithm from Chapter 4 is used to generate a summary. Finally, a comparison of the generated summaries is conducted.

document. As with the previous study, the definition of importance was left up to the participants' interpretation. To prevent the participants from highlighting the entire document, they were given the option of highlighting up to 25 sentences. This task will henceforth be referred to as the 'summarisation task'.

Using the best-performing NLP summarisation algorithm from Chapter 4 (see Section 4.3.1 for

a comparison of different NLP summarisation algorithms), 'modified TextRank', a summary of 25 sentences was generated. Finally, a comparison between the NLP's and the participant's summaries was conducted. In order to conduct this comparison, two sets of metrics were used: those that measure intra-group agreement and those that measure inter-group agreement. To measure intra-group agreement in Chapter 4, the metric $RAIS$ was introduced (Eq. (4.18)). This metric is modified for this study to $RAIS'$. $RAIS'_i$ for a sentence indexed as i is merely the of readers who included the sentence in their summaries. $RAIS'$ is a measure of how important the readers found each sentence in the text collectively. Unlike $RAIS$, which relied on amalgamating reader responses to generate reader-agreed points, an inherently subjective process, $RAIS'$ does not introduce any bias.

In Chapter 4, an inter-group comparison was conducted using conceptual F_C, and uni-gram $F_{\text{ROUGE-1}}$ overlap (see Eq. (4.20) and Eq. (4.16) respectively). These metrics were either inherently flawed or subjective. The modification of the summrisation task in this study allows these two metrics to be replaced with a single non-subjective metric, F_A. The metric F_A is a modification of F_C. The overlap F_A between two summaries i and j is defined as

$$F_{A_{ij}} = \frac{2P_{A_{ij}}R_{A_{ij}}}{P_{A_{ij}} + R_{A_{ij}}}, \tag{6.8}$$

where

$$P_{A_{ij}} = \frac{S_{ij}}{N_i}, \tag{6.9}$$

and

$$R_{A_{ij}} = \frac{S_{ji}}{N_j}, \tag{6.10}$$

where S_{ij} is the number of sentences identified as important both in summaries i and j and N_j is the number of sentences identified as important in summary j.

The average overlap of a summary i with all other readers' summaries, $\langle F_A \rangle'_{R'_i}$, is defined as

$$\langle F_A \rangle'_{R'_i} = \frac{1}{M}\sum_{m=1}^{M} F_{A_{im}}, \tag{6.11}$$

where $m \in \{1, \ldots, M\}$ is an element in the set of M reader summaries excluding summary i.

The third part of the online questionnaire addresses objective (2). This section of the question-

naire is referred to as the statement-analysis task. The methodology here is similar to that of Ref. [6] where a group of participants rated the importance of a series of statements derived from responses to a keypoints task on a text reviewing terrorism hostage-taking. This task asked the participants to analyse a series of 30 statements derived from the keypoints/summaries generated by the NLP/participants in Chapter 4. Of these statements, 9 originated from the expert, 9 from the NLP and 11 from the readers.

The first component of the statement-analysis task was a rating task aiming to establish if the participants agreed on how important they found the statements. Participants were asked to assign each of the statements a 1-5 rating of importance using a drop-down menu, where 5 is very important, and 1 is not at all important. An example of the rating task is shown in Figure 6.2.

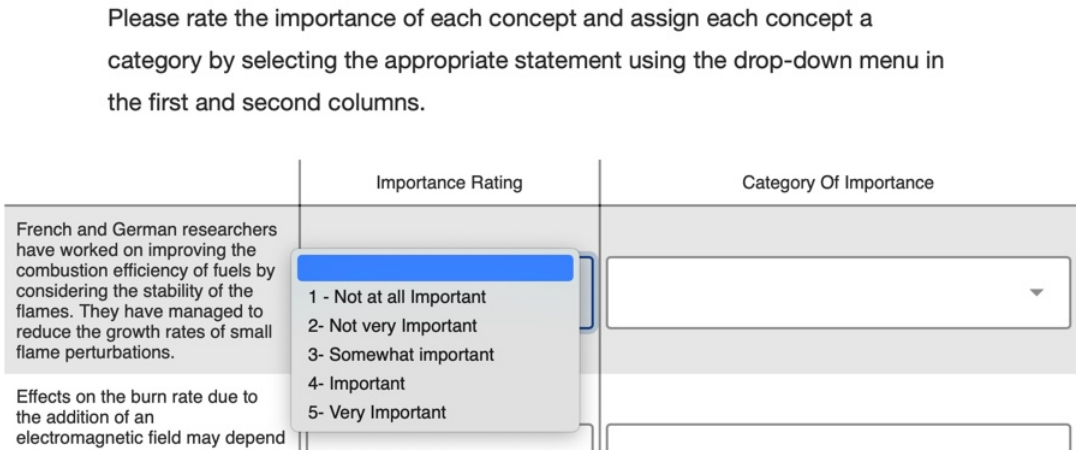


Figure 6.2: Example of a question from the statement-analysis task. Participants are asked to assign statements an importance rating from a drop-down menu.

The second component of the statement-analysis task aims to identify the extent to which the participants agree on why statements are important. In Chapter 5, an interview with the expert was conducted to understand why they felt specific points in the text were important. The analysis generated a series of conceptual categories that represented the range of explanations given by the expert. These categories are referred to as 'independent-reasoning categories' and are summarised in Table 5.4. In this study the independent-reasoning categories are referred to as 'categories of importance'. Participants were asked to select the categories of importance they felt best represented why the statement was important from a drop-down menu. Figure 6.3 shows an example question from this exercise.

To measure the extent of the agreement among participants, an IRR metric will be used. A

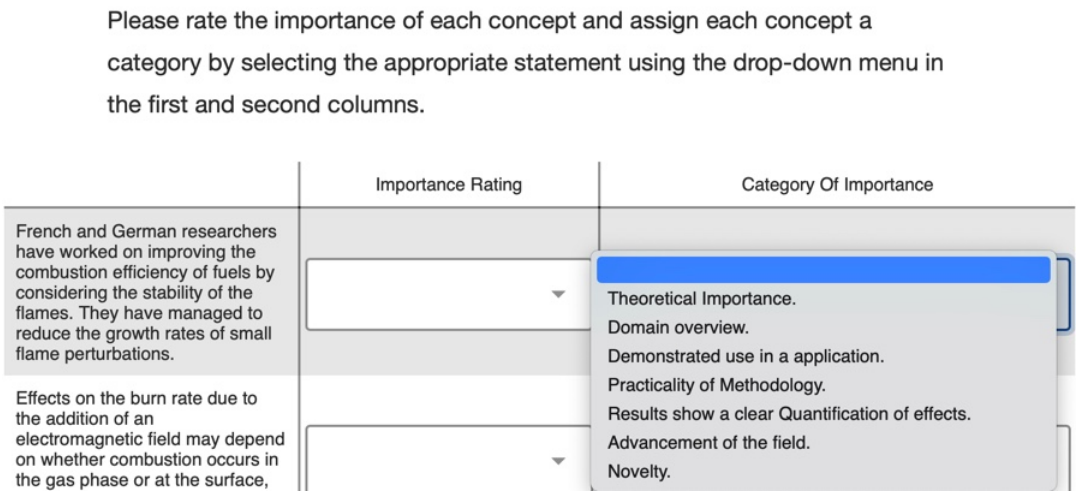


Figure 6.3: Example of a question from the statement-analysis task. Participants are asked to assign a category of importance to statements using a drop-down menu.

review of these metrics can be found in Section 6.1. For simplicity, it was decided to use one metric to analyse both parts of the statement-analysis task. Since this involves the analysis of both ordinal and categorical data, a flexible metric is needed. Therefore, the flexible $\hat{\kappa}_G$ (Eq. 6.5) was selected. This metric was selected over other surveyed flexible metrics as it does not suffer from the kappa paradoxes. Use of this metric was implemented using the irrCAC Python package [369].

6.3 Results

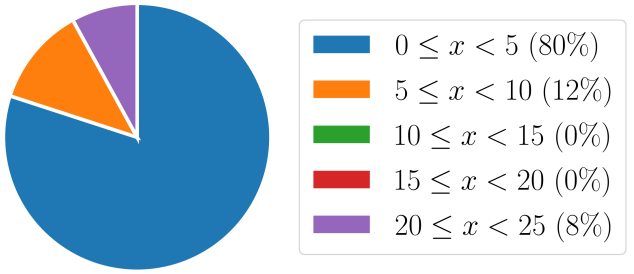
This section presents the results of this study. First, an overview of the background information of the readers will be presented in Section 6.3.1. Second, Section 6.3.2 describes the results of the summarisation task. Then, in Section 6.3.3, the results of the statement-analysis task will be presented. Finally, in Section 6.3.4, the participants' performance across the summarisation and statement-analysis tasks will be analysed.

6.3.1 Background information of the readers

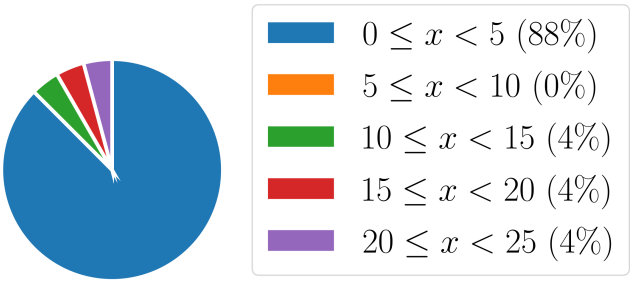
In this subsection an overview of the background information of the 25 readers will be presented.

The readers were asked to specify the number of years of experience they had in academic research, commercial research, government research and government policy. None of the readers

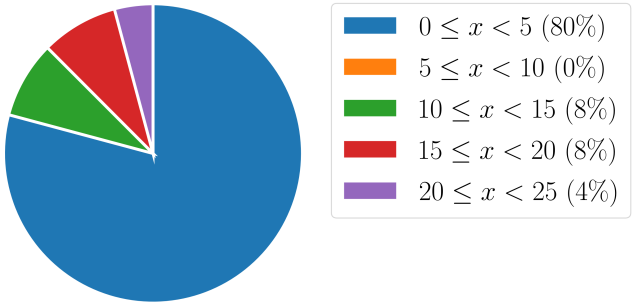
had any experience in government policy. The number of years of experience of the readers in each of the aforementioned domains were classified into ordinal categories, e.g. 0-5, 5-10 etc. The proportion of readers by the number of years of experience in academic research, commercial research and government research are displayed in Figures 6.4 (a), (b) and (c), respectively[2]. Total number of years of experience of each reader was converted into ordinal categories for this analysis. The proportion of readers by the total number of years of experience is shown in Figure 6.4 (d).



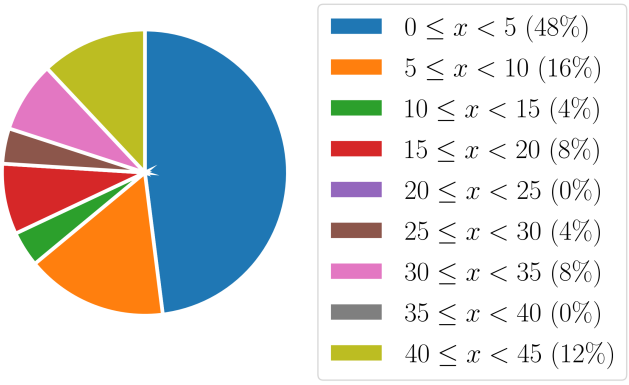
(a) Proportion of readers by the number of years of experience in academic research.



(b) Proportion of readers by the number of years of experience in commercial research.



(c) Proportion of readers by the number of years of experience in government research.



(d) Total number of years experience of the readers.

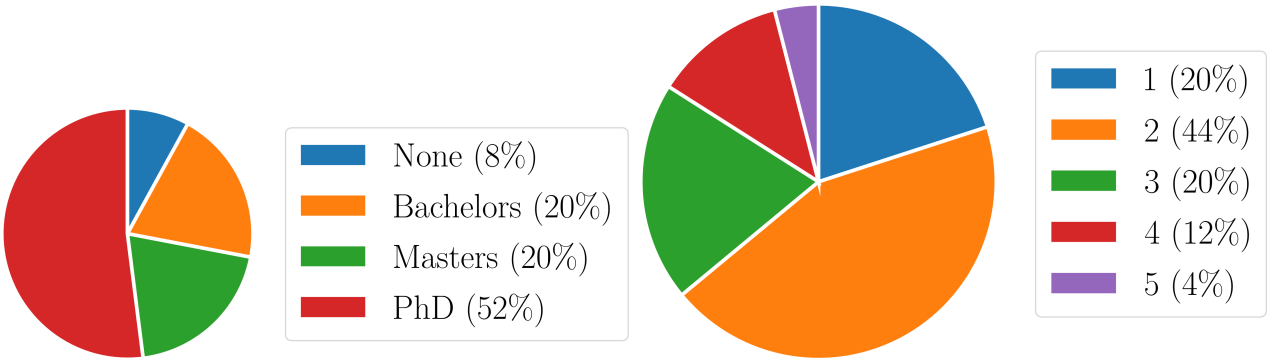
Figure 6.4: Overview of the number of years of experience of the readers in academic research (a), commercial research (b), government research (c) and total (d). The number of years of experience are divided into ordinal categories as shown in the legends.

Figure 6.4 indicates the recruited readers have a broad spectrum of backgrounds. This is in contrast to the study conducted in Chapter 4 which recruited a small group of participants mainly consisting of experienced academics.

In addition, the readers were also asked to list any higher-education qualifications. The inclusion criteria for the readers (excluding the novices) was to have a STEM background. This question

[2]The total number of years of experience of each reader was calculated by summing the values for the number of years of experience in academic research, commercial research, government research and government policy.

was used to identify if the readers met the inclusion criteria. The proportion of readers by their STEM higher-education qualifications is shown in Figure 6.5 (a). For this analysis, these STEM higher-education qualifications were classified as 'None', 'Bachelor's', 'Masters' and 'PhD'. The readers were also asked to rate from 1-5 their expertise in the subject matter of the text. A score of 5 would mean the participant considers themselves an expert on the effect of electric and magnetic fields on propellant combustion. The proportion of readers by their self-assigned expertise rating is shown in Figure 6.5 (b).



(a) Proportion of readers by their STEM higher-education qualifications.

(b) Proportion of readers by their self-assigned expertise rating.

Figure 6.5: Overview of the STEM higher-education qualifications and self-assigned expertise ratings of the readers.

It is also noted the majority of the readers are highly educated, with 52% of the readers having STEM PhDs. From Figure 6.5 (b), 20% of the readers rated themselves as having the lowest possible expertise rating (1); this included the novices. Thus, interestingly, some of the readers with STEM qualifications rated themselves as having the same expertise levels as the novice participants.

Kendall's Tau was used to examine the relationship between the self-assigned expertise level and the higher-education qualifications of the readers. No statistically significant correlation was found. This process was repeated for number of years of experience of the readers in government research, commercial research and academic research and the total number of years of experience of the reader. No statistically significant correlations were observed apart from the total number of years of experience of the readers. A value of 0.40 was obtained at a 0.01 significance level. This indicates that as the experience level of the reader increases, so does their self-assigned expertise level with a moderate strength [370]. Whilst this indicates that, as expected, the majority of more experienced readers rated themselves as having a higher-

expertise level than those with less experience, this was not the case for every reader. This may be because the questions asked were not specific enough. Future studies should ask specific questions about the reader's knowledge of, and experience in, energetic materials as a discipline. For instance, participants could be asked to specify the number of papers they have read in the energetic-materials domain.

In summary, the majority of readers are in the early stages of their career with the majority having PhDs in STEM subjects. The backgrounds of the readers are considerably more varied than the readers recruited for the study in Chapter 4.

6.3.2 The summarisation task

In this subsection, a comparison of the ability of NLP tools, novices and experts to identify important points in the text will be conducted.

Intra-group agreement: readers vs expert vs NLP

Section 6.2 defined the $RAIS'$ metric for measuring how important the readers found each sentence in the text as a group. Each sentence in the text was assigned a $RAIS'$ score. The text has a total of 200 sentences; 89 sentences were found to be important by two or more readers, and 38 were found to be important by one reader only, which indicated a variety of interpretations of importance from the readers. Some of the sentences not identified as important by the readers were identified as important by the NLP and/or the expert.

It was hypothesised the group of readers would identify sentences in the conclusion and introduction of the text as important; as in academic papers/technical reports, salient information is typically included in these sections. To examine this hypothesis, the index of each sentence was plotted against the assigned $RAIS'$ score in Figure 6.6.

Figure 6.6 shows no clear relationship between the sentence index and $RAIS'$ for the bulk of the sentences. However, there is a noticeable series of larger peaks for sentence in the range of 190-200. This indicates that sentences in the conclusion were viewed as more important on average than those in the main text and introduction. As illustrated in Figure 6.6, the sentences that obtained three of the highest values of RAIS' (including the highest value of 17) were all

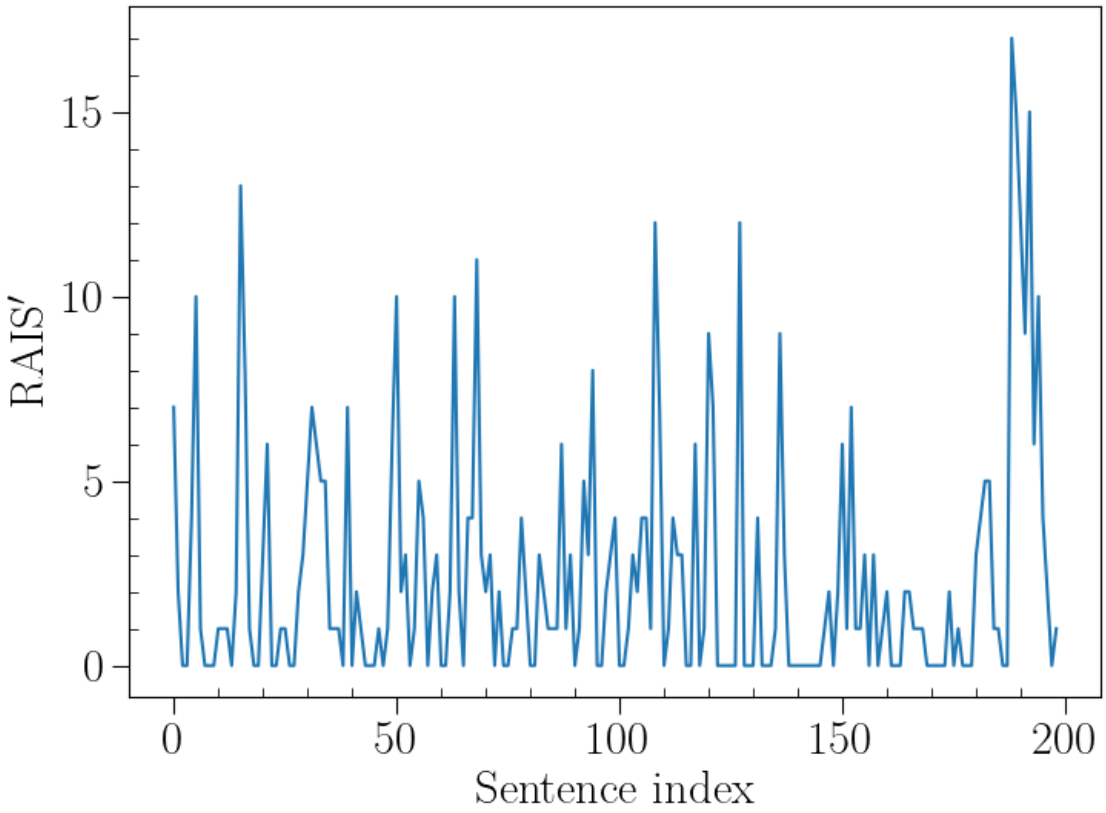


Figure 6.6: Index of each sentence plotted against the assigned $RAIS'$ score.

in the document's conclusion. However, some sentences in the conclusion were not identified as important by any of the participants, even if they were making a conclusive remark, e.g. 'A watching brief should be maintained for new publications on the magnetic and electrical control of combustion.'.

One of the two novices only identified sentences in the introduction and conclusion as important. It is well-documented that when presented with a problem, novices tend to focus on surface features, e.g. facts and equations, rather than concepts [371]. This novice's response to the task may be an example of such behaviour. The relationship between other basic surface features, such as sentence length and the number of keywords in a sentence, and the $RAIS'$ score was examined. No relationships were identified.

Intra-group agreement between the readers and the expert was conducted. The approach followed in Chapter 4 was adapted to the data in this study. The sentences in the text were divided into two datasets: those the expert found important and those the expert did not find

important. Figure 6.7 shows the distribution of $RAIS'$ for these two datasets of sentences.

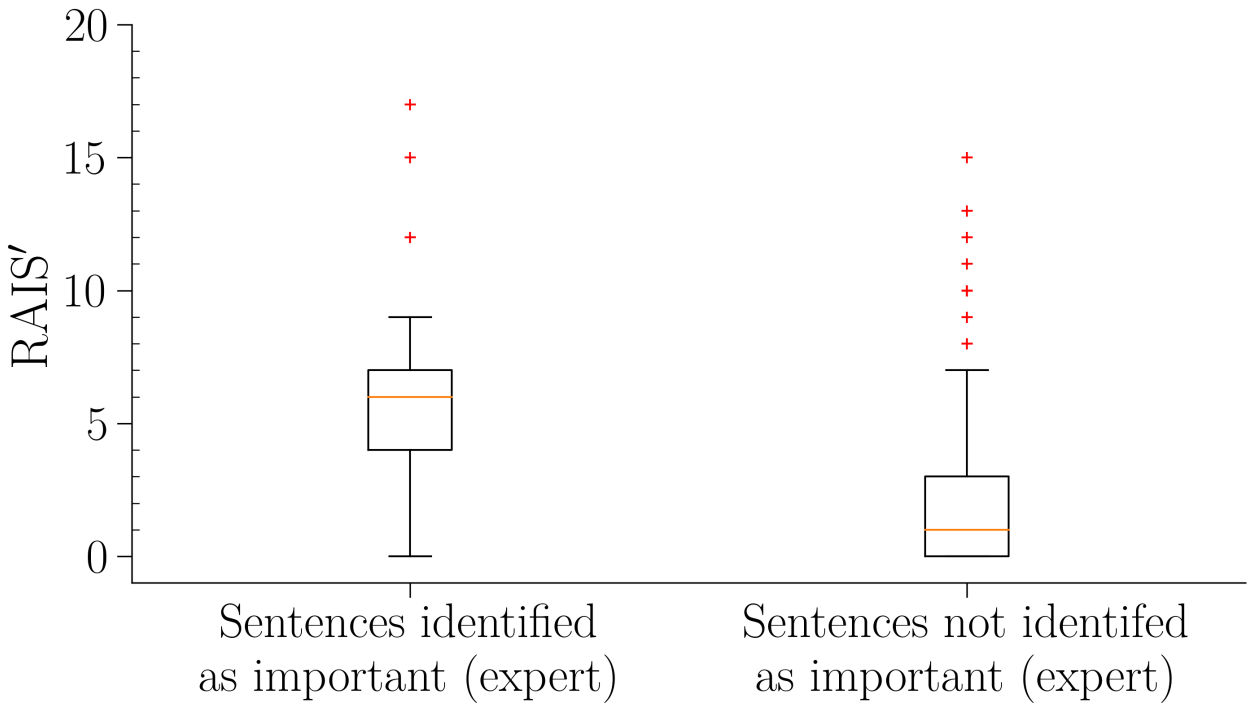


Figure 6.7: Distribution of $RAIS'$ for sentences identified (respectively, not identified) as important by the expert are on the left (respectively, right).

Figure 6.7 shows the distribution of $RAIS'$ for sentences the expert did not identify as important is heavily skewed towards lower values of $RAIS'$ and thus is non-normal[3]. On the other hand, the distribution of $RAIS'$ for sentences the expert found important is significantly more symmetric (except for a few outliers). In order to determine the correct statistical test, the normality of the distributions was examined using the Shapiro-Wilkes test. P-values of 0.08 and 3×10^{-19} were obtained for the $RAIS'$ of sentences identified and not identified as important by the expert, respectively. At a 0.05 significance level[4], it can be determined that the distribution of $RAIS'$ for sentences identified as important by the expert is normally distributed, whereas the distribution for sentences not identified as important by the expert is not normally distributed.

Given one of the datasets is non-normal a Wilcoxon rank sum test was conducted[5]. This test compares the medians between the two datasets. The median $RAIS'$ for sentences not

[3] As mentioned in Section 4.3.3, determining the normality of the data is essential to determining the correct statistical test.

[4] Here, and throughout the rest of this chapter, we assume statistical significance at a value of 0.05.

[5] See Chapter 4 Appendix E for an overview of the decision-making process used to determine the appropriate statistical test.

identified as important by the expert was 1, significantly lower than the median $RAIS'$ for sentences identified as important by the expert which was 6. The p-value of the Wilcoxon rank sum test was 1.88×10^{-6}. Therefore it can be concluded the median $RAIS'$ for sentences identified as important by the expert is significantly different than the median $RAIS'$ for sentences not identified as important by the expert. The median of the difference between a sample from $RAIS'$ for sentences identified as important by the expert and a sample from $RAIS'$ for sentences identified as important by the expert was determined to be 4.76. The size effect was measured using Vargha and Delaney's A, a value of 0.82 was obtained, which is interpreted as there being a large difference between the $RAIS'$ of sentences identified and not identified as important by the expert. It is therefore concluded the readers were more likely to identify a sentence as important if the expert also identifies the statement as important. This is in agreement with the results found in Section 4.3.3.

Intra-group agreement between the readers and the NLP was conducted. The sentences in the text were divided into two datasets: those the NLP found important and those the NLP did not find important. Figure 6.8 shows the distribution of $RAIS'$ for these datasets.

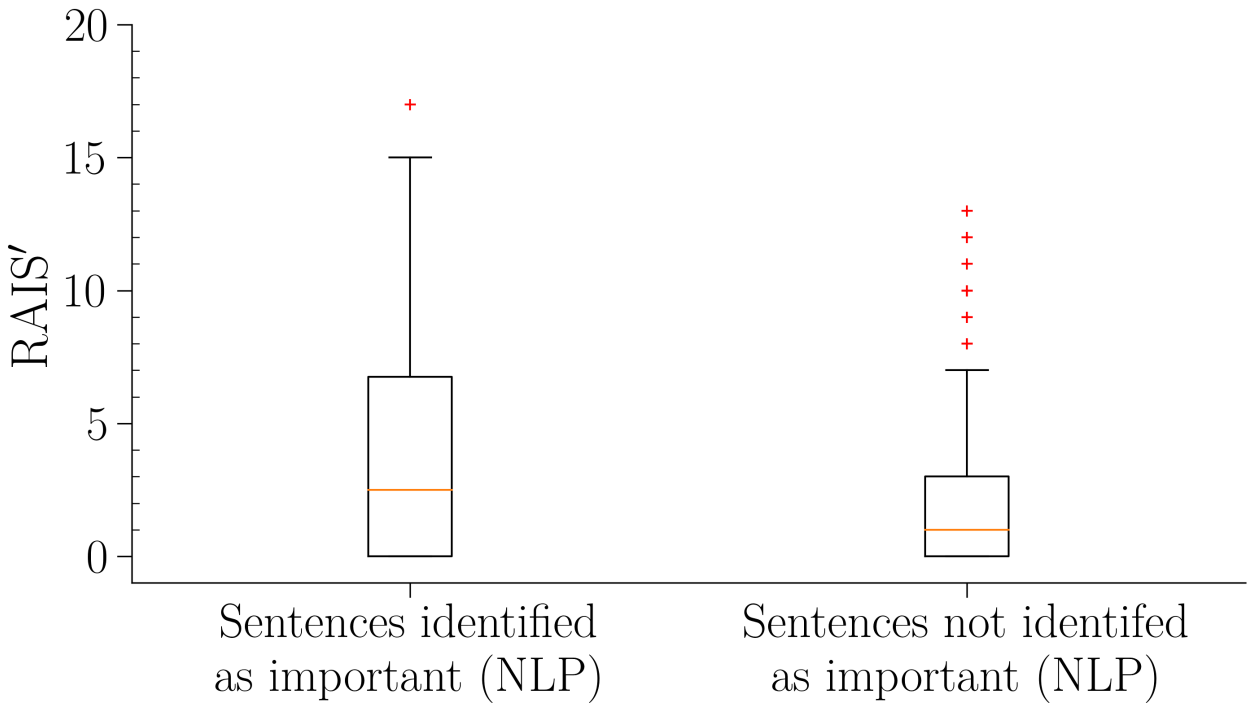


Figure 6.8: Distribution of $RAIS'$ for sentences identified (respectively, not identified) as important by the NLP are on the left (respectively, right).

The median RAIS' for sentences not identified as important by the NLP (1) is slightly lower than the median $RAIS'$ for sentences identified as important by the NLP (2.5). Both the

distributions of $RAIS'$ are heavily skewed towards lower values indicating they are non-normal[6]. Therefore a Wilcoxon rank sum test was conducted. A p-value of 0.06 was obtained. Therefore it is concluded the median $RAIS'$ for sentences identified as important by the NLP is not significantly different from the median $RAIS'$ for sentences not identified as important by the NLP.

Inter-group agreement: readers vs expert vs NLP

Inter-group agreement is examined by first calculating the overlap between each of the readers', NLP's and experts' summaries using the F_A metric (Eq. (6.8). Figure 6.9 displays the results in the form of a lower half of a similarity matrix.

Figure 6.9 shows a significant number of pairings did not identify any of the same sentences as being important, e.g. between readers 1 & 7, 21 & 3 and 25 & 12. This reflects the sheer variety of sentences identified as important by the readers. This result is in agreement with Chapter 4 which similarly observed a large variation in the concepts the readers found important in the text. As mentioned in Chapter 4, the variety of responses is unsurprising from a psychological perspective as experts bring to any task their own set of expectations and ideas. In this study, a broad spectrum of readers were recruited across a range of STEM backgrounds. Since different scientific disciplines are taught to evaluate technical information in different ways, it is unsurprising that summary overlap varied considerably. Interestingly, none of the pairings that obtained $F_A = 0$ included either the expert or the NLP. As mentioned in Chapter 4, the NLP returned many conclusion-style sentences. These conclusion-style sentences often portray important conceptual points. Hence it is unsurprising no values of $F_A = 0$ were obtained for any reader-NLP pairings.

[6]This was verified using Shapiro-Wilk tests, for which p-values of 0.001 and 1.28×10^{-15} were obtained for the $RAIS'$ of sentences the NLP did and did not find important respectively. Hence it can be concluded both distributions are significantly different from the normal distribution.

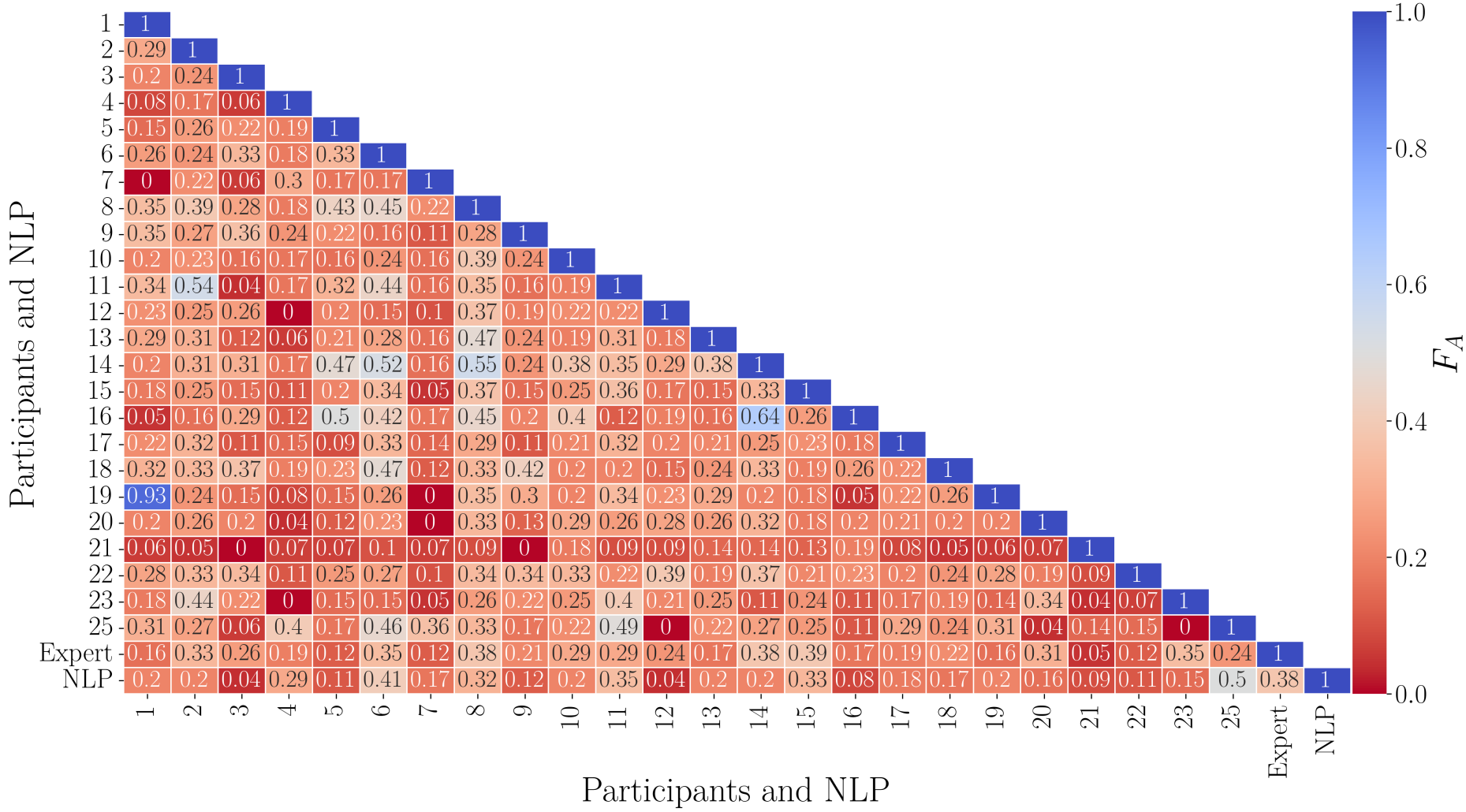


Figure 6.9: Similarity (F_A) between the readers, experts and NLP summaries. The colour bar displays the relationship between F_A and the cell colour.

Figure 6.9 shows the highest obtained F_A was between Readers 1 and 19, with a value of $F_A = 0.93$. These participants assigned themselves the same expertise rating and had similar educational backgrounds but had differing years of experience in academia, commercial research, government research and overall. In total, there were 7 pairings out of a possible 351 that achieved $F_A \geq 0.5$, i.e. agreed on average on 50% of the selected sentences. A total of 6 of these pairings were between the readers. Overall, there is low agreement between the participants and NLP on what is important in the text.

Readers 4 and 15 are novice readers. It was expected they would produce similar summaries as they both had little knowledge of STEM and would approach the task similarly. However, the novice readers obtained $F_A = 0.11$, which is a low similarity. The novices had differing approaches to the task. As mentioned, Reader 4 conducted its analysis based on surface features highlighting sentences exclusively found in the introduction and conclusion. Reader 15, however, identified sentences throughout the text and interestingly had the highest similarity with the expert's summary, obtaining $F_A = 0.39$. This was also the highest similarity for the expert. This could be a coincidence. Alternatively, unlike the technically-trained readers who would have approached the task with preconceived ideas or methods on how to evaluate the importance of technical information in the text, the novice may have been able to distinguish what the expert, who was also the author of the text, found important in the text by following semantic cues in the text itself. As highlighted in Ref. [327], novices are more sensitive to the form in which information is presented, and minor changes to the wording can influence their perception of how important a specific point in the text is.

Participant 25 had the highest similarity with the NLP obtaining a value of 0.50. Participant 25 had highlighted a high proportion of summary style sentences this likely accounts for the high similarity. The NLP and expert obtained $F_A = 0.38$. This value is unsurprising since the method used to select the techniques to generate NLP's summary was optimised to obtain high conceptual and uni-gram overlap with a summary generated by the expert, see Section 4.3.1.

For each of the participants and the NLP, the average F_A with the other readers $\langle F_A \rangle'_{R_i}$, (Eq. (6.11), was calculated. This process was repeated with the novices, Readers 4 and 15, removed. Figure 6.10 shows $\langle F_A \rangle'_{R_i}$ for each of the participants and NLP with (red line) and without (blue line).

Figure 6.10 shows novice Reader 4, obtained low average agreement with the other readers.

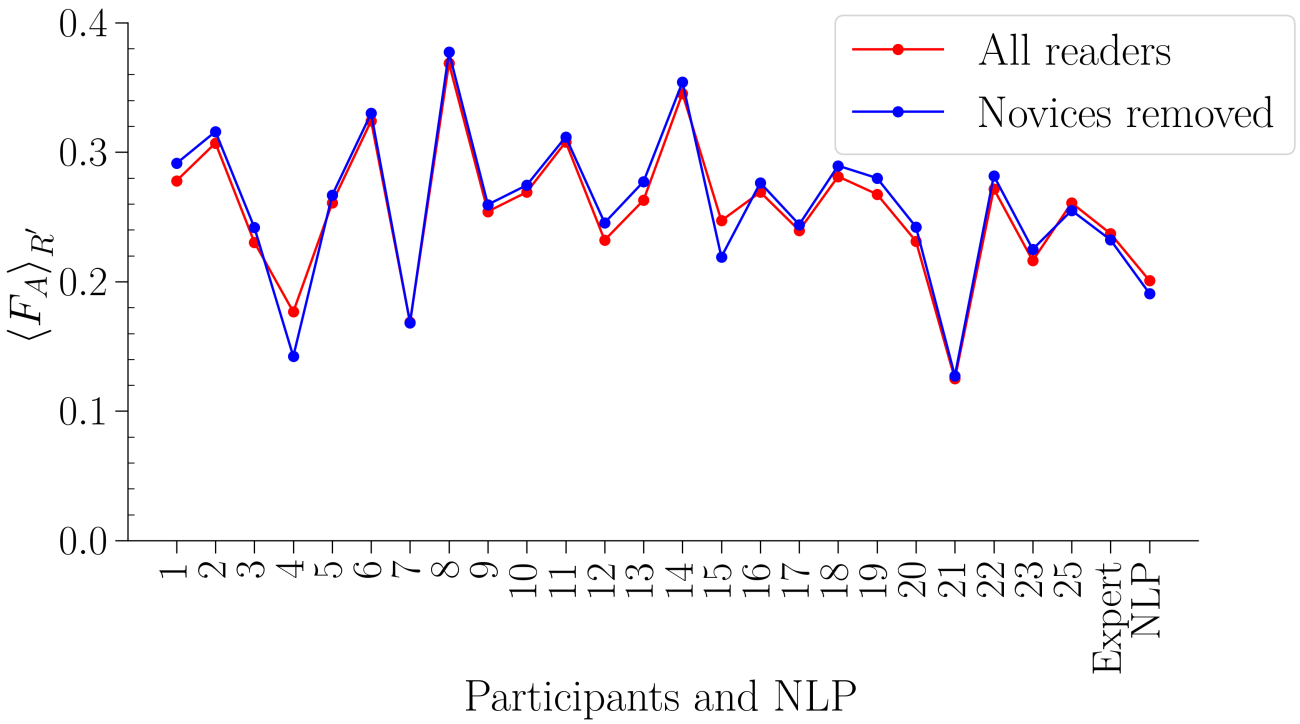


Figure 6.10: Average overlap of each of the participants/NLP summaries with all other reader summaries, $\langle F_A \rangle'_{R_i}$. Values are calculated with all readers (red line) and with the novices removed from the dataset (blue line).

This was likely because the reader's approach to the task differed substantially from the others (Reader 4 exclusively highlighted sentences in the introduction and conclusion of the text).

Figure 6.10 shows only three readers, one of which was the novice participant, obtained lower $\langle F_A \rangle'_{R_i}$ values than the NLP. It can therefore be interpreted that the average agreement with the readers is lower for the NLP than for the majority of other participants. The NLP failed to recognise the importance of the sentences in the text as the readers did. This is in agreement with the results of intra-group comparison made with the NLP summary earlier in this subsection. Overall, the NLP failed to identify sentences that the readers as a group found important and obtained a lower average agreement with the readers than the majority of the other readers. This may be because the NLP identified sentences using statistical techniques rather than conceptually. Therefore, it may have returned sentences with a high frequency of relevant words about a specific topic but does not necessarily portray a relevant conceptual point, or the conceptual point is portrayed better elsewhere in the text. In addition, unlike the participants, the NLP evaluated the importance of the sentences on a sentence level. As a result, it was common to see multiple sentences in a row highlighted in the summaries produced by the participants. These multiple sentences would express a single concept. The NLP,

however, was unable to analyse the text in this way. Furthermore, unlike the participants, the NLP could not identify when redundant information was returned effectively. As a result, the NLP's summary contained several sentences portraying the same conceptual point.

It was hypothesised the readers' experiences impacted how they approached this task. Therefore for each possible pairing of readers, the relationships between the difference in higher-education qualification obtained (using the ordinal scale mentioned earlier in this section) and F_A was examined using Kendall's Tau. A value of -0.10 was obtained. This value was statistically significant, indicating a weak-negative association between the F_A of two generated summaries and the difference in education levels of the readers, i.e. there is a relationship between the educational backgrounds of the summaries and the extent to which the generated summaries overlap. This process was repeated for self-assigned expertise rating, and no statistically significant relationship was observed[7].

6.3.3 The statement-analysis task

The statement-analysis task consists of two components where participants are asked to read and digest series of statements then assign each statement (1) a 1-5 importance-rating and (2) a category of importance. This subsection will present the results of each component of the statement-analysis task.

Importance ratings

To examine the agreement on the ratings of importance between every possible pair of participants, $\hat{\kappa}_G$ was used. Figure 6.11 shows an agreement matrix containing the $\hat{\kappa}_G$ values between every possible pair of participants. In addition, the Landis-Koch scale is shown in the colour bar for interpretation of $\hat{\kappa}_G$.

Figure 6.11 shows the majority of participants exhibit at least moderate levels of agreement on the rating of statements' importance. Approximately 10% of participants had almost perfect

[7]The process was repeated for the difference in the number of years of experience in academia, commercial research, government policy, government research and the total number of years of experience using Pearson's correlation. Therefore, normality tests were performed using the Shapiro-Wilk test to verify the data was normally distributed. No statistically significant correlations were found.

agreement on ratings of importance. Furthermore, 44% of pairings had a substantial agreement among their importance ratings. There were very few participant pairings with poor agreement. Readers 20 and 21 had the highest agreement, obtaining a value of $\hat{\kappa}_G = 0.93$, which can be interpreted as near-perfect agreement. Though they had the same level of higher-education qualifications these readers had completely different backgrounds.

Overall the participants agreed to an extent on the importance of the statements given to them. This contrasts to the results shown in a similar study conducted in Ref. [6], which saw low levels of agreement on participants' ratings of importance. This may be due to the choice of text, in Ref. [6] the subject of the text is terrorism hostage-taking. Unlike the topic of the text in this study, terrorism hostage-taking is an emotive topic even when approached from a technical perspective. Given this and that the participants used in the said study were not familiar with the domain from a technical perspective, the lack of agreement is expected.

Figure 6.11 shows the readers obtained fair to substantial agreement with the expert, except for Reader 9, who obtained $\hat{\kappa}_G = 0.01$. Novice Reader 4 obtained the third lowest agreement with the expert with a value of $\hat{\kappa}_G = 0.39$, which can be interpreted as fair agreement. It would be expected that novices would have low agreement with experts as they do not have an in-depth conceptual understanding of the domain. However, novice Reader 15 obtained the highest agreement with the expert, obtaining $\hat{\kappa}_G = 0.72$. This same reader obtained the highest agreement with the expert on the summarisation task. Unlike the technically-trained readers who would have approached this task with their predefined ideas of importance, the novices' perception of importance in this domain may have been entirely shaped by reading the text, resulting in Reader 15 rating the importance of the statements in the text similarly to the expert.

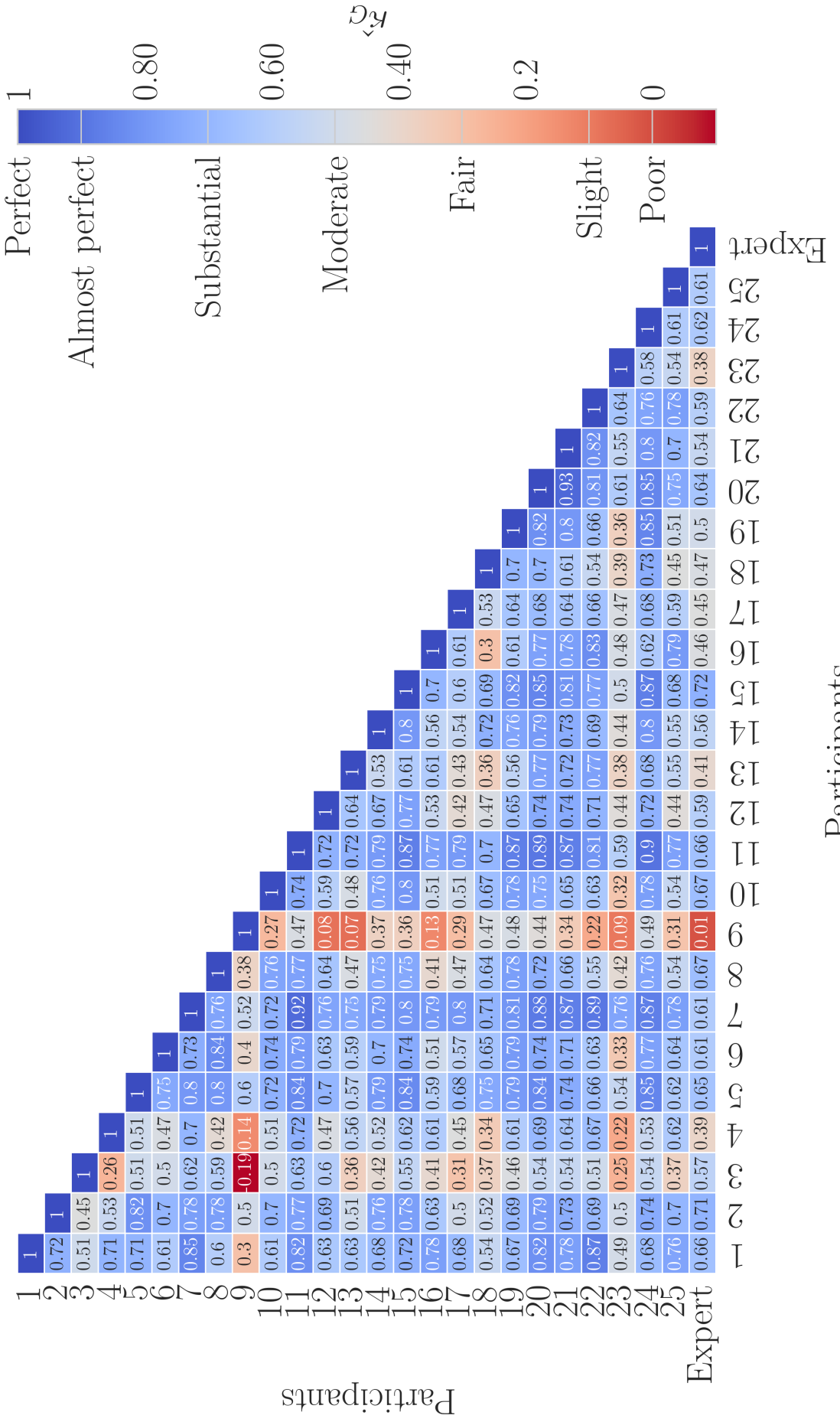


Figure 6.11: Agreement matrix calculated using $\hat{\kappa}_G$. For interpretation of $\hat{\kappa}_G$, the Landis and Koch scale is shown on the colour bar.

In addition, the agreement between the novice readers was 0.62, which can be interpreted as substantial agreement. In Section 6.3.2, the overlap of the summaries produced by the novice readers was 0.11. Whilst the novices did not agree on identifying important sentences in the text, they did agree a substantial amount when it came to rating the importance of statements about the text.

Readers 1 and 19 had a very high F_A on the summary task (Section 6.3.2); likewise, on this task, these participants had a substantial agreement on rating the importance of the categories. Similar results are observed for Readers 14 and 16. The relationship between the performance on each task will be explored further in Section 6.3.4.

For each participant, the average agreement with the other readers ($\langle \hat{\kappa}_G \rangle_{R'}$) was calculated using Eq. (6.11) where F_A is substituted with $\hat{\kappa}_G$. This process was repeated with the novice Readers (4 and 15) removed from the dataset. Figure 6.12 displays the results for $\langle \hat{\kappa}_G \rangle_{R'}$ for each participant with all readers.

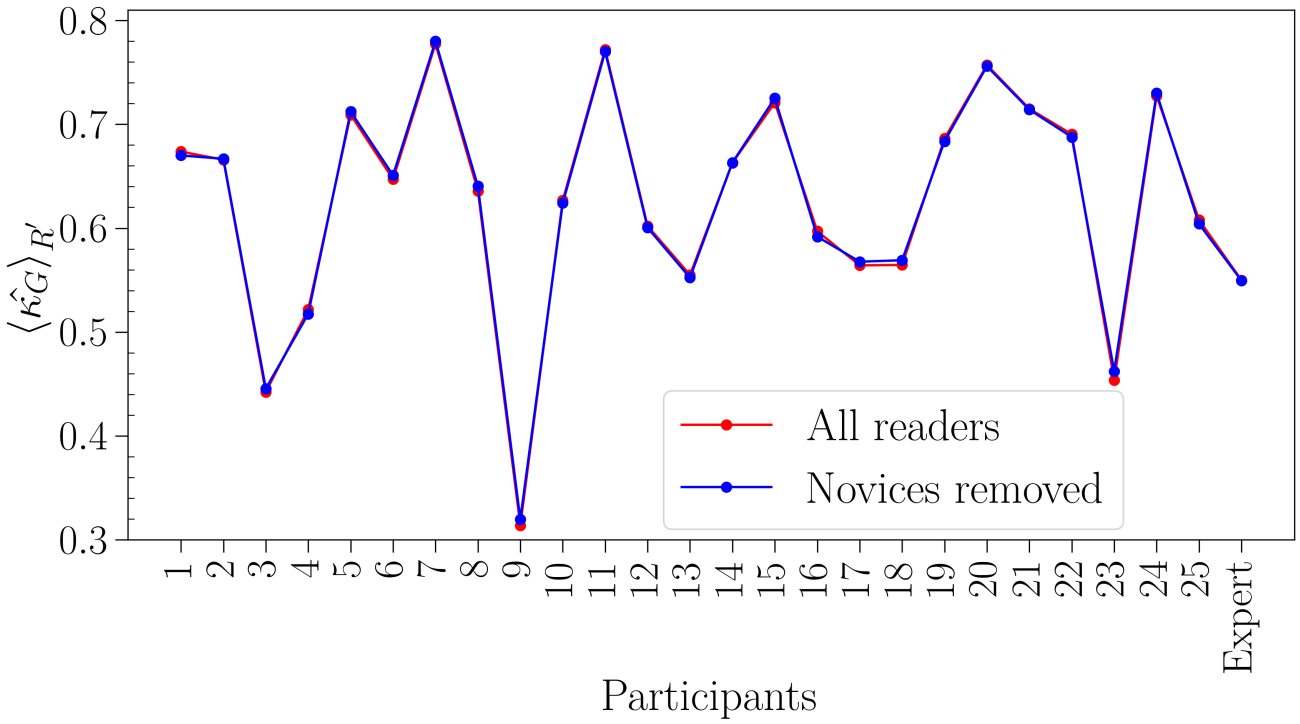


Figure 6.12: $\langle \hat{\kappa}_G \rangle_{R'}$ for each participant, where all readers, and without the novices in the dataset, are shown in red and blue respectively.

Figure 6.12 shows the effect of removing the novices from the dataset has minimal impact on $\langle \hat{\kappa}_G \rangle_{R'}$ for each participant. Therefore, for each participant, the average agreement with the other readers is largely unaffected by the agreement with the novices.

In addition, Figure 6.12 shows that Reader 9 had a low $\langle \hat{\kappa}_G \rangle_{R'}$. Readers 7, 11 and 20 are noticeable peaks in Figure 6.12 with high values of $\langle \hat{\kappa}_G \rangle_{R'}$ indicating high levels of agreement with the other participants. Interestingly, Reader 7 had low agreement with the other participants in the previous task, see Figure 6.10.

As with the summarisation task (Section 6.3.2), it was hypothesised that the readers' experiences impacted how they approached this task. The procedure outlined in the previous section was followed. Results found the difference in the number of years of experience in academic research and $\hat{\kappa}_G$ are significantly correlated with a correlation coefficient of 0.17. This can be interpreted as a weak-positive correlation [370]. Therefore, the greater the difference in the number of years of experience of the readers in academic research the higher the agreement.

The analysis of the importance-rating task thus far has been on an individual basis (inter-group agreement). Here, the agreement among the readers as a group will be considered. For each statement, the mean importance rating assigned by the readers is calculated[8]. This will be referred to as $MRIS'$.

The statements were segmented into two datasets: statements the expert rated as important (assigned an importance rating of 4 or 5) and statements the expert rated as not important (assigned an importance rating of 1 or 2). The statements the expert rated as 3, neither important nor not important, were disregarded.

Figure 6.13 shows the distribution of $MRIS'$ for statements rated as important by the expert and for the statements rated as not important by the expert are symmetric, indicating they are normally distributed[9]. As mentioned in Section 4.3.3, normality is not the only condition for a t-test. The variances of the distribution must be equal. Variances of 0.17 and 0.40 were obtained. Therefore, a Welch's t-test [372] was performed. A Welch's t-test determines if the means of two populations populations with unequal variances are statistically different. A p-value of 0.02 was obtained, indicating the two means of 3.63 and 3.01 for statements rated important and not rated important by the expert, respectively, are statistically different from

[8]The mean was taken as opposed to the sum as some of the readers did not assign every question an importance rating. The presence of missing data would result in lower scores (using the sum) for those with missing data compared to those without. This would result in the perceived notion that these sentences are less important than those without missing data. Therefore one needs to account for this by using the mean by averaging only over participants who did assign a rating.

[9]Shapiro-Wilk tests were performed to determine the normality of the data. P-values of 0.27 and 0.87 were obtained, indicating the two distributions are normally distributed.

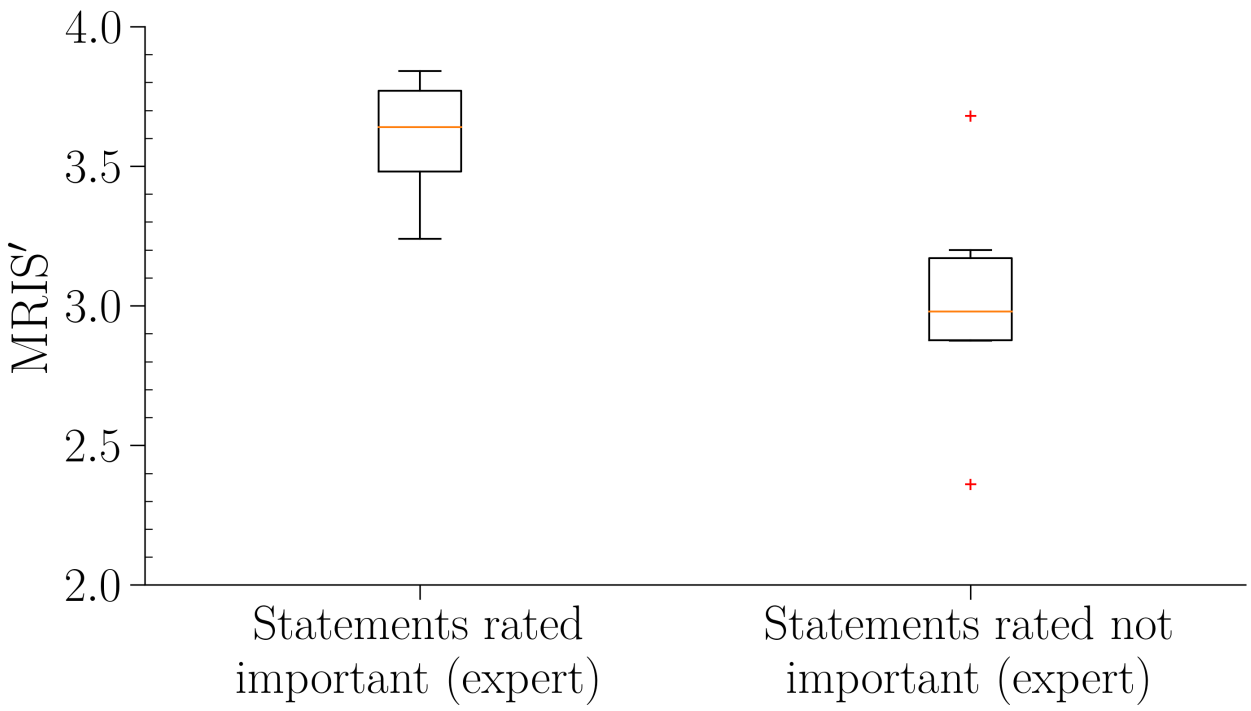


Figure 6.13: Distribution of $MRIS'$ for statements rated as important by the author and rated as not important by the expert.

each other. Therefore, on average, statements the author rated as important had higher $MRIS'$ values, i.e. were found more important by the participants than those rated not important by the author.

As discussed in Section 6.2, the statements used in this task were generated from the key-points/summaries outputted from the preliminary study in Chapter 4. To examine the consistency of the author's ratings of importance, the relationship between the rated importance of the statements in the statement-analysis task and whether the statements were identified as important by the expert in the preliminary task was conducted. First, the statements were divided into two datasets: those identified as important and not identified as important by the expert in the preliminary study.

Figure 6.14 shows the distribution of importance ratings for the statements identified as important by the expert is not symmetric and is not normally distributed. The distribution of importance ratings for the statements not identified as important by the expert in the preliminary task, however, is symmetric and appears normally distributed[10].

[10]These hypotheses were verified with Shapiro-Wilk tests. P-values of 0.006 and 0.16 were obtained for those statements not identified as important and identified as important, respectively. It is therfore determined the distribution of importance ratings for the statements identified as important by the expert is significantly

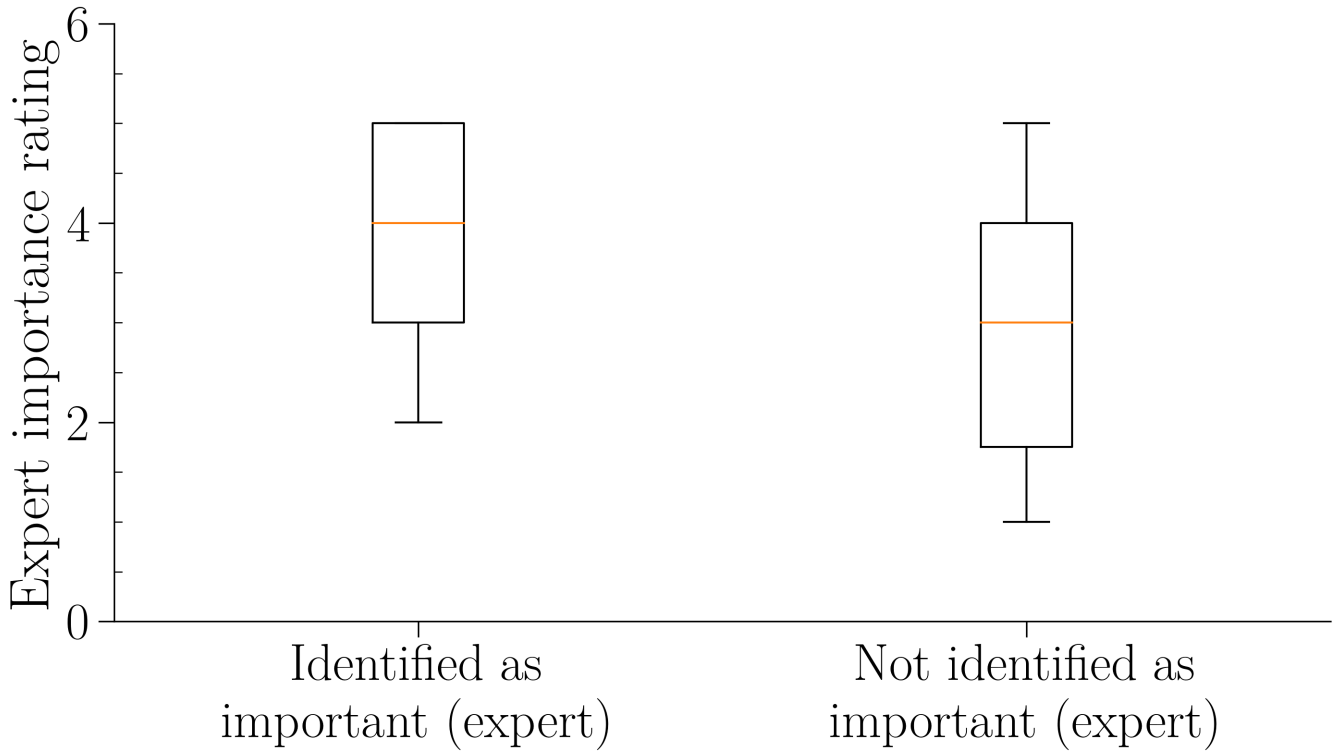


Figure 6.14: Distribution of importance ratings by the expert for statements identified as important and not identified as important by the expert in the preliminary study.

To examine the differences between the two distributions, a Wilcoxen rank test was performed. The p-value of this test was 0.03. Therefore it can be concluded that the median importance rating for statements identified as important by the expert (3.64) is significantly different from the median importance rating for statements not identified as important by the expert (2.98). The median of the difference between a sample from importance ratings of statements identified as important by the expert and a sample from importance ratings of statements not identified as important by the expert was 2.16. The size effect was measured using Vargha and Delaney's A. A value of 0.74 was obtained; this is interpreted as a medium difference between the importance of statements identified as important by the expert and not identified as important by the expert. It is therefore concluded the author was consistent in their evaluation of the importance of the technical information present in the text.

different from the normal distribution and is therefore not normal. In contrast, the distribution of importance ratings for the statements identified as important by the expert is normally distributed.

Categories-of-importance task

Participant intra-group agreement on assignment of categories of importance is calculated using $\hat{\kappa}_G$. Figure 6.15 an agreement matrix containing the $\hat{\kappa}_G$ between participants.

Figure 6.15 shows lower levels of agreement between participants than on the importance-rating task. Of the 325 possible pairings (excluding pairings with the participant's self), none of the participant pairings had greater than moderate agreement. This is in direct contrast to the rating task, which had 54% of pairings at the aforementioned agreement levels. In addition, on this task, 8% of the participant pairings had a moderate agreement, 46% had a fair agreement, and 44% had a slight agreement. A small percentage of pairs of participants had a poor agreement. This implies the participants agree with each other less on the reasons why a particular statement is important than they do on how important a statement is. This may be because the participants come from a broad spectrum of STEM backgrounds (Section 6.3.1), so whilst they may agree on how important a statement is, their own experiences will have shaped their idea of why these points are important. Furthermore, there are various valid reasons why a statement is important; therefore, it is expected that agreement is lower for this task.

Readers 11 and 2 have the highest agreement in Figure 6.15 with a $\hat{\kappa}_G$ of 0.58, which is interpreted as moderate agreement using the Landis and Koch scale. These readers have the same educational background and similar numbers of years experience in academic research. Reader 2 also obtained a similarly high agreement with Reader 5 with $\hat{\kappa}_G = 0.57$. The lowest observed agreement was between novice Reader 4 and Reader 3, with a $\hat{\kappa}_G = -0.01$. Reader 4 also obtained the second-lowest agreement, a value of $\hat{\kappa}_G = -0.06$, with Reader 7. As with previous tasks, the agreement between the novice Readers 4 and 15 is low, 0.02. Future studies should recruit a larger group of novice readers to see if similar results are obtained.

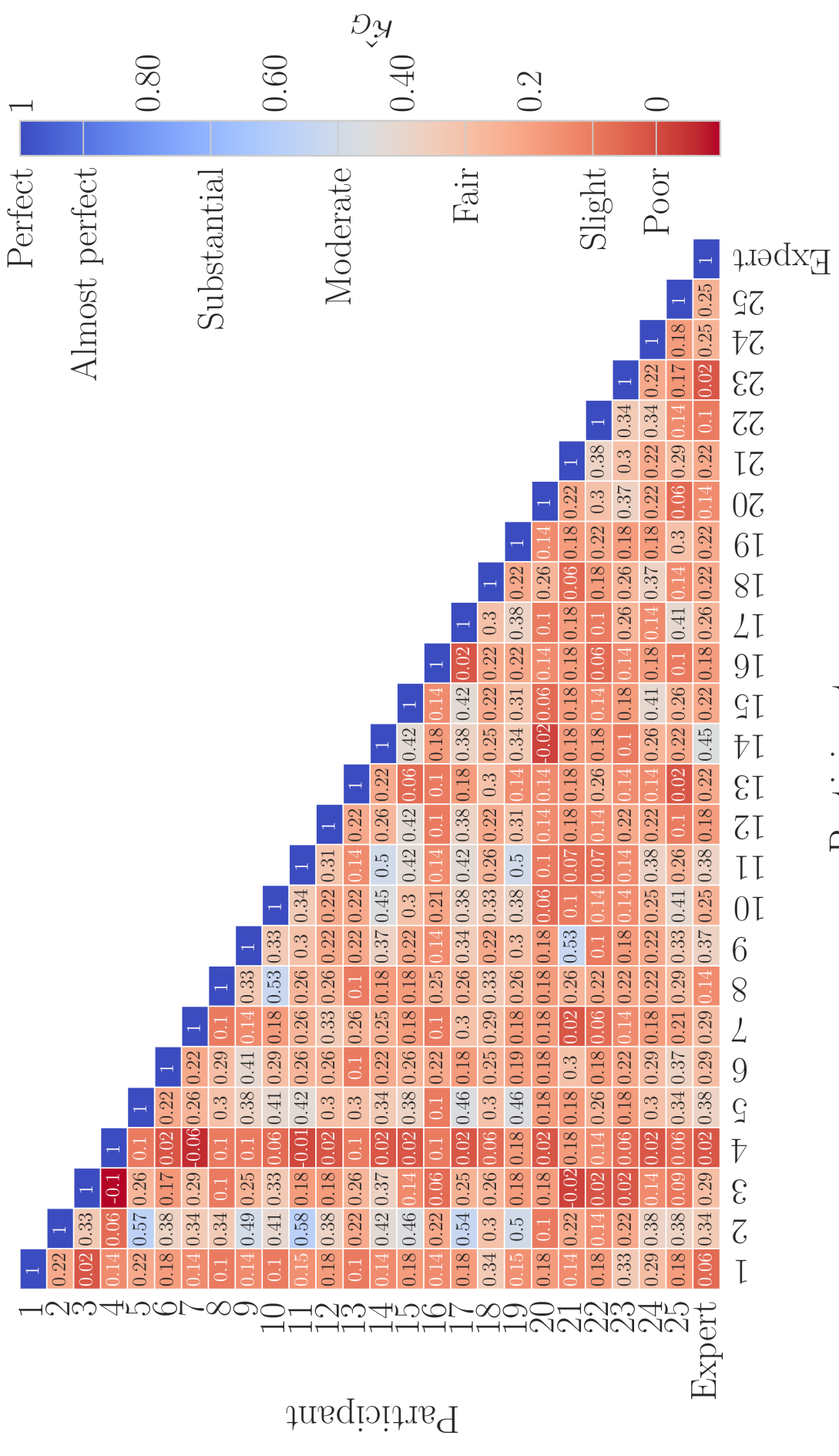


Figure 6.15: $\hat{\kappa}_G$ between every possible pair of participants for assigning categories of importance to statements about the text. Interpretation of $\hat{\kappa}_G$ using the Landis and Koch scale is shown on the colour bar.

As shown in Figure 6.15, the expert obtained a moderate agreement with Reader 14. Otherwise, the agreement between the readers and the expert was in the range of poor to fair. In previous tasks, the highest agreement with the expert has been from novice Reader 15. On this task, Reader 15 and the expert obtained an agreement of 0.22, a significantly lower agreement with the expert than on previous tasks. Whilst Reader 15 may have been able to use semantic clues in the text to identify which specific sentences in the text are important and what the expert considers to be important in the text, understanding and identifying why a specific point is important requires a complex content schema that a novice will not possess. This may explain why the expert exhibited higher agreement with other readers on this task.

For each of the participants, the average agreement with the other readers $\langle \hat{\kappa}_G \rangle_{R'}$ was calculated. This process was repeated with the novice Readers (4 and 15) removed. Figure 6.16 displays the results for $\langle \hat{\kappa}_G \rangle_{R'}$ for the participants with all readers and with the novices removed.

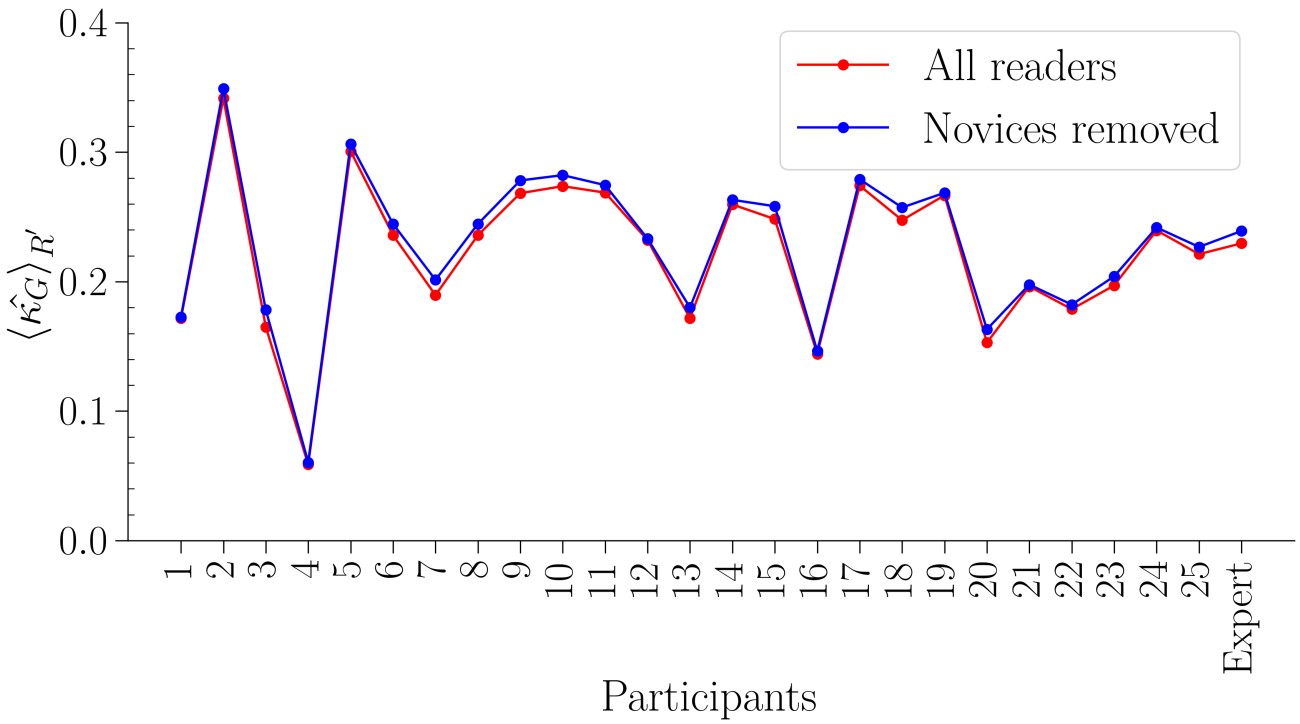


Figure 6.16: $\langle \hat{\kappa}_G \rangle_{R'}$ for each participant. This calculation was done with all readers and with the novices removed, shown in red and blue respectively.

Figure 6.16 shows that $\langle \hat{\kappa}_G \rangle_{R'}$ is, in general, higher for most of the participants when the novices are removed from the dataset indicating readers have a lower agreement with the novice readers than the other readers. Furthermore, $\langle \hat{\kappa}_G \rangle_{R'}$ is lowest for novice Reader 4 with a value of 0.06. This is unsurprising since identifying why a specific statement is important requires a deeper

understanding of the domain than a novice will possess. As shown in Figure 6.10, Reader 4 also had low levels of agreement with the other readers on the summarisation task. Readers 2 and 5 have the highest $\langle \hat{\kappa}_G \rangle_{R'}$, with values of 0.35 and 0.31 respectively. Reader 2 had similarly high average reader agreement on the summarisation task. This indicates that there may be a relationship between the agreement on different tasks, this will be investigated further in Section 6.3.4.

As with the previous tasks in this study, it is hypothesised the readers' background impacted how they approached this task. Results concluded there was a statistically significant weak-negative association (-0.13) between the difference in education levels and $\hat{\kappa}_G$. This can be interpreted as a greater similarity in readers' educational background corresponding to higher agreement.

The strength of the relationship between the difference in the number of years of experience in commercial research experience/academic research/government policy/government research/total number of years experience and $\hat{\kappa}_G$ was measured using Pearson's correlation[11]. Only 3 of these correlations were statistically significant. The difference in the number of years of experience in commercial research experience and $\hat{\kappa}_G$ is significantly correlated with a correlation coefficient of -0.15, which can be interpreted as a weak-negative correlation [370]. A value of -0.01 was obtained for the difference in the number of years of experience in government research experience and $\hat{\kappa}_G$, which also indicates no correlation. A value of -0.16 was obtained for the difference in the total number of years of experience and $\hat{\kappa}_G$. Therefore, the more similar the number of years experience in commercial research and the total number of years experience, the higher the agreement on assigning categories of importance.

The intra-group agreement on this task was determined for each statement taking the most-popular assigned category of importance and calculating the agreement with the expert's assignments of categories of importance. A value of $\hat{\kappa}_G = 0.33$ was obtained, which can be interpreted as moderate agreement. With the novices removed from the dataset $\hat{\kappa}_G = 0.37$ was obtained. Overall, this indicates the readers as a group agree with the expert to a moderate degree on assigning categories of importance, this agreement slightly increases with the novices removed from the dataset.

[11] Pearson's correlation requires normally distributed data. Normality tests were performed using the Shapiro-Wilk test.

6.3.4 Comparison between tasks

This subsection will combine and analyse the results across the statement-analysis (Section 6.3.3) and summarisation (Section 6.3.2) tasks.

First, the results of the two statement-analysis tasks are combined to examine if the readers associated some categories with a higher importance rating than others. To complete this analysis a Kruskal-Wallis test was performed. This is a non-parametric test used to determine if there are statistically significant differences between groups of an ordinal variable [373]. A value of 13.7 was obtained with a p-value of 0.03. It is therefore determined that there is a statistically significant difference in how important the readers viewed each category of importance. However, further analysis using a post hoc Dunn's test [374] determined no significant differences between the importance ratings of the categories of importance.

Second, the agreement between pairs of participants across tasks was examined using Pearson's correlations for the agreement on specific tasks between pairs of participants. Results found there was a weak-positive correlation of strength 0.19 on the agreement between participants on the summarisation task (F_A) and the importance-rating task ($\hat{\kappa}_G$). This correlation is statistically significant with a p-value of 3.8×10^{-6}. This is to some degree expected as the two tasks ask the participants to perform similar tasks centred around identifying the important points in the text. No statistically significant correlations were observed between the categories-of-importance task and the summarisation or importance-rating tasks.

6.4 Conclusions and future work

This chapter presented a follow-up study to Chapter 4 investigating the boundary between using NLP tools to supplement and replace human analysis in the domain of energetic materials. Recommendations and insights from Chapters 4 and 5 were used to develop a study that compared the ability of NLP tools, experts and novices to analyse a single document in the domain of energetic materials (the text) according to a subjective concept (importance). This study had two aims (1) to compare the ability of NLP tools, novices, and experts to identify important points in the text and (2) to assess the agreement amongst experts and novices on how they evaluate the importance of technical information in the text.

Section 6.1 contained an overview of inter-rater reliability (IRR) metrics. For this thesis, it was determined that IRR metrics would be interpreted as the extent to which participants agree. First, popular metrics such as Cohen's kappa, Fleiss's kappa and the highly flexible Krippendorff's alpha were introduced. The limitations of these metrics, namely the kappa paradoxes, were discussed. Next, Gwet's $AC1$, another flexible metric developed to overcome the kappa paradoxes, was introduced. Finally, methods for interpreting IRR metrics, such as the Landis and Koch benchmark scales, were discussed.

In Section 6.2, a detailed overview of the methodology was given. This methodology centred around a three-part questionnaire given to a group of 25 readers and an expert (collectively referred to as the participants). The inclusion criteria for this study was described. An overview of the questionnaire was given. It was described how the participants were asked to perform exactly the same task as an extractive-summarisation algorithm and highlight the points they felt were significant in the text. This task addresses aim (1) of this chapter. Third, a description of the statement-analysis task was given. It was described how the participants were asked to rate the importance of and assign a category oof, and assign a category of importance to, 30 statements derived from the keypoints/summaries generated by the NLP/participants in Chapter 4. This task addresses aim (2) of this chapter. Finally, evaluation methods were introduced. $RAIS'$ is an intra-group agreement metric that measures how important the readers as a group found each specific sentence in the text. F_A is an inter-group agreement metric that gives the average overlap between a pair of summaries. It was determined that Gwet's $AC1$ ($\hat{\kappa}_G$) would be used to measure agreement among participant responses in the statement-analysis task.

In Section 6.3.1, an overview of the responses to the background-information questions from the 25 recruited readers was given. Results revealed the majority of the readers are in the early stages of their career. However there was a sizeable proportion of readers who were experienced. The majority of recruited readers had STEM PhDs. The self-assigned expertise level and the total number of years experience of the readers were determined to be correlated with moderate strength. This indicates that, as expected, the majority of more experienced readers rated themselves as having a higher expertise level than those with less experience. However, it is recommended that future studies should ask specific questions about the reader's knowledge of and experience in energetic materials as a discipline in addition to their general experience.

Overall, in comparison to the study conducted in Chapter 4, a significantly more varied group of readers was recruited for this study. However no readers with experience in government policy were recruited. Future studies may wish to recruit participants with experience in government policy as to get a full comparison of how individuals in defence and security-related fields interpret importance.

In Section 6.3.2, the results of the summarisation task were presented. Here, a comparison of the ability of NLP tools, novices and experts to identify important points in the text was conducted. The inter-group analysis determined the F_A average with the readers is lower for the NLP than for the majority of other participants. Similarly, the intra-group analysis found no statistically significant difference in how important the participants as a group found sentences the NLP identified and did not identify as important. However, every participant identified at least one of the same sentences as the NLP. This was due to the NLP returning conclusion-style sentences, which tended to be identified as important by the participants. Overall it is concluded that in a data-limited environment, NLP algorithms cannot effectively identify important points in a domain-specific document in the same way an expert can.

In addition, both inter and intra-group analysis revealed there was a significant variation in reader responses. This resulted in considerable variation in F_A with values in the range of 0 to 0.93 observed. This was expected because of the broad spectrum of readers recruited. Similar variations in reader responses were observed in the study in Chapter 4, where readers were asked to summarise the important points in the text. It was hypothesised that the readers' experiences impacted how they approached this task. Results revealed a weak-negative correlation between the difference in higher-education qualifications and F_A. Therefore it was concluded that readers with the same higher-education qualifications tend to identify a higher proportion of the same sentences in the text as important. Due to the sheer variation in reader responses, reader F_A values with the expert varied considerably. However, the intra-group analysis determined the readers were more likely to identify a sentence as important if the expert also identified the statement as important. This result was found to be in agreement with the results found in Chapter 4 which similarly determined the readers as a group rated higher on average and found more important reader-agreed points that were in agreement with the expert.

Furthermore, it was observed that the two novice participants in this study had very different approaches to this task. Novice Reader 4 highlighted sentences exclusively in the introduction

and conclusion of the text. This novice had a low agreement with the other readers and the expert. However, novice Reader 15 obtained the highest F_A with the expert's summary. One reason for this is that, unlike the technically-trained readers who would have approached the task with preconceived ideas on how to identify important sentences in the text, the novice's perception of importance in this domain may have been entirely shaped by reading the text. Therefore, by following semantic cues in the text, the novice may have identified the sentences in the text that the expert found important.

In Section 6.3.3, the results of the statement-analysis task were presented. Here, the agreement amongst experts and novices on how they evaluate the importance of technical information in the text was assessed. The statement-analysis task consisted of two parts: an importance-rating task and a categories-of-importance task. The analysis in this section determined there was considerably more agreement on the ratings-of-importance task than on the identifying-the-category-of-importance task. On the categories-of-importance task, none of the participant pairings had a greater than moderate agreement. This was in direct contrast to the rating task, which had 54% of pairings at these agreement levels. Therefore it was concluded that the participants agree with each other less on the reasons why a particular statement is important than they do on how important a statement is. This may be because there are various valid reasons why a statement is important, and the assignment of these reasons may vary considerably by field.

Similarly, it was determined that agreement with the expert on the importance-ratings task was higher than on the categories-of-importance task. For the importance-ratings task, $\hat{\kappa}_G$ in the range of 'fair' to 'substantial' were obtained, whereas, for the categories-of-importance task, values in the range of 'poor' to 'moderate' were obtained. The intra-group analysis found that, on average, statements the author rated as important were more important for the readers than those rated not important by the expert. This is in agreement with the results of the summarisation task. It is therefore concluded that the readers largely agreed with the expert as a group. The Intra-group analysis of the categories-of-importance task identified that the readers as a group agree with the expert to a moderate degree on assigning categories of importance.

In addition, analysis in Section 6.3.3 investigated the hypothesis that the reader's experiences impacted how they approached the tasks. For the importance-rating task, a weak-positive

correlation between the difference in the number of years of academic experience and $\hat{\kappa}_G$ was observed. One possible explanation is that the longer the time spent in academia, the more deep-rooted opinions on what is important become and, therefore, the lower the agreement. In the categories-of-importance task, weak-negative correlations were observed between the difference in the total number of years of experience in commercial research, the number of years of experience in government research and the total number of years of experience with $\hat{\kappa}_G$. Therefore, similar numbers of years of experience in each of these fields results in higher agreement when assigning categories of importance. For both the categories-of-importance task and the summarisation task there was an association between agreement and the similarity of the highest STEM qualification obtained of the readers.

As observed for the summarisation task, there was a considerable difference in the approaches the two novice participants took. Novice Reader 4 obtained one of the lower $\hat{\kappa}_G$ values with the expert for the importance-rating task. On the other hand, novice Reader 15 obtained the highest agreement with the expert with a score of 0.72. However, on the categories-of-importance task the novice readers obtained significantly lower $\hat{\kappa}_G$ values with the expert. This may be because Reader 15 could be use semantic clues in the text to identify which specific sentences in the text are important and what the expert considers to be important in the text. However, understanding and identifying why a specific point is important requires a complex content schema that a novice will not possess and thus agreement on the categories-of-importance task was lower. Future studies should recruit a larger group of novices to observe how novice behaviour differs.

In Section 6.3.4, a comparison of the agreement across all three tasks was conducted. It was determined that there were weak-positive correlations for the agreement on the summarisation task and the importance-rating task only. This expected to some degree expected as the two tasks ask the participants to perform similar tasks centred around identifying the important points in the text. However, no statistically significant correlations were observed between the categories-of-importance task and the summarisation or importance-rating tasks.

To conclude, this chapter aimed to (1) compare the ability of NLP tools, novices, and experts to identify important points in the text and (2) assess the agreement amongst experts and novices on how they evaluate the importance of technical information in the text. Results concluded that, in a data-limited environment, NLP algorithms could not effectively identify important

points in a domain-specific document as an expert can. Furthermore, there is considerable variation between readers in even simple tasks such as identifying important points in a single technical document. In this study, a novice with no STEM background obtained the highest agreement with an expert on identifying important points in the text and rating the importance of statements about the text. The lack of agreement among experienced readers leads to two fundamental points. Firstly, as pointed out in Ref. [375], policymakers should rethink how they view expert opinions and apply the same critical analysis that is applied to data. Secondly, if experts do not agree on basic points, it brings the reliability of the data used to train algorithms into question. Chapter 9 will explore this further by presenting a methodology to evaluate the reliability of human annotations.

Chapter 7

Developing a preprocessing pipeline for the energetic-materials domain

Chapter 3 identified the most fundamental barrier to successfully implementing an ontology-learning system in the energetic-materials domain as the preprocessing and extraction of text from text-sources. As illustrated in Chapter 4, preprocessing can significantly impact the performance of downstream tasks [376]. Therefore, developing an effective preprocessing pipeline is essential not just for developing an ontology but for any future NLP-based research in the energetic-materials domain. This chapter, therefore, aims to identify the challenges that preprocessing of text in the energetic-materials domain presents and develop a novel general-purpose preprocessing pipeline catered for small datasets of conference proceedings in the energetic-materials domain.

Chapter achievements:

- Identification of the challenges that preprocessing PDFs of conference proceedings in the energetic-materials domain presents.
- Novel general-purpose preprocessing pipeline catered for preprocessing small datasets of conference proceedings in the energetic-materials domain.
- In-depth qualitative evaluation of preprocessing pipeline.

Section 7.1 will review the relevant techniques for preprocessing conference proceedings in the

energetic-materials domain. Section 7.2 gives an overview of the methodology to construct this preprocessing pipeline. Evaluation and discussion of the performance of the pipeline on a test set of documents is conducted in Section 7.3. Finally, in Section 7.4, the conclusions of this chapter will be presented alongside suggestions for future work.

7.1 Preprocessing conference proceedings in the energetic-materials domain

In this section, a literature review with the aim of identifying problems with preprocessing text from a small corpus of conference proceedings PDFs in the energetic-materials domain will be conducted. The appropriateness of existing techniques for various parts of the preprocessing pipeline will be discussed.

There is only one open-source publication that mentions preprocessing text in the energetic-materials domain [239]. In this paper, the authors introduce a preprocessing pipeline for patents and proceeding papers from the International Detonation Symposium, Fraunhofer Institute for Chemical Technology Annual Conference and New Trends in Research of Energetic Materials (NTREM) abstracts. Whilst influences are drawn from the work done in Ref. [239] there are several primary differences with the approach presented in this chapter.

Firstly, the approach presented in this chapter uses a significantly smaller quantity of data. As a result, methods that may not be suitable for a more extensive dataset can be considered. Secondly, the scope of this pipeline is limited to extracting text from PDFs of conference proceedings from NTREM and High Energy Materials Conference and Exhibits (HEMCE). Thirdly, the purpose of developing the pipeline is different from that of the pipeline developed in Ref. [239] which is specifically catered for identifying energetic materials and their application words. The pipeline in this chapter is a general-purpose pipeline applicable for a range of downstream tasks. Similarities and differences between the two pipelines will be highlighted throughout this chapter.

A preprocessing pipeline typically consists of four stages: text extraction, tokenisation (the identification of sentence and word boundaries), text normalisation, and what in this thesis will be referred to as 'task-specific processing'. The purpose of this chapter is to develop

a preprocessing pipeline that can be used on a range of different tasks, so any task-specific processing, such as stemming and lemmatisation[1] are not included in this pipeline. Therefore, a literature review for the aforementioned three stages is conducted in Sections 7.1.1 - 7.1.3.

7.1.1 Extracting text from PDFs

The first step in preprocessing conference proceedings is to extract the text from the documents. Conference proceedings typically appear in PDF or HTML format. This literature review will concentrate on text extraction methods for PDFs. There is limited literature on the preprocessing of chemistry journals. However, Ref. [376] provides a thorough overview of issues with preprocessing text from conference proceedings in the domain of chemistry and some tools that resolve these issues.

Extracting text from PDFs, in general, is a difficult task. The PDF is a layout-based format, unlike other formats that specify semantic information about the text (for instance, sentences, paragraphs, headers, footers etc.) PDFs specify the positions and the fonts of characters in the text [377]. This makes tasks such as word identification, word order, semantic-role detection (e.g. detecting if a piece of text is part of the title/header/main text) and paragraph-boundary detection nontrivial. Furthermore, text extraction from conference proceedings PDFs is difficult due to variations in style and formatting, making standard rule-based approaches error-prone. Common errors include incorrect sentence handling in two-column documents and misplacing paragraph separation [376].

There are a range of existing Python packages for extracting text from PDFs. Examples include 'pdftotext' [378], a basic library that struggles to identify paragraph boundaries or the semantic roles of the items in the text, and 'PyPDF2' [379], which can extract meta-data from the PDF such as title and page number but has limited capabilities for performing layout-aware text extraction.

Ref. [305] uses PDF.miner.six [380] for text extraction. PDF.miner.six uses a layout-analysis algorithm that works in stages. The first stage is grouping characters into words and lines. Every character in a PDF has two sets of (x, y) coordinates that can be used to define a box around the character. The layout-analysis algorithm uses these coordinates to determine which

[1] See Section 4.1.1 for details of stemming and lemmatisation.

characters should be grouped into words and lines. PDFs have no notion of the space character; at this stage, spaces are inserted if two characters are far apart. The second stage consists of grouping lines, the output from the previous stage, into boxes. As with the characters, each line can be defined by two sets of (x, y) coordinates that can be used to define a box around the lines. Lines that overlap in x and have similar y values are grouped into text boxes. In the final stage, these text boxes are grouped hierarchically by merging text boxes close to one another. This process gives an order output of the layout of objects on a PDF page [381].

GROBID is a JAVA machine learning (ML) library for text extraction [382]. This library does, however, have a Python client [383]. Unlike the previously described approaches which output text files, the algorithm's output is a structured XML TEI file. Sequence labelling is a supervised-learning task that aims to predict a label for each element in a sequence [384]. GROBID uses several sequence-labelling models based to parse PDFs. Each model maintains a small number of labels specialising in a specific substructure, e.g. the segmentation model is used to detect the main parts of a document. The models can be cascaded to create detailed nested structures that represent the document's overall structure. GROBID uses fast linear-chain conditional random fields and a range of deep-learning models for sequence labelling. A more detailed overview of GROBID can be found in Ref. [385].

Unlike the other aforementioned tools, GROBID is specifically designed for PDFs of scientific literature. In Ref. [377], a comparison of these tools and some not mentioned is concluded. These tools were evaluated on randomly selected scientific articles and compared on four criteria: paragraph boundaries, identification of text and non-body text, reading order and word boundary. In general, GROBID was found to outperform PDFMiner (an earlier version of PDFMiner.six) and pdftotext (PyPDF2 was not studied). Ref. [386] compares GROBID, Pdfminer.six and pdftotext on sentence tokenisation and returning correct sentences on text from the Electric Double-Layer Capacitor Domain. GROBID was found to perform the best.

PDFMiner.six and GROBID are able to perform layout-aware text extraction [380, 382], unlike pdftotext they claim to be able to identify the main body of text [378]. This would allow non-relevant parts of the documents (noise) such as headers, footers, and page numbers to be removed. For smaller datasets, it is important that data is of high quality and noise is reduced. Therefore this makes PDFMiner.six and GROBID the most appropriate choices of the text-extraction tools reviewed.

Initial analysis from the literature suggests GROBID may be the most appropriate choice due to its better overall performance. However, it should be noted the performance of text-extraction algorithms depends significantly on the text they are tested on. Therefore, to obtain an idea of which of the algorithms are the most appropriate, a comparison should be conducted on the conference proceedings from the energetic-materials domain. Conference proceedings in this domain present additional challenges, such as the handling of parsing of equations, chemical diagrams, tables, and chemical names, which include superscripts and subscripts.

7.1.2 Tokenising text

The second step in preprocessing the documents is to perform tokenisation. This step is significantly more complex for text with chemical names and symbols. Tasks relevant for developing an ontology, such as named entity recognition, are sensitive to sentence tokenisation errors [376].

Sentence tokenisation can be defined as the identification of sentences and their boundaries in text. Traditional approaches to sentence tokenisation are rule-based; e.g. for English texts, the most simplistic approach is to identify full stops and capital letters to identify sentence tokens [376]. It is of note that such approaches are inappropriate for character-based languages. Furthermore, as highlighted in Ref. [376], full stops are not always indicators of the end of a sentence, particularly in chemistry-based conference proceedings. Full stops can appear in initials, as decimals, in-text citations, acronyms and most notably in chemical entities (e.g. C34H24N6O14S4.4Na, 5,10,11,16,17-pentaisocyano-3,6,9,12,15,18-hexazatetracyclo [12.4.0.02,7.08,13]octadeca-1(14),2, 4,6,8(13),9,11,15,17-nonaene-4-carbonitrile). As a result, rule-based tokenizers tend to use hand-crafted lists to identify abbreviations.

Recent tools for sentence tokenisation use ML algorithms. Supervised approaches are trained on text annotated with sentence boundaries. Currently, there is no such dataset available for the energetic-materials domain. Most supervised tools for sentence tokenisation used in the chemical domain use tools trained on either general domain corpora or the resource-rich biomedical domain [376]. For instance, the JULIE sentence tokeniser [387] is a widely used JAVA-based tool trained on a biomedical corpus that has been used in the chemical domain [388].

Unsupervised approaches are trained on raw text. The widely used NLTK sentence tokeniser uses the Punkt algorithm for sentence tokenisation [389]. The Punkt algorithm is a language-independent approach that assumes that errors from sentence tokenisation can be mainly eliminated by identifying abbreviations. The algorithm learns to identify abbreviations, collocations (words that commonly occur together) and sentence starters in an unsupervised fashion with the aim of identifying sentence boundaries. The algorithm tends to perform best when trained on the target domain [390]. The NLTK sentence tokeniser is trained on a non-scientific corpus. In Chapter 4, the NLTK tokeniser was used on text in the energetic-materials domain and struggled in the presence of chemical entities and abbreviations. The Punkt algorithm is ideal for applications in the chemical domain, as it is unsupervised, and there are large quantities of unlabelled data but few examples of labelled corpora.

ChemDataExtractor is a recently developed Python toolkit designed for preprocessing chemical documents [247]. This sentence tokeniser is based on the Punkt algorithm trained on text from conference proceedings published in the Royal American Chemistry Society.

Word tokenisation is the process of dividing sentences into their constituent components (words, punctuation and numbers) and is a critical step in a preprocessing pipeline that relies on the identification of word breaks. As observed in Section 4.3.1, word tokenisation can significantly impact downstream tasks. In general, word tokenisers in the English domain rely on identifying punctuation marks and whitespace. This is more complex for chemistry corpora as tokenisers need to be adapted to cope with chemical or mathematical formulae. Chemical entities can contain hyphens, whitespace, punctuation marks, e.g. brackets, super- and sub-scripts and numbers. These features can confuse traditional English tokenisers, making them insufficient for use on chemical text [391]. Furthermore, in scientific fields, it is not uncommon for entities to have complex naming structures or nested terms. Traditional tokenisers will not be able to identify the boundaries of these entities correctly.

Popular chemical tokenisers include the JAVA-based tools OSCAR4 [392], and ChemTok [393]. These tools rely on manually-defined rules to identify word boundaries. ChemDataExtractor is a Python package that contains a word tokeniser adapted to the chemical domain. The base of the tokeniser is similar to the widely used English tokeniser, the Penn Treebank tokeniser [394]. The ChemDataExtractor tokeniser is adapted to the chemical domain by splitting most tokens on the basis of whitespace and punctuation characters except for brackets, and colons

[247]. The tokeniser also implements several custom rules.

Most of the aforementioned tools are JAVA-based. Given the pipeline presented in this chapter will be developed in Python, ChemDataExtractor will be used.

7.1.3 Normalising text

Text normalisation removes inconsistencies in the text that may later influence the performance of downstream tasks [247]. Standard text-normalisation procedures include correcting human-based errors, for instance, spelling errors or Homoglyph substitution. This is when a character or symbol is mistaken for a similar looking character/symbol, e.g. '1' for 'l' or '0' for 'o' [376].

As noted in Section 3.6, the energetic-materials domain requires some additional steps for normalisation. Acronyms and abbreviations are frequently used in the energetic-materials domain. Mapping terms to their variants will often reduce the size of a numeric representation of the text and will result in improved semantic representations. Abbreviations often have domain-specific meanings; therefore, exploring domain-specific solutions makes sense. Existing approaches to abbreviation mapping include the use of hand-crafted domain-specific lists. The development of these lists is a time-exhaustive and expensive solution. It is also noted that, even within the same domains, the same abbreviation can denote different things. For instance, the acronym 'IR' can be used to denote infra-red or the chemical name 'Iridium'.

ChemDataExtractor has a tool for detecting definitions of chemical abbreviations. This tool is based upon the algorithm described in Ref. [395]. The algorithm aims to identify abbreviations by searching for an unabbreviated word followed by an abbreviated word or an abbreviated word followed by unabbreviated word. On identification of one of these pairs, the algorithm examines the word pair. For instance, an unabbreviated word is only considered valid if it consists of a maximum of two words and its length is between 2-10 characters. Full details of the algorithm can be found in Ref. [395]. This simple algorithm was developed for identifying acronyms in biomedical text and claims to obtain a precision of 92% on a corpus of MEDLINE articles. No other tools for the chemistry domain were found.

Chemical entities can appear in the text in various forms, for instance: molecular formulae, IUPAC names, Canonical SMILES or even CAS numbers. However, as with abbreviations, these

chemical entities must be mapped to a single format. In order to do this, chemical entities in the text must first be identified. This task is referred to as chemical entity recognition (CER).

Refs. [376, 396] provide a thorough overview of CER. Approaches to entity recognition have been discussed in Section 3.3.1. CER approaches tend to be either dictionary-based, ML-based or statistical. To develop and test such a technique requires resources that do not currently exist in the energetic-materials domain, e.g. a labelled corpus or a parsable dictionary. The best-performing approaches to CER tend to combine dictionary and rule-based methods with ML-techniques [247]. CER is a research area in its own right. Given the purpose of this pipeline is to preprocess text for a range of different downstream tasks, only preexisting tools are considered.

There are a range of different off-the-shelf CER-tools, to name a few: BANNER-CHEMDNER [397], ChemEx [398], OCSCAR4, LeadMine [399], chemtagger [400] and Chemspot [401]. The scope of the investigation is limited to open-source tools that can be implemented in Python. This leads to the ChemDataExtractor package. The ChemDataExtractor package is an off-the-shelf CER Python package. It has a hybrid approach combining a CRF-based recogniser with a dictionary-based recogniser that uses a wordlist from the 'Jochem' chemical dictionary and an automatic filtering process based on the method used in LeadMine [399].

In Ref. [239], the package ChemDataExtractor was used to identify chemical names in the text. Whilst the paper makes no specific reference to how well the algorithm performed, it appears to allow successful implementation of downstream tasks in the energetics domain. In Ref. [402], this tool was evaluated against a range of other off-the-shelf NER-tools and was found to outperform the tools on biomedical text with a precision of 88%. In Ref. [247], Chemmtagger achieved an F score[2] of 88% on the CHEMDNER corpus.

Common errors with CER systems include sensitivity to spelling and tokenisation errors [403, 401]. In addition, the CER systems can be confused by identifying part of a chemical entity instead of the complete chemical entity [376]. Qualifiers are a cause of this; for instance, 'Ammonia Aqueous' could be misidentified as 'Ammonia'.

Once chemical entities have been identified, they must be mapped to a consistent form. This is typically done through the use of a dictionary or gazetteer[3]. Ref. [376] provides a list of

[2]This metric was introduced in Eq. (3.6).
[3]A gazetteer is a list of entities and their synonyms.

chemical gazetteers, most of which are relevant to the biomedical domain. Chemspider [404] is a popular chemical-structure database that enables retrieval of chemical properties and associated synonyms; it also has a Python interface [405]. PubChem [406] is the world's most extensive dynamic collection of freely accessible chemical information; like Chemspider, PubChem enables retrieval of synonyms and molecular formulae and has a Python interface [407]. There is no currently open-source database specific to the energetic-materials domain. One possible error with dictionary lookup is that the dictionary may not contain the chemical entity. This error seems unlikely if the dictionary method is comprehensive enough.

7.2 Methodology

The literature review in the previous section identified the following key challenges: layout-aware text extraction and parsing of chemical names. In addition, tools to overcome these challenges were identified. This section presents the methodology used to develop a preprocessing pipeline catered for preprocessing small datasets of conference proceedings in the energetic-materials domain. Figure 7.1 gives an overview of the preprocessing pipeline.

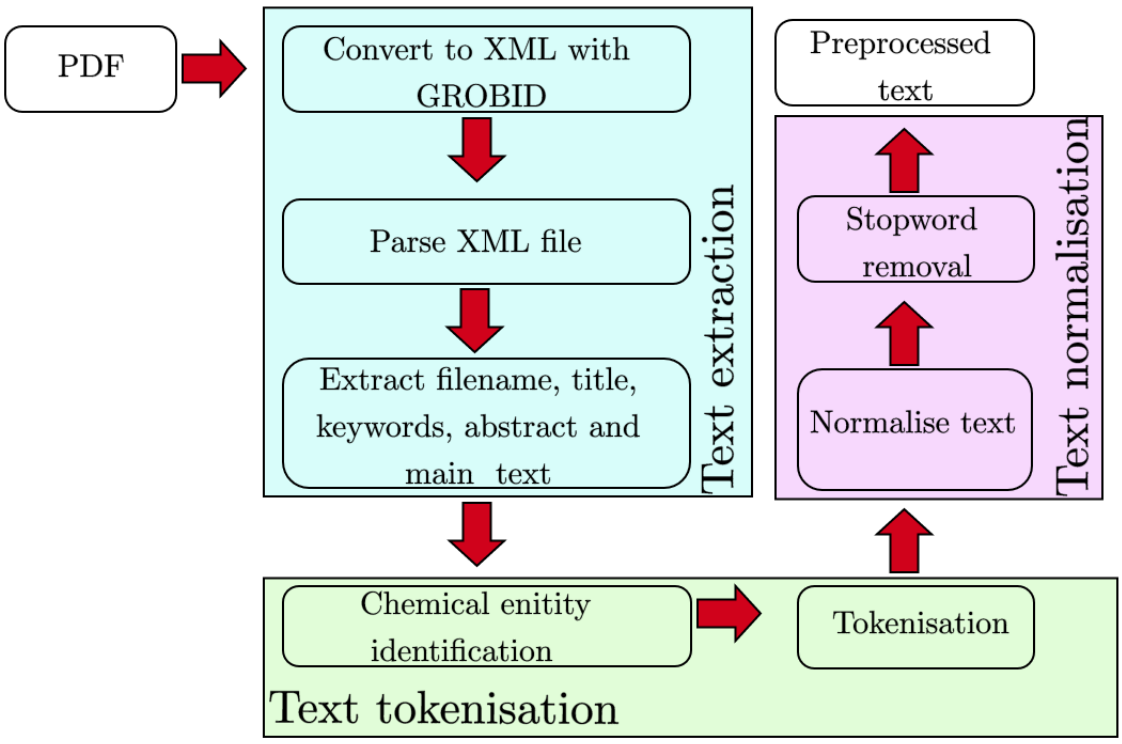


Figure 7.1: Overview of the proposed preprocessing pipeline, consisting of three main stages: text extraction, tokenisation and text normalisation.

The pipeline can be divided into three parts: text extraction, tokenisation and normalisation. Sections 7.2.1 - 7.2.3 will discuss details of each of the aforementioned parts of the pipeline. Following this, methods to evaluate the performance of the pipeline will be discussed in Section 7.2.4.

7.2.1 Text extraction from PDFs

The literature review in Section 7.1.1 determined GROBID was the most appropriate tool of those reviewed for extracting text from PDFs of conference proceedings in the energetic-materials domain. However, it was recommended that since no direct comparison between text-extraction tools has been conducted on energetic-materials text in the literature, these tools should be tested to determine the best-performing tool. Therefore, PDFMiner.six and GROBID were tested on a small sample of randomly selected NTREM and HEMCE conference proceedings. An evaluation was conducted using qualitative analysis with the criteria (1) extraction of full text without headers, footers and page numbers and (2) correct parsing of chemical names in mind.

PDFMiner.six did not perform semantic-role detection effectively - the package failed to distinguish between the body of the text, headers, footers, references and page numbers. For the NTREM conference proceedings, it was found that PDFMiner.six failed to extract large quantities of the main body of the text for some documents. This was not observed for the HEMCE conference proceedings. PDFMiner.six effectively parsed most chemical names but added unnecessary whitespace between chemical names with hyphens.

The output of GROBID's text-extraction tool is a TEI file. Since the desired output format is raw text, this file must be parsed. GROBID has a module for parsing TEI files; however, this module did not perform well with the conference proceedings used in this study, sometimes failing to return the full text. It was therefore decided to use the Python package XML.ETree to write a custom script for parsing the TEI files. It was found that a straightforward script could parse both HEMCE and NTREM conference proceedings effectively. Using a custom script meant the abstract, title, author keywords and the main body of text could be effectively extracted separately.

The GROBID tool effectively performed semantic-role detection, it returned the body of the

text without headers, footers, and page numbers. However, the GROBID tool struggled to parse super- and sub-scripts, adding unnecessary whitespace between the tokens resulting in unusually separated chemical names[4]. This seperation is confusing to word tokenisers. For instance, 'H_2O' became 'H 2 O'. This error could be corrected at the tokenization stage, whereas the error of leaving out large quantities of text associated with the PDFMiner.six tool cannot be resolved. It was therefore decided to use GROBID with a custom script for parsing the TEI file for extraction of text from the PDFs. Figure and table captions were included in the extracted text as they were found to contain semantically-meaningful information.

The output of this stage of the preprocessing pipeline is the raw text of the main body of the articles, abstracts and author-selected keywords. This was stored in a Pandas data frame[5].

7.2.2 Chemical entity recognition and tokenisation

The next step in the pipeline is to perform tokenisation of the text; however, as mentioned in Section 7.2.1 incorrect parsing of super- and sub-scripts results in errors in parsing chemical names. This error needs to be resolved at the tokenisation stage. To do this, chemical entities in the text need to be identified.

The approach used in Ref. [239] to identify chemical entities in the text was adapted to this pipeline. Ref. [239] first used ChemDataExtractor to identify chemical entities and then verified the identified chemical entity was, in fact, a chemical entity by performing a PubChem search using its Python interface.

To develop the pipeline in this chapter, first ChemDataExtractor's chemical-entity detector was run on the extracted text to obtain a list of chemical entities in each paper. This method was found to have high recall of chemical entities but low precision. Non-chemical entities were frequently incorrectly identified as chemical entities, e.g. it incorrectly identified 'cook-off' and 'shock energy' as chemical entities. The chemical-entity detector identified a total of 33,814 unique chemical entities in a corpus consisting of 1160 NTREM papers.

To verify if the chemical entities were in fact chemical entities, a search of PubChem using its Python interface was performed. For each element in the list, the following process was

[4]This is mentioned as an open issue on the Github of GROBID.
[5]This stage of the preprocessing pipeline is noticeably different from the pipeline in Ref. [239].

followed: first, a compound search was performed by name; if no result was yielded, then the search was repeated by formula, then CID (compound ID), IUPAC name and finally SMILES. If this yielded no result, a substance search was performed first by name, then by formula, SID (substance ID), IUPAC name, and finally SMILES. This iterative approach is very time consuming and is only suitable for small datasets. If a chemical entity was verified, its synonyms and molecular formula were identified using the PubChem Python interface and were stored in a local dictionary. This analysis reduced the number of unique identified chemical entities that had not been verified from 33,814 to 26,168.

Qualitative analysis was then conducted on the 26,168 unverified chemical entities to identify why PubChem did not identify them as chemical entities. The primary reason was ChemDataExtractor incorrectly identifying names, formulae etc. as chemical entities. In addition, many possible entities had incorrect insertion of whitespace, mainly in the form of double whitespace. Therefore, potential entities in the list with double or triple whitespace were identified and had the excess whitespace removed before rerunning the Pubchem searches described above. Unverified entities with hyphens in had the hyphens removed and the PubChem searches were rerun. This was done assuming that chemical names with hyphens such as 'C-H' would have already been identified in the previous steps of the pipeline. This analysis reduced the number of identified chemical entities not verified from 26,168 to 22,001.

Another cause of PubChem's inability to identify chemical entities included typos in chemical names. Although this could be resolved with a custom dictionary of chemical elements and common misspellings, developing such a list would be very time consuming. Furthermore, PubChem did not identify some valid chemical names as they are not in the PubChem database e.g. '2-tert-butylnitroamino-4,6-bis(trinitromethyl)-1,3,5-triazine'.

For each paper, a dictionary containing chemical entities as they appear in the text and their corrected form (e.g. without whitespace) is used to remove parsing errors introduced by parsing via GROBID. In addition, the whole text is searched to remove double and triple spaces. Finally, sentence and word tokenisation is performed using ChemDataExtractor chemically-aware tokenisers.

The output of this part of the text tokenisation pipeline is an updated data frame containing the tokenised full text, abstract and author keywords and a chemical dictionary containing all identified chemical entities in the text, their synonyms and chemical formulae.

7.2.3 Text normalisation

The final stage of the preprocessing pipeline is to normalise the text.

First, in-text citations were removed. Then, the issue identified in Section 7.1.3, whereby CER systems can be confused by identifying part of a chemical entity instead of the entire chemical entity, is addressed. All combinations of consecutive tokens in the range of 2 to 6 were merged into a single token. i.e. consider the tokenised sentence ['Hydrogen', '-peroxide', 'was', 'added']. In this case, the following merged tokens would be generated: 'Hydrogen-peroxide', '-peroxidewas', 'wasadded', 'Hydrogen-peroxidewas' etc. If the merged token was found to have an exact match in the chemical dictionary, then the single tokens were replaced with the merged token in the text. This also resolves errors in CER associated with tokenisation. This may result in an error in rare instances, but the probability of consecutive chemical names being incorrectly merged to form another chemical name seems rare given the presence of punctuation.

Following this, every chemical entity in the text was mapped to a single form, using the chemical dictionary generated in Section 7.2.2. It is noted this meant that qualifiers adjacent to chemical elements were ignored. For instance, 'titanium powder' was mapped to 'titanium'.

A general search in the text was then performed, whereby any identified tokens in the dictionary were mapped to the root form of the chemical entity. It was quickly realised, however, that this approach meant chemical names with chemical symbols that are also common words in the English language, such as arsenic (As) and indium (In), proved problematic. This step meant commonly occurring English words were mapped to chemical elements. In addition, it also meant acronyms with the same form as chemical symbols were also incorrectly mapped to chemical entities. It was, therefore, decided to add a caveat to this step whereby this step only took place if the word being considered was not in a custom stopword list. The custom stopword list was created by combining Genism's English stopword list [408] with punctuation marks, numbers, percentage signs and common phrases, e.g. 'fig.', 'table', 'No.'.

As mentioned in Section 7.1.3, ChemDataExtractor does have a function for identifying acronyms in the text. However, manual inspection of this function on a small subset of documents found it was very imprecise. It would often identify acronyms not present in the text. For instance, identifying 'NOZZLE' as an acronym for the phrase 'the rocket propellant was added to'. Therefore, it was determined that this function would not be used in the pipeline.

Different text-representation methods require different techniques for downstream processing. For instance, BERT requires minimal processing and contains an in-built tokeniser. Therefore, the data frame contained two versions of the processed text: a detokenised version (obtained by performing detokenising using NLTK's Treebank dentokenisers [311] suitable for BERT, and a tokenised version with the stopwords removed for alternative feature-representation methods.

7.2.4 Pipeline evaluation

The purpose of this pipeline is to extract text from conference proceedings to allow for downstream NLP tasks such as classification. The ultimate evaluation of the pipeline's performance is if it allows for downstream tasks to be completed effectively. This will be tested in Chapters 8 and 9. Nevertheless, in this subsection, a method used to evaluate elements of the pipeline's performance on two small datasets will be detailed.

The two datasets used to evaluate the performance of the pipeline consists of 10 randomly-selected papers from each of the HEMCE and NTREM proceedings. These are henceforth referred as the HEMCE and NTREM test sets. The papers in each test set vary in length from 2 to 10 pages. Evaluation of the pipeline will be conducted after the normalisation stage before the text has been detokenised/has stopwords removed. Conducting an in-depth analysis using small sample sizes is an obvious limitation of this method. However, a full-scale in-depth analysis of the performance of the pipeline would require a large annotated corpus. No such resource exists for the energetic-materials domain.

The pipeline will be evaluated against its ability to return correct sentences and correctly identify chemical entities in the text. For the former criteria, the average percentage of correct sentences returned by the pipeline $S_{\%A\cap B}$ and the average percentage of incorrect sentences retrieved, $S_{\%A\cap B'}$ are introduced as evaluation metrics.

The metric $S_{\%A\cap B}$ is defined as

$$S_{\%A\cap B} = \frac{1}{N}\sum_{i=1}^{N}\frac{A_i \cap B_i}{B_i}, \tag{7.1}$$

where B_i is the set of ground-truth sentences[6] in the main text and abstract for document i, A_i

[6]The actual sentences in the PDF.

is the set of sentences returned by the pipeline, therefore $A_i \cap B_i$ is the total number of correct sentences retrieved by the pipeline in document i, and N is the total number of documents.

The metric $S_{\%A\cap B'}$ is defined as

$$S_{\%A\cap B'} = \frac{1}{N}\sum_{i=1}^{N}\frac{A_i \cap B_i'}{A_i}, \tag{7.2}$$

where $A_i \cap B_i'$ is the set of total number of incorrect sentences retrieved in document i, i.e. those found in A_i but not B_i.

To evaluate the performance of the pipeline against its ability to identify chemical entities in the text correctly, the following metric is introduced: the average percentage of chemical entities returned by the pipeline $CE_{\%A\cap B}$ across a set of N documents, i.e.

$$CE_{\%A\cap B} = \frac{1}{N}\sum_{i=1}^{N}\frac{CE_{A_i\cap B_i}}{CE_{B_i}}, \tag{7.3}$$

where CE_{B_i} is the set of the ground-truth chemical entities in the main text and abstract of document i, and $CE_{A_i\cap B_i}$ is the set of correct chemical entities retrieved in document i[7].

The average percentage of unique chemical entities returned by the pipeline correctly $CE_{\%A*\cap B*}$ across a set of N documents is defined as

$$CE_{\%A*\cap B*} = \frac{1}{N}\sum_{i=1}^{N}\frac{CE_{A*_i\cap B*_i}}{CE_{B*_i}}, \tag{7.4}$$

where CE_{B*_i} is the set of the ground-truth chemical entities in the main text and abstract of document i. In this set, each element can appear only once. The set of incorrect chemical entities retrieved in document i is denoted by $CE_{A*_i\cap B*_i}$. This is the number of unique chemical entities in the PDF also found in the preprocessed text.

To summarise, $CE_{\%A\cap B}$ is a measure of how well the pipeline performs at identifying the frequency of chemical entities, which is important for some text-representation methods, whereas $CE_{\%A*\cap B*}$ is a measure of how well the pipeline performs at identifying different types of chemical entities.

[7]This is defined as the number of chemical entities in the PDF that are also found in the preprocessed text.

7.3 Evaluation and discussion of pipeline performance

In this section, general observations on the performance of the pipeline described in Section 7.2 will be discussed then a more detailed evaluation on two test sets will be conducted.

The developed pipeline did not handle units in the text consistently. For example, across the NTREM corpus, several different versions of 'degrees Celsius' were seen: ('°' 'C'), ('°C'), ('° C'), ('°' 'Celsius'), ('° Celsius'), ('°Celsius'), ('θ C'). Future work to correct this depends on the downstream task. If units are deemed to be of little significance to the downstream task, it is recommended to remove units from the text using a custom stopword list. On the other hand, if the units are deemed to be relevant to the downstream task, a dictionary-based approach could be used to normalise the units in the text.

Furthermore, it is common practice in the energetic-materials domain to label complex chemical names. For example, in one article, chemical entities were referred to as compounds A, B, C, etc. The pipeline developed in this chapter could not map the labels to the chemical names. One solution would be to create a dictionary mapping the label to the chemical name for each paper. However, developing an algorithm to identify this mapping for each paper would be error prone as there is no set convention for labelling. Moreover, developing such a dictionary by hand would be very time consuming.

A further observation is that qualifiers in front of chemical names were often either not recognised as part of the chemical name or when recognised as part of the chemical name, were mapped to the base chemical entity. Whether or not this is positive depends on the downstream task/representation method. For example, in some applications distinguishing between 'Titanium' and 'Titanium powder' may not be helpful to the overall goal.

Also, the HEMCE proceedings had a more significant variation in formatting than the NTREM proceedings as they included abstracts for papers that did not submit in time and, in some cases, incomplete papers. These incomplete papers and abstracts were not subject to peer review and, as a result, some were poorly formatted; for instance, the font changing style and size mid sentence. There were also more spelling and typographical errors. Despite the formatting issues, the developed pipeline successfully extracted text from the PDFs of the proceedings, even in the aforementioned instances of poor formatting. However, the typographical and spelling errors did result in errors in the tokenisation of sentences and words.

For each of the HEMCE and NTREM test sets, each sentence in the PDF of each document was copied and pasted into a Microsoft Excel table. The chemical entities in each sentence were recorded along with the frequency of occurrence in the sentence. This process was then repeated using the text generated by the pipeline. Using these values, metrics given in Eqs. (7.1)-(7.4) were calculated for each test set. The results are summarised in Table 7.1.

Test set	$S_{\%A\cap B}$	$S_{\%A'\cap B}$	$CE_{\%A\cap B}$	$CE_{\%A*\cap B*}$
NTREM	97 %	1%	85%	93%
HEMCE	97%	1%	88%	96%

Table 7.1: Average percentage of correct sentences returned by the pipeline ($S_{\%A\cap B}$), incorrect sentences retrieved ($S_{\%A'\cap B}$), chemical entities returned by the pipeline ($CE_{\%A\cap B}$), and unique chemical entities returned by the pipeline ($CE_{\%A*\cap B*}$) for the HEMCE and NTREM test sets.

In Table 7.1 the $S_{\%A\cap B}$ column shows a high percentage of correctly returned sentences by the algorithm for both test sets (97%). A manual inspection of the documents identified the causes of error. These causes of error included diagrams of chemical structures containing text confusing the GROBID parser. This led to some sentences not being returned. For instance, in Ref. [409], there are several large diagrams of chemical structures which do not have figure captions. GROBID mistakenly parsed some of the body of the text surrounding these diagrams as either the figure caption or part of the headers/footers. Whilst the pipeline did return figure captions, it deliberately filtered out headers and footers. This led to some of the main body of text not being returned by the pipeline.

Another cause of error was generated by roman numerals. Roman numerals are used in chemical text to label chemical entities, e.g. referring to a chemical entity as compound V or even in chemical names. For instance, methyl carbanilide is sometimes referred to as Acardite II. In some instances, the presence of roman numerals caused the sentence tokeniser to insert a sentence break incorrectly. This resulted in some sentences not being returned correctly. Furthermore, typographical errors whereby an extra full stop was inserted by mistake (e.g. '.One' instead of 'one') also resulted in incorrect sentences being returned.

Table 7.1 shows the $S_{\%A\cap B'}$ values for both test sets are very low (1%). This positive result indicates the pipeline can successfully perform semantic-role labelling. The only incorrect sentences that were returned were footers found in some of the NTREM 2007 proceedings after the presence of chemical-structure diagrams. The pipeline performed equally well on both journal types indicating it is adaptable to different styles and formatting.

In addition, Table 7.1 shows $CE_{\%A*\cap B*}$ values of 93% and 96% were obtained for NTREM and HEMCE test sets respectively. This indicates the pipeline performed better at identifying different chemical entities for the HEMCE test set. Manual inspection revealed most errors occurred due to incorrect tokenisation, which could not be resolved as the chemical entities did not appear in the PubChem database. The papers included in the test set for NTREM had a higher number of different chemical entities that were not found in the PubChem database than those in the HEMCE test set hence resulting in a lower value of $CE_{\%A*\cap B*}$.

$CE_{\%A\cap B}$ values of 85% and 88% were obtained for NTREM and HEMCE test sets, respectively. These values are similar to those found in the literature [402, 247]. The smaller value obtained for the NTREM test set is expected given the lower value of $CE_{\%A*\cap B*}$ for NTREM since these two metrics are related. If a particular chemical entity is not identified as appearing once in the text, it will have a low frequency of identified occurrence. Errors included misspelling chemical names, e.g. 1-nitronphtalene, and the chemical entities in the text not appearing in PubChem. If PubChem does not recognise a chemical entity, additional preprocessing steps, such as correcting tokenisation errors, are not applied. This may result in the entity not appearing in the correct form, thus lowering the value of $CE_{\%A\cap B}$. To increase the performance of the pipeline, spelling errors could be corrected using chemistry-based spell checkers.

There are a few noticeable design limitations of the pipeline that should be mentioned. This pipeline was designed to only process PDFs of conference papers and focused on extracting the main body of the text only. Information in tables and figures is ignored. Both GROBID [382] and ChemDataExtractor [247] have tools for extracting data from tables; these tools could be used to increase the performance and capabilities of the pipeline. In addition, Optical Character Recognition (OCR) tools such as Tesseract [410] could be used to transform text in figures into raw text. This pipeline can be easily expanded to extract text from other PDF documents such as patents by writing additional scripts to parse the XML file produced by GROBID correctly.

7.4 Conclusion

This chapter aimed to identify the challenges that preprocessing of text in the energetic-materials domain presents and develop a novel general-purpose preprocessing pipeline for a small dataset of conference proceedings in the energetic-materials domain.

In Section 7.1, a review of the literature surrounding the development of a preprocessing pipeline for the energetic-materials domain was conducted. In Section 7.1.1, challenges associated with extracting text from PDFs of journal articles were identified. It was determined that text extraction in the energetic-materials domain poses additional challenges due to chemical entities and inline formulae. Therefore, tools for extracting text from PDFs were reviewed. This initial analysis determined that GROBID, a layout-aware tool designed to extract text from scientific literature PDFs, would be the most appropriate tool for developing a preprocessing pipeline in the energetic-materials domain. However, since the performance of text-extraction tools depends on the text and no in-domain comparison had previously been completed, a comparison of GROBID and another popular tool (PDF.Miner.six) was recommended.

In Section 7.1.2, it was identified that tokenisation of extracted text in the energetic-materials domain is problematic due to chemical names and symbols and, as a result, traditional rule-based tokenisers do not suffice. A range of sentence and word tokenisers suitable for chemical tokenisation were discussed, but most of the tools were JAVA based. Given this pipeline was to be developed in Python, the only Python-based tool (ChemDataExtractor) was identified as the most appropriate of these tools.

Section 7.1.3 discussed issues concerning normalising text. Mapping abbreviations to their unabbreviated form and mapping chemical entities to the same form is an essential step in text normalisation for text in the energetic-materials domain. The ChemDataExtractor package was found to be a tool identified as useful for this.

Section 7.2 gave a detailed description of the method used to produce the preprocessing pipeline. First, in Section 7.2.1, a qualitative comparison of the performance of GROBID and PDFMiner.six on a small sample of randomly selected NTREM and HEMCE conference proceedings determined that GROBID performed the best at text extraction. However, it was decided to create a custom script to parse the TEI files outputted by GROBID instead of using GROBID's in-built module. Second, in Section 7.2.2, the techniques for chemical entity recognition and tokenisation were described. It was identified that the ChemDataExtractor's acronym tool did not perform well on text in this domain. It was, therefore, not included in the proposed pipeline. Third, in Section 7.2.3, the techniques for text normalisation were described. Finally, in Section 7.2.4, a description of the techniques used to evaluate the pipeline were given. A series of four evaluation metrics were introduced: the average percentage of correct sentences

returned by the pipeline $S_{\%A\cap B}$, the average percentage of incorrect sentences retrieved $S_{\%A'\cap B}$, the average percentage of chemical entities returned by the pipeline $CE_{\%A\cap B}$, and the average percentage of unique chemical entities returned by the pipeline correctly $CE_{\%A*\cap B*}$.

Section 7.3 presented the performance of the developed pipeline on a small subset of 20 hand-annotated papers from NTREM and HEMCE conference proceedings. Overall, the preprocessing pipeline appeared to perform well on both proceedings. However, some issues were identified, e.g. some chemical entities were not in the PubChem database. This is problematic as the preprocessing pipeline relies on identifying the chemical entities using PubChem to normalise the text and resolve tokenisation errors. A suggestion for further work would be to investigate using more advanced chemical entity detection methods for implementation in this pipeline.

Another identified issue is the inability to map abbreviations to their unabbreviated form and the developed pipeline's mishandling of units. Further work should aim to correct this by developing hand-curated dictionaries. Spelling errors were also a cause of error. This could be resolved through the use of chemistry-aware spell checkers. This pipeline could be easily expanded to extract text from other useful PDF documents such as patents by writing additional scripts to parse the xml file produced by GROBID correctly.

To conclude, this chapter introduced a general-purpose preprocessing pipeline for conference proceedings in the energetic-materials domain. Evaluation of the pipeline on a small dataset showed the pipeline performed well on both HEMCE and NTREM proceedings. Whilst the pipeline's performance appears to be promising, the developed pipeline's usefulness will be ultimately determined on its ability to facilitate downstream tasks. This will be examined further in Chapters 8 and 9 whereby the pipeline will be used to facilitate document classification.

Chapter 8

Predicting the Technology Readiness Level (TRL) of journal articles

As discussed in Chapter 1, the underlying theme of this thesis is examining the limitations of NLP tools in small-data environments. Chapters 3 and 4 discussed the approaches to various NLP tasks, such as text summarisation in the data-limited environment of energetic materials. Supervised algorithms were identified as high-performance methods but were ruled out as they require large labelled datasets. As mentioned in Chapter 4, training supervised algorithms on small datasets can lead to overfitting, particularly when the dataset's number of input features (dimensions) is greater than the number of training samples [296], as is often the case with text data. Despite small sample sizes being common in a range of different domains, there are a limited number of papers on the topic [411].

In this chapter, a preliminary study will compare the use of supervised and unsupervised techniques for document classification on a small dataset of NTREM journal articles. The work in this chapter will also evaluate the usefulness of the preprocessing pipeline developed in Chapter 7 by examining its ability to facilitate document classification. The desired outcome of this chapter is to identify the problems that document classification in a data-limited environment presents and summarise a series of guidelines to inform future studies. This study is novel as it is the first study into the NLP-based classification of documents by the inherently subjective concept of Technology Readiness Level (TRL).

TRLs assign a numerical value to the maturity of a technology, allowing for comparison of

maturity across a range of different technologies [412]. TRLs allow for effective communication and understanding between researchers and policymakers [413] and are used in defence to assist decision making in the development and usage of a technology [414]. Using NLP to classify scientific journal articles by TRL is a document classification problem of interest to private industry and government. The UK MoD's definition of TRLs is displayed in Table 8.1 [415].

TRL	MoD Definition
1	Basic principles observed and reported
2	Technology concept and/or application formulated
3	Analytical and experimental critical function and/or characteristic proof of concept
4	Technology basic validation in a laboratory environment
5	Technology basic validation in a relevant environment
6	Technology model or prototype demonstration in a relevant environment
7	Technology prototype demonstration in an operational environment
8	Actual technology completed and qualified through test and demonstration
9	Actual technology qualified through successful mission operations

Table 8.1: UK MoD's definition of Technology Readiness Levels (TRLs) [415]

Chapter achievements:

- Examination of a range of preexisting tools for their suitability for document classification for small datasets.
- Established a series of recommendations applicable to future studies classifying long text in data-limited environments.
- The initial results of this study were published in NTREM 2022 conference proceedings [416].
- Demonstration of the usefulness of the Chapter 7 preprocessing pipeline.

The content of this chapter largely follows that of my published work in the proceedings of the 2022 conference on New Trends in Research of Energetic Materials (NTREM) [416].

In Section 8.1, an overview of the relevant theoretical background behind developing a text classification pipeline in a data-limited environment will be given. In Section 8.2, a summary of the methodology used in this study is presented. In Section 8.3, the results are presented and discussed. Finally, in Section 8.4 the key findings are discussed and summarised.

8.1 Background

Document classification is the process of sorting documents into categories based on the document's content [314]. A document-classification pipeline consists of dataset development, feature representation, model development, model validation and model evaluation. This section presents an overview of the theoretical information relevant to developing a document-classification pipeline in a data-limited environment. In Section 8.1.1, feature representation and transfer learning will be discussed. In Section 8.1.2, classification algorithms will be reviewed and Section 8.1.3 will review model validation for small datasets. Common evaluation metrics for classification will be introduced in Section 8.1.4.

8.1.1 Feature representation and transfer learning

Feature representation for classification is the process of converting text into a numerical representation. In Section 3.3.1, traditional vector representations of words and documents using Term Frequency - Inverse Document Frequency (TF-IDF) (Eq. (3.3)) were introduced. Previous work in Ref. [417] has found that using bigram representations to generate TF-IDF vectors has increased performance on downstream tasks such as classification.

Briefly introduced in Chapters 3 and 4, transfer learning is the process of using knowledge learnt on a previous task to improve the learning of a new somewhat-related task [261]. Pretrained language models[1] such as Bidirectional Encoder Representations from Transformers (BERT) [129] are an effective tool for transfer learning in NLP.

BERT is a transformer-based network that has achieved state-of-the-art performance on various downstream NLP tasks such as text classification and entity recognition. BERT is pretrained

[1] See Section 3.2 for an overview of language models.

on a large corpus. The original 'BERT-base' model was trained on 3.3 billion words in a self-supervised fashion. Contextualised representations of text learnt using BERT capture linguistic knowledge, making them ideal for transfer learning on downstream tasks [418]. Applying BERT to downstream tasks has achieved good results, even for small datasets, and reduces the risk of overfitting [419]. There are two strategies for applying pretrained language representations to downstream tasks: feature-based and finetuning [129].

Finetuning is training a pretrained model on a downstream task. A language model is initialised with the pretrained model's parameters, and all the parameters are fine-tuned using a labelled dataset [420]. This approach introduces minimal task-specific parameters, thus reducing the risk of overfitting whilst ensuring the nuances of the target dataset are learnt. Finetuning deep pretrained models has significantly improved state-of-the-art performance on NLP tasks, and BERT is one of the most effective models for this process [421].

Feature-based transfer learning uses a pretrained language model to generate numerical representations of the text and use the representations in task-specific architectures [129]. Sentence Transformers (S-BERT) is a modification of a pretrained BERT network that uses Siamese and triplet network structures to obtain semantically meaningful sentence/paragraph embeddings [319]. S-BERT offers a range of pretrained models that can be used to generate paragraph embeddings [422].

8.1.2 Classification algorithms

Machine-learning (ML) classification algorithms can be divided into three categories: supervised, unsupervised and semi-supervised. Supervised algorithms use labelled training data, where each input has a corresponding output, to build a model that can predict an output from an input. Unsupervised approaches find patterns in unlabelled data to identify groups of similar samples. Typically, unsupervised approaches do not give a class label. Semi-supervised approaches combine a small portion of labelled data and lots of unlabelled data. In this study, supervised and unsupervised approaches are considered only as these are easy to implement in Python. Semi-supervised approaches to text classification can be found in Refs. [423, 424].

Supervised approaches

There are a range of supervised classification models that are easily implemented using the Python package scikit-learn [425]. This section provides an overview of these approaches.

Logistic regression

The logistic regression classifier is a simple linear discriminative classifier effective at modelling linearly-separable classes. Discriminative classifiers learn which input features are the most effective for distinguishing between classes. Logistic regression is considered the 'go-to' algorithm for classification problems.

The logistic regression model uses the logistic function to model the probability of a feature vector $\vec{X}$ to be a binary output variable y

$$p(y = 1|\vec{X}; w, c) = \frac{1}{e^{-(\vec{X}_i^T w + c)}}, \tag{8.1}$$

where w and c are weights and intercepts of the logistic regression model respectively. Using maximum-likelihood estimation, the optimal w and c values are determined through optimisation. A detailed derivation can be found in Ref. [426], and details of the specific functions optimised in Python's scikit-learn's implementation can be found in Ref. [425].

Logistic regression assumes a non-complex linear decision boundary making it less prone to overfitting than other non-linear classifiers [427]. Identifying models that are less prone to overfitting is important in data-limited environments. Logistic regression is often combined with regularisation, which attempts to overcome overfitting by penalising model complexity [428]. This reduces the magnitude of model coefficients so they are less tailored to the data used to tune model parameters and therefore are better at generalisation [429]. Logistic regression has two primary types of regularisation terms: L1 and L2 [425]. L1 penalises the absolute values of the weights, and it is used for feature selection and shrinkage. L2 penalises the sums of squares of the weights and does not perform feature selection [430]. When c is sufficiently large, the coefficient estimates for L1 will become zero, whereas, for L2, this will go to approximately zero [431].

Support-vector machine

Support-vector machines (SVMs) are another discriminative classification method that are effective in a range of classification problems [432, 433, 434] including document classification problems [435]. SVMs aim to separate training examples according to their classes using a hyperplane that maximises the margin between them, i.e. the distance between the hyperplane to the nearest of each of the points [436]. SVMs maximise the model's generalisation ability [437]. When the data is not linearly separable, kernel functions can be used to map the input data into a higher-dimensional space so that an optimal separating hyperplane can be identified [438]. The complexity parameter C is the sum of the distances of all the points on the wrong side of the hyperplane. C can be considered as the amount of error that can be ignored in the classification [436, 425].

An advantages of SVMs is good performance even when the number of dimensions exceeds the number of samples. In such situations, to avoid overfitting, choosing the correct kernel function and regularisation term is crucial [425]. SVMs are argued to be less prone to overfitting and generalise better than other supervised classification algorithms [439].

Naive Bayes

Naive Bayes (NB) methods are a set of supervised generative (meaning they model the distributions of the individual classes) learning algorithms based on applying Bayes' theorem [425, 440]. Bayes' theorem states that given a class variable y and a set of dependent feature vectors $x_1, \ldots, x_n$, the following relationship exists

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, \ldots, x_n|y)}{P(x_1, \ldots, x_n)}. \tag{8.2}$$

Scikit-learn contains a range of implementations of NB classifiers, the most common for implementation in text analysis is the multinomial NB classifier [425]. This classifier is recommended for discrete features such as word-frequency counts. Multinomial NB is recommended for small sample sizes due to its inherent regularisation making it less susceptible to overfitting [441, 440, 442].

k-nearest neighbours

k-nearest neighbours (k-NN) is a non-linear discriminative learning algorithm [443]. The algorithm identifies the output class of an item by first computing the 'similarity' (via Minkowski

distance) to every other item in the dataset. The k_n most highly ranked samples are the item's nearest neighbours. Voting is then performed using the class labels of the item's nearest neighbours. In theory, the underlying rules in the algorithm can be applied to datasets of small sample size and high dimensions [444]. However, the 'curse of high dimensionality' means distance at high dimensions has little meaning [445]. As a result, k-NN tends to perform poorly with high numbers of features, and small sample sizes [440]. Choosing distance metrics adjusted for high dimensional data such as the one introduced in Ref. [445] can lead to improved performance for textual data with high dimensions; however, some studies have found k-NN is prone to overfitting [446].

Decision tree

Decision tree classifiers are discriminative supervised classification models [425]. A decision-tree algorithm aims to create a model that can predict an outcome by using the input features of the data to make simple decisions. This process consists of forming a tree-like structure [447] by partitioning the input features of the dataset into smaller subsets until some stopping criteria is reached. This splitting process is referred to as recursive partitioning. A range of different metrics can be used to determine the nature of a split, example metrics include the Gini index, and information gain. Ref. [448] gives a comparison between these two metrics. CART (Classification and Regression Trees) [449] is a popular decision-tree algorithm. Python's scikit-learns implementation of decision-tree classification is based upon the CART algorithm [425]. The CART algorithm is categorised by the fact that it constructs binary trees. A detailed overview of CART can be found in Ref. [450]. Other popular tree-based classifiers include C4.5 [451] and ID3 [425].

Decision-tree models can be overcomplicated, which can lead to overfitting. However, there are tactics to limit the tree complexity, e.g. pruning and setting the maximum depth of the tree [425]. Furthermore, decision trees can be unstable and create biased learners when there are dominant classes [425]. Ensemble methods such as random forest classifiers are a solution to this.

Random forest

Random forest classifiers are discriminative, consisting of a range of decision trees trained on a random subset of input features. The random forest then uses the votes of each decision tree to determine the class of each sample. This process is referred to as bagging [452]. As a result of

bagging, random forest classifiers are less noisy and affected by outliers; hence there is a lower risk of overfitting than with decision tree classifiers [453, 454, 452].

Unsupervised approaches to classification

Unsupervised ML algorithms identify patterns in unlabelled data. Clustering, the process of finding groups of data (clusters) in a dataset, is an example of an unsupervised ML-approach first introduced in Chapter 3. By setting the number of clusters to two and using an additional algorithm to assign a label to each cluster, clustering can be applied to binary classification.

There are a range of different approaches to clustering (e.g. DBSCAN [455], spectral clustering [456], OPTICS [457]). However, for a clustering algorithm to be applied to binary classification, it must have the number of output clusters specified as an input parameter and be able to predict cluster assignments for unseen data. K-means clustering, a popular clustering approach, is the only clustering algorithm with a Python implementation able to do both. See Section 4.1.6, Eq. (4.13), for an explanation of the K-means clustering algorithm.

8.1.3 Model validation with small datasets

An important step in the development of an ML model is model validation. This is the process of quantifying the ability of an ML model to produce predictions on an unseen test set [458]. Model validation can be used to identify problems such as overfitting as well as to optimise model performance [411].

A common approach to model validation in ML is the 'train/test split' approach. This evaluates the performance of a model on a portion of the data kept separate from the training data. The trained model can then be tested on an independent test dataset. However, this approach requires a substantial amount of data and is not suitable for data-limited environments. Cross-validation is a solution for model validation in data-limited environments [459]. K-fold cross-validation is a common cross-validation approach. This involves splitting a training dataset into K subsets. The model is trained on $K-1$ of these subsets, and its performance is then evaluated on the remaining subset. This process is repeated K times. In each iteration, the model is tested on a different subset of the data. All the data is used for validation. The total of

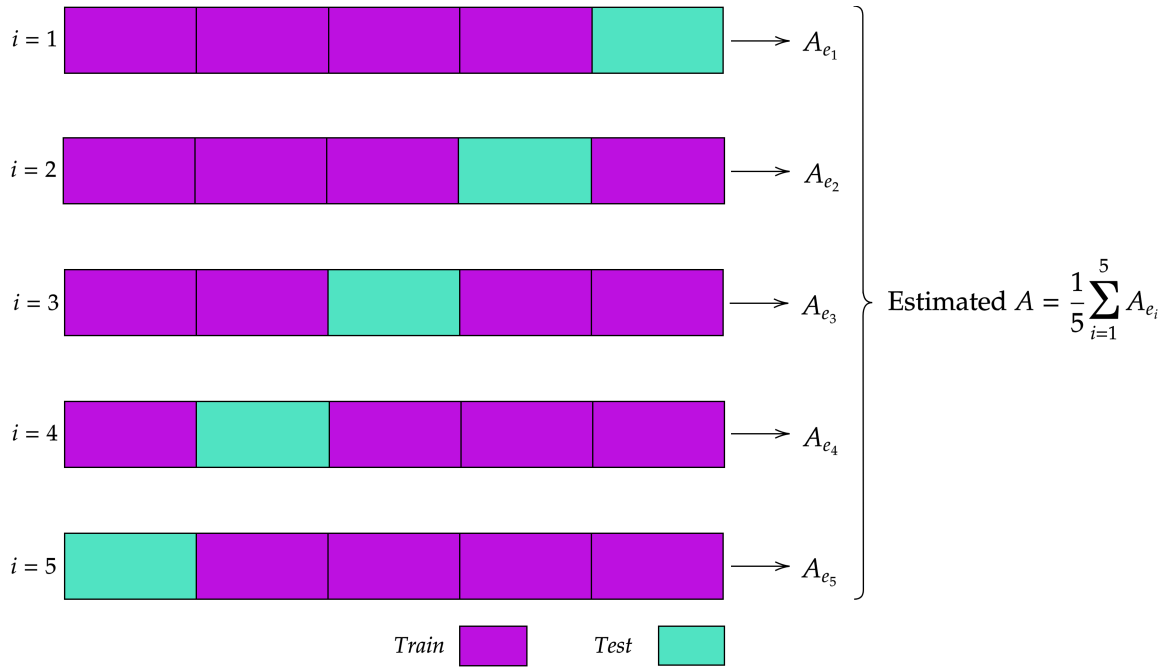


Figure 8.1: 5-fold cross-validation. A training dataset is partitioned into 5 subsections. Then, the model is trained on 4/5 of the training dataset and tested on the remaining 1/5 of the training dataset to obtain an estimate of model performance. This process is repeated 5 times: testing and training on different combinations of the subsets of the training data. Finally, an estimated accuracy (Estimated A) and a generalisation error is generated by taking the mean and standard deviation of the 5 estimated accuracies A_{e_i}.

K scores of the model's performances are averaged to estimate the overall model performance. Testing the model on different sections of the training data in each iteration or 'fold' indicates the model's generalisation performance [460]. Basic five-fold cross-validation is illustrated in Figure 8.1. Repeated K-fold cross-validation performs the splitting process N times to generate $N \times K$ estimates of model performance.

Model optimisation is part of model validation. ML algorithms have hyperparameters (tuning parameters) that need to be tuned to a specific dataset [461]. Selection of model hyperparameters is typically performed through optimisation algorithms that evaluate the model's performance with a specific set of hyperparameters using cross-validation such as grid search [462]. However, since hyperparameters can significantly impact algorithm performance, performing hyperparameter tuning separate from cross-validation can lead to a biased estimate of algorithm performance [461].

Nested cross-validation estimates the model's generalisation error and its hyperparameter search. It consists of an inner loop responsible for hyperparameter optimisation and an outer loop for

estimating the generalisation error [461]. When K-fold cross-validation is used for the inner and outer loops, the training data for the outer layer of the nested cross-validation is partitioned into K folds. Each fold uses a fraction $(K\text{-}1)/K$ of the original training data to train the model and the remaining $1/K$ to estimate the cross-validation error on that loop. This is iterated for the inner loop by using $1/K$ of the remaining training data from the outer loop ($(K\text{-}1)/K^2$ of the original training data) to compute the cross-validation error for the inner loop. The inner loop identifies the optimal hyperparameters for the fraction $(K-1)^2/K^2$ of data it is trained on. Nested cross-validation is recommended for small datasets as studies have shown it produces unbiased estimates of model performance [411, 461].

8.1.4 Evaluation metrics

In this section, standard metrics for evaluating the performance of ML algorithms will be introduced. These metrics are based on the following:

- True positives (T_P) are the number of items correctly labelled as the positive class (High-TRL).
- True negatives (T_N) are the number of items correctly labelled as the negative class (Low-TRL).
- False positives (F_P) are the number of items incorrectly labelled as the positive class (High-TRL).
- False negatives (F_N) are the number of items incorrectly labelled as the negative class (Low-TRL).

Using these interpretations of T_P, T_N, F_P and F_P, the metrics recall R (Eq. (3.4)), precision P (Eq. (3.5)) and F (Eq. (3.6)) introduced in Section 3.4 can hence be adapted to binary classification. F-measure in this context will be referred to as F_1.

R is the ratio of the number of positive predictions to the number of true positive items. R is a value between 0 and 1, the closer to 1, the better the model's ability to predict the positive class [463]. P is the ratio of the number of positive class predictions to the number of items belonging to the positive class [463]. An intuitive interpretation of precision indicates how

much a model can be trusted when an item is labelled as positive [463]. Like recall, precision and F_1 are values between 0 and 1, the closer to 1, the better the model.

Accuracy

Accuracy, A, is defined as the probability that a model's prediction is correct, i.e.

$$A = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}. \tag{8.3}$$

As with the metrics mentioned earlier, accuracy is between 0 and 1, where a higher value indicates better performance.

Area under the receiver operating characteristic curve (AUC ROC)

Area under the receiver operating characteristic curve, AUC ROC is a standard metric for evaluating classifiers. A receiver operating characteristic curve (ROC curve) is a probabilistic graphical representation of the true positive rate (the number of true positives divided by the total number of positives) vs the false positive rate (number of negatives incorrectly identified as positives divided by the total number of negatives) [464]. The AUC ROC gives a value between 0 and 1 to the ability of the classifier to distinguish between two classes. The closer to 1, the better the ability of the classifier to distinguish between classes. One advantage of this metric is it can be used to evaluate classifiers with a class imbalance [465].

Silhouette coefficient

The silhouette coefficient is a measure of the effectiveness of a clustering algorithm. It measures the cohesiveness of items in a cluster, and the separation between clusters [466]. The silhouette coefficient is a value between 1 and -1. A score of 1 indicates well-defined clusters.

The silhouette coefficient is defined as

$$S_c = \sum_{i=1}^{N} \frac{b(i) - a(i)}{\max(a(i), b(i))}, \tag{8.4}$$

where $a(i)$ is the mean distance between sample i and all other samples in the same cluster

(C_I), i.e.

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j), \tag{8.5}$$

where $d(i, j)$ is the distance between data points i and j in cluster C_I. One interpretation of the value of $a(i)$ is how well a sample is assigned to its cluster.

The minimum mean distance of sample i to all other samples contained in any other cluster of which i is not a member $(b(i))$ is defined as

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j). \tag{8.6}$$

The metric $b(i)$ can be interpreted as a measure of cluster separation [466].

8.2 Methodology

In this section, the methodology used in this study will be described. Section 8.2.1 will detail the data used in this study and the methods used to preprocess the data. In Sections 8.2.2 - 8.2.6, details of the algorithms used and how they are implemented will be discussed.

8.2.1 Dataset production and preprocessing

The dataset used in this study consists of 148 NTREM 2018 papers hand labelled by an expert with over 30 years of experience in the domain of energetic materials. The expert determined whether each paper presented a technology that was either 'High-TRL' or 'Low-TRL' according to the MoD's definition of TRL levels (see Table 8.1). Papers associated with TRLs 1-5 were defined as being Low-TRL, and those of TRLs 6-9 were defined as High-TRL. This information was elicited by a questionnaire in the form of a two-column spreadsheet consisting of document titles and a blank column for the expert to assign the TRL. Of the documents labelled by the expert, 81 (54.7%) were labelled as Low-TRL, and 67 (45.3%) were labelled as High-TRL. A separate dataset of 1120 unlabelled NTREM papers were used for unsupervised algorithms.

This set of conference proceedings was chosen because it comes from a peer-reviewed journal in the field of energetic materials. The proceedings contain a broad spectrum of topics and

research areas within energetic materials, making it ideal for classification via a subjective concept such as TRL.

A keyword-matching algorithm was used as a baseline measure of performance. This required an input of two lists of keywords associated with High- and Low-TRLs. The keywords generated by the expert are shown in Table 8.2. Each list contains 34 keywords.

TRL	Expert keywords
Low	'Synthesis', 'Milligram', 'Purity', 'Impurity', 'Spectra', 'Laboratory-scale', 'Reflux', 'Nitration', 'Reaction', 'Calorimetry', 'Test-tube', 'Bunsen-burner', 'Solvent-extraction', 'Reagent', 'reactant', 'Side-product', 'by-product', 'Crystallisation', 'Filtration', 'Chromatography', 'Nuclear-magnetic-resonance', '%yield', 'Research', 'science', 'Spectroscopy', 'laboratory', 'lab', 'small-scale', 'future', 'next-generation', 'patent', 'invention', 'laboratory-based', 'mg'
High	'Output', 'Additives', 'Factory', 'Processing', 'Field-trial', 'Range-trial', 'Diagnostics', 'Adhesion', 'Debris', 'Fragment', 'Application', 'Mixing', 'Mixer', 'Enhanced', 'Blast', 'Detonation', 'velocity-of-detonation', 'Ageing', 'Storage', 'Transport', 'Packaging', 'Development', 'technology', 'safety', 'compatibility', 'vulnerability', 'utility', 'large-scale', 'upscaling', 'manufacture', 'demonstrator', 'preproduction', 'monitoring', 'through-life', 'range-testing', 'trial'

Table 8.2: Lists of keywords associated with High-TRL and Low-TRL generated by the expert.

For the NTREM proceedings to be suitable for NLP, the articles' text needs to be extracted from PDF documents of the articles. The preprocessing pipeline outlined in Chapter 7 was used to extract text from the proceedings. Task-specific processing then needs to be performed. As discussed in Chapter 7, different feature-representation methods require different preprocessing methods. Therefore, the pipeline outputs preprocessed text in two forms: (1) detokenised text and (2) processed text with stop words removed. This study examines two different feature representations, each requiring its own downstream preprocessing method (method A and method B).

Method A was used for supervised classification using TF-IDF and the keyword-matching algorithm. This method takes an input of processed text with stop words removed from the preprocessing pipeline. The text is further normalised by performing lowercasing. Non-chemical

names were stemmed using the Porter stemmer algorithm [467][2]. These steps significantly reduced the dimensions of the vocabulary of the corpus thus reducing the dimensions of the numerical representations of the documents.

Method B was developed for finetuning BERT and supervised classification using S-BERT. BERT-based models have a maximum number of input tokens of 512 [129]. Therefore, the articles were truncated so they were suitable to be input into the keyword-matching algorithm.

Following the preprocessing, the datasets were split into a training set of 118 documents and an independent test set. The independent test set consisted of 30 instances: 16 documents of Low-TRL and 14 of High-TRL.

8.2.2 Keyword-matching algorithm

The keyword-matching algorithm followed preprocessing method A to preprocess the text. The keyword lists given in Table 8.2 were also stemmed. The algorithm totalled the frequency of High (C_H) and Low (C_L) TRL keywords in each document. The number of matches for High (M_H) and Low (M_L) TRL words was also calculated to give the coverage of the keywords. Eq. (8.7) denotes how the algorithm determines each journal article's TRL

$$\text{Predicted TRL} = \begin{cases} \text{High}, & \text{if } C_H > C_L \text{ or if } C_H = C_L \text{ and } M_H \geq M_L \\ \text{Low}, & \text{if } C_L < C_H \text{ or if } C_H = C_L \text{ and } M_H < M_L. \end{cases} \tag{8.7}$$

The keyword-matching algorithm assumes a higher frequency of associated TRL words corresponds to a higher likelihood the document is of that TRL. In instances where the frequency of High- and Low-TRL words are identical, the coverage of the keywords in the documents is considered. Since this is an unsupervised statistical method, requiring no model training, the algorithm is tested on both the training set and the independent evaluation set, a total of 148 documents.

[2]In Chapter 4, a small comparative study of the performance of stemming algorithms in the energetic-materials domain concluded the Porter stemmer performed the best.

8.2.3 Finetuning Sci-BERT

Preprocessing method B was used to generate the input text for finetuning Sci-BERT. When conducting transfer learning, it is advantageous to utilise a pretrained model that has been trained on text as close as possible to the domain of interest. This will mean the model will be better at capturing the semantic nuances of the field. There is currently no open-source BERT language model trained on text in the energetic-materials domain. Sci-BERT is a pretrained model trained on a large corpus of scientific publications. It was developed to address the lack of large-scale labelled scientific data and is found to improve performance on a range of different downstream tasks [315]. Sci-BERT was determined to be the most appropriate of the pretrained BERT models available and was therefore finetuned on the training dataset of 118 documents. This is a very small dataset for finetuning and will likely yield unstable results. Following recommendations found in the literature for small datasets, a small batch size of 8 and low learning rate of 10^{-5} was chosen [129, 421]. Class weights were adjusted to enable mistakes in under-represented classes to be penalised to a higher degree. The performance of the model was evaluated using the independent evaluation test set.

8.2.4 TF-IDF and S-BERT feature representation and feature selection for clustering and supervised classification

The TF-IDF procedure explained in Section 8.1.1 was applied to the preprocessed text using preprocessing method A. This produced 17419-dimensional TF-IDF vectors.

The sentence-transformers framework in Python for S-BERT was used on text preprocessed using Method B to generate embeddings of the first 512 tokens of the documents [319]. S-BERT has various pretrained models available [422]. In this study, the 'all-distilroberta-v1' model was chosen as it is a general-purpose model [468]. This model was developed by finetuning the pretrained 'distil-roberta-base' model (a variant of the BERT model) on 1 billion sentence pairs [469]. Typically, S-Bert is used for comparisons of semantic similarities between sentences. However, there are examples in the literature of S-BERT used to generate sentence embeddings for classification models [470]. This technique produced embeddings of 768 dimensions.

For both TF-IDF and S-BERT, the vector representations of text were produced before splitting

the data into training and test sets.

Both feature-representation methods generated high-dimensional embeddings. Training small sample sizes using high-dimensional embeddings can result in overfitting. Given this, an important step in producing a classifier is to perform feature selection (selecting a subset of features to reduce the dimensions of the data inputted into the classifier) and/or feature reduction (transforming data to lower dimensions) [471]. There are a range of different methods available for feature selection/reduction, Ref. [472] gives a good survey. In this initial study, based on some exploratory analysis, the KBest feature-selection approach was used to reduce the dimensions of the embeddings produced. For each feature-representation method, the d_n most relevant features of the embeddings were selected using the KBest function in scikit-learn [425]. The KBest function ranks every feature by their mutual information value and returns the d_n most relevant features, which generates vectors of d_n dimensions [425]. The number of dimensions d_n was selected as 50[3].

After performing feature-representation-specific processing and feature selection, the data is inputted into the K-means clustering algorithm (Section 8.2.5). For supervised classification approaches, model validation and hyperparameter optimisation are additional steps requiring further discussion (Section 8.2.6).

8.2.5 K-means clustering

K-means clustering (Eq. 4.13) was applied to binary classification by setting the number of clusters to two and using an additional algorithm to assign a label to each cluster. The algorithm used to assign labels to the clusters was voting-based. First, for every document in each cluster, the keyword-matching algorithm was used to assign each document a TRL. Secondly, the ratio of High-TRL documents to Low-TRL documents was calculated for each cluster. Then, the cluster with the highest ratio was labelled High-TRL, and the other cluster was labelled Low-TRL. Due to the algorithm's unsupervised nature, the clustering algorithm was trained on 1120 unlabelled documents.

[3]This value was selected as it is the smallest number of dimensions commonly observed in the literature for text representation [122].

8.2.6 Model validation and hyperparameter optimisation for supervised classifiers

For supervised classification approaches, model validation and hyperparameter optimisation are additional steps requiring further discussion.

For model validation, nested cross-validation was performed[4]. The inner and outer loops consisted of 5-fold repeated cross-validation. This number of folds is a typical value of K commonly found in the literature [473]. In addition, hyperparameter optimisation was performed in the inner loop of the nested cross-validation.

The supervised algorithms have a range of hyperparameters. In the nested-cross-validation process, some hyperparameter optimisation is performed. Since this study aims to compare the performance of different algorithms, the size of the hyperparameter grid (number of hyperparameters optimised $\times$ their possible input values) is kept constant at 9 dimensions.

For the SVM classifier, a linear-kernel function was selected as using simple models reduces the risk of overfitting. Linear SVM has one hyperparameter requiring optimisation. The regularisation parameter C determines the strength of the regularisation and is the only hyperparameter that needs to be optimised. The regularisation parameter determines the algorithm's tolerance when identifying the decision boundary, i.e. it determines how much the algorithm wants to avoid classifying each training example [474]. A significant value of C could lead to overfitting. Therefore, when performing hyperparameter optimisation, the parameter grid over which the optimisation is performed was restricted to smaller values of C, namely [10, 1, 0.5, 1.5, 0.1, 0.01, 0.001, 0.0001, 5]. The class weights parameter determines the weighting that different classes are given. The default value is 1. A value of 1 means mistakes in the classification of the two classes are treated as being of equal importance. Adjusting the class weights parameter can enable mistakes in under-represented classes to be penalised to a higher degree. This is useful for datasets with imbalanced classes. As mentioned in Section 8.2.1, the TRL dataset is unbalanced. Therefore, the class-weights parameter was set to 'balanced'. This sets the class weights to values inversely proportional to the class frequencies [425].

The logistic regression classifier was used with $L2$ regularisation to reduce the risk of overfitting. As with the SVM classifier, the class-weight parameter was set to 'balanced' to account for the

[4]See Section 8.1.3 for an overview of nested cross-validation.

slight class imbalance. The tolerance was set to the default value [425]. As with the SVM classifier, the parameter C controls the strength of the regularisation and the smaller the value, the higher the regularisation. The logistic regression model was therefore optimised for parameter C over the following grid: [10, 1, 0.5, 1.5, 0.1, 0.01, 0.001, 0.0001, 5].

The number of neighbours, k_n, in a k-NN classifier has a significant influence on results. Smaller values of k_n are associated with overfitting. For the k-NN classifier, hyperparameter optimisation is performed over the following values of k_n, [4, 5, 6,7, 8, 9, 10, 11, 12]. The distance metric used in this classifier was selected as the Manhattan distance as it tends to perform better in high-dimensional spaces than other metrics such as Euclidean distance [475].

Decision tree and random forest classifiers have several different parameters. In this study, the number of minimum samples required to split a node will be optimised. When this value is high, it will result in trees of a smaller depth, avoiding overfitting. The following grid: [5, 6, 7, 8, 9, 10, 11, 12], was used for hyperparameter optimisation of minimum samples per split for both the decision tree and random forest classifiers. Post-pruning of the decision tree is not used in the scikit-learn implementation of the algorithm [425]. However, this can reduce the risk of overfitting [476].

Performing nested cross-validation will generate a series of estimated accuracies (A_e), the mean and standard deviation of these values can be taken to obtain the Estimated A and generalisation error.

After model validation, hyperparameter optimisation is performed using grid-search with 5-fold repeated cross-validation. Using the optimised hyperparameters, the classifiers are trained on the whole training set. The performance of the model is evaluated on the independent test set.

8.3 Results and discussion

The results generated using the keyword-matching algorithm, finetuning Sci-BERT, supervised algorithms, and K-means clustering with TF-IDF and S-BERT for feature representation are summarised in Table 8.3. The metrics F_1, R, P, ROC AUC and A (see Section 8.1.4) are used to evaluate the performance of the algorithms. Estimated A of the linear SVM, decision tree, k-NN, NB, logistic regression and random forest classifiers are also displayed. These values are

calculated by taking the mean and standard deviation of the accuracy estimates A_e outputted by the nested cross-validation.

Algorithm	F_1	P	R	ROC AUC	A	Estimated A
Keyword matching	0.74	0.65	0.85	0.73	0.73	
Finetuning Sci-BERT	0.77	0.66	0.92	0.77	0.77	
K-means S-BERT	0.62	0.62	0.62	0.62	0.62	
K-means TF-IDF	0.82	0.72	0.77	0.79	0.78	
Linear SVM TF-IDF	0.81	0.72	0.93	0.81	0.80	0.78 ±0.08
Linear SVM S-BERT	0.90	0.87	0.93	0.90	0.90	0.80±0.10
Decision tree TF-IDF	0.66	0.71	0.67	0.68	0.67	0.67 ± 0.06
Decision tree S-BERT	0.77	0.77	0.77	0.76	0.77	0.73 ± 0.10
k-NN TF-IDF	0.77	0.77	0.77	0.77	0.77	0.70 ± 0.06
k-NN S-BERT	0.87	0.87	0.87	0.87	0.87	0.83 ± 0.08
NB TF-IDF	0.65	0.74	0.67	0.68	0.77	0.70 ± 0.09
NB S-BERT	0.90	0.92	0.90	0.91	0.90	0.82 ± 0.08
Logistic regression TF-IDF	0.79	0.85	0.80	0.79	0.80	0.71 ± 0.08
Logistic regression S-BERT	0.93	0.94	0.93	0.94	0.93	0.80 ± 0.08
Random forest TF-IDF	0.90	0.90	0.90	0.90	0.90	0.81 ± 0.07
Random forest S-BERT	0.93	0.94	0.93	0.94	0.93	0.81 ± 0.08

Table 8.3: Performance of the algorithms evaluated using the metrics F_1, R, P, ROC AUC and A. Estimated values of A are displayed for the linear SVM, decision tree, k-NN, NB, logistic regression and random forest classifiers, as these methods used nested cross-validation.

From the results shown in Table 8.3, it is evident that the preprocessing pipeline designed in Chapter 7 has effectively facilitated document classification.

The keyword-matching algorithm (Section 8.2.2) obtained a R of 0.85, meaning 85% of all High-TRL documents were identified correctly, implying a small proportion of Low-TRL documents were incorrectly labelled. The P of 0.65 indicates when a document is predicted to be of High-TRL, a significant proportion of these predictions (35%) are incorrect. Ideally, a classification algorithm will have high R and high P, returning most of all labelled High-TRL documents accurately. The difference in P and R may have resulted from the class imbalance since there is a higher number of Low-TRL documents to incorrectly label than High-TRL documents. The A of the keyword-matching algorithm (how many documents were classified correctly) was 0.73. Previous studies have shown a class imbalance can impact the value and meaning of A, R and P [477]. Metrics such as ROC AUC and F_1 are less influenced by class imbalance. According to Refs. [478, 479], an ROC AUC value of 0.73 is considered 'good' or 'fair' [480].

The keyword-matching algorithm could be of use when the goal is to return most of the High-

TRL documents regardless of whether some Low-TRL documents are misclassified as High-TRL. However, the overall ability of the algorithm to distinguish between documents of High- and Low-TRL is suboptimal. There are a few possible reasons for this. Firstly, whilst the frequency of keywords in a document might indicate the TRL of a document, using keywords alone is not enough information to decide on the TRL. The context in which the keywords appear is not considered. In specific contexts, a TRL keyword may not be a good indicator of the associated TRL. For instance, 'small-scale' is a Low-TRL keyword; however, in specific contexts, it can indicate High-TRL, e.g. in the sentence 'Analysis is not conducted on a small-scale'. Secondly, the algorithm assumes the number of keywords used in the list is sufficient, which may not be the case. As a result, the algorithm could be improved by expanding the list of keywords and phrases. This could be achieved by consulting a larger group of experts or performing unsupervised training of a word-embedding algorithm, such as Word2Vec, on an extensive collection of energetic-materials-based papers. The trained word-embedding model could then be used to identify semantically similar keywords which could be added to the existing keyword list.

Similar to the keyword-matching algorithm, Sci-BERT (Section 8.2.3) was found to have a low P (0.66) and high R (0.92). These values indicate that most of the High-TRL documents are identified, but many documents are incorrectly identified as High-TRL. This trend of higher R and lower P is also observed in the results generated for supervised classifiers using both TF-IDF and S-BERT. Again, this may be due to the class imbalance. It is also noted that with a smaller number of High-TRL documents to learn from, the ability for the classifier to identify a High-TRL document correctly will be poorer than its ability to identify a Low-TRL document correctly. Whilst the effect of the class imbalance was attempted to be mitigated through adjusting the class weights parameter, this may not be sufficient as the impact of class imbalance at a small sample size is significant.

Overall, finetuning Sci-BERT achieved A and ROC AUC values of 0.77. This is a slight improvement on the keyword-matching algorithm and in line with results achieved on similar sample sizes [481]. Most likely, the suboptimal result is due to the lack of training data. Fine-tuning BERT on small datasets is unstable, and this instability was unmeasured. It is noted that this approach had several limitations. Firstly, only the first 512 tokens of the text were used. A suggestion for future work would be to utilise other language models, such as Long-

former, designed for longer text as an input [482]. It could be the case that the first 512 tokens of the text are not sufficient for predicting the TRL of the model. Universal Language Model finetuning (ULMFiT) is an effective transfer learning method for NLP that has achieved promising results with as few as 100 samples [483]. Future work should utilise this framework.

The method described in Sections 8.2.4 and 8.2.6 was used to obtain measures of the supervised classifiers' performances. First, the distribution of A_e will be examined. Figure 8.2 displays the distribution of A_e for each supervised classifier for both TF-IDF and S-BERT feature-representation methods.

Figure 8.2 shows a significant variation in the predicted A_e of the classifiers depending on what portion of the data it was tested/trained on. For the decision tree, k-NN, linear SVM and NB classifiers, the distribution of A_e is higher when using S-BERT for feature representation. This implies the classifiers are more stable, utilising TF-IDF as a feature-representation method.

However, as mentioned in Section 8.2.4, TF-IDF and S-BERT vector representations of text were produced before splitting the dataset into training and test sets. Whilst generating S-BERT representations before splitting the data into training and test sets is no different from generating the representations after the split, this is not true for TF-IDF representations. This is because TF-IDF values and the generated document vectors depend on all the documents in a corpus. Therefore, if feature representation with TF-IDF is performed before splitting the data into training and test sets, and is not performed as part of the nested cross-validation process, there will be substantial data leakage which will introduce bias. This may lead to over or underestimates in classifier performance. Furthermore, due to the variances in fitting TF-IDF not being fully captured as part of the validation process, the error in the Estimated A for TF-IDF feature-representation methods in Table 8.3 and the distributions in Figure 8.2 are likely underestimating the actual variance.

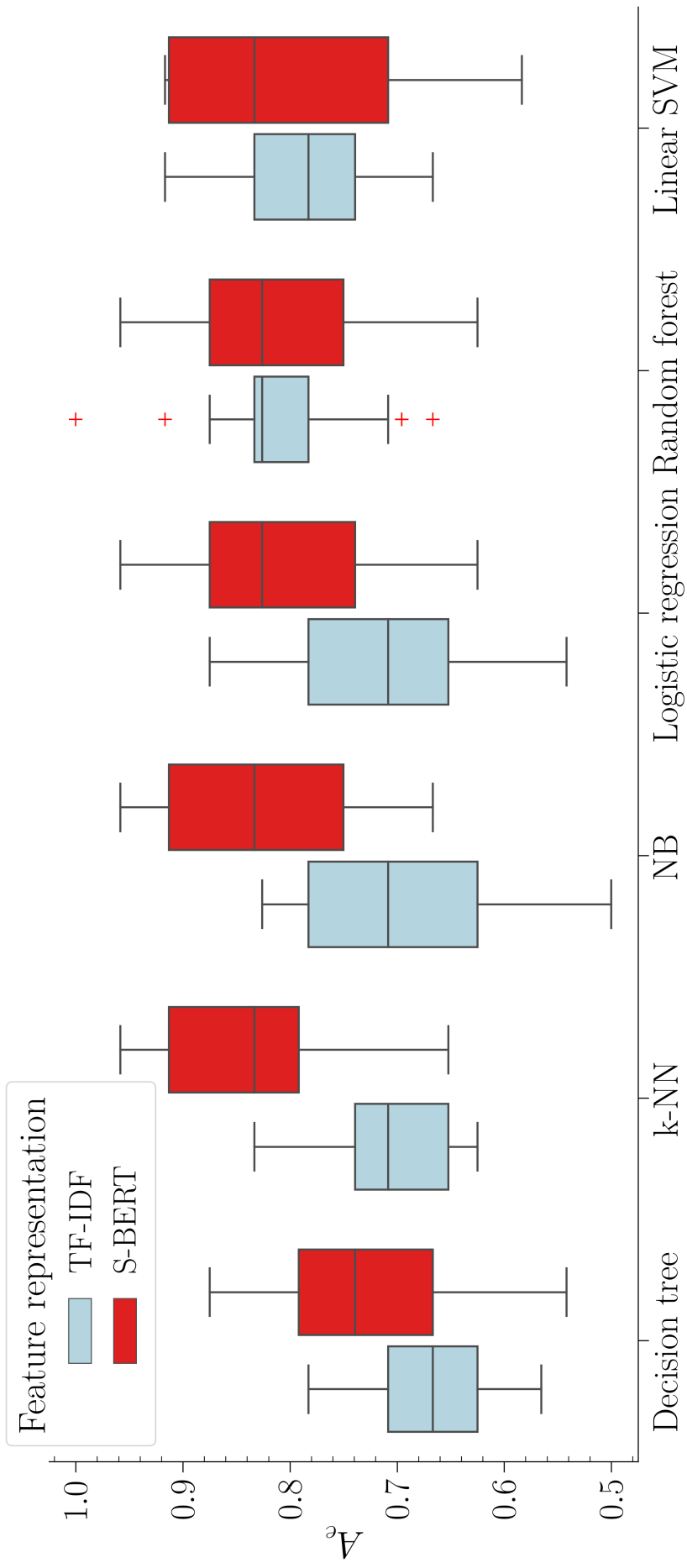
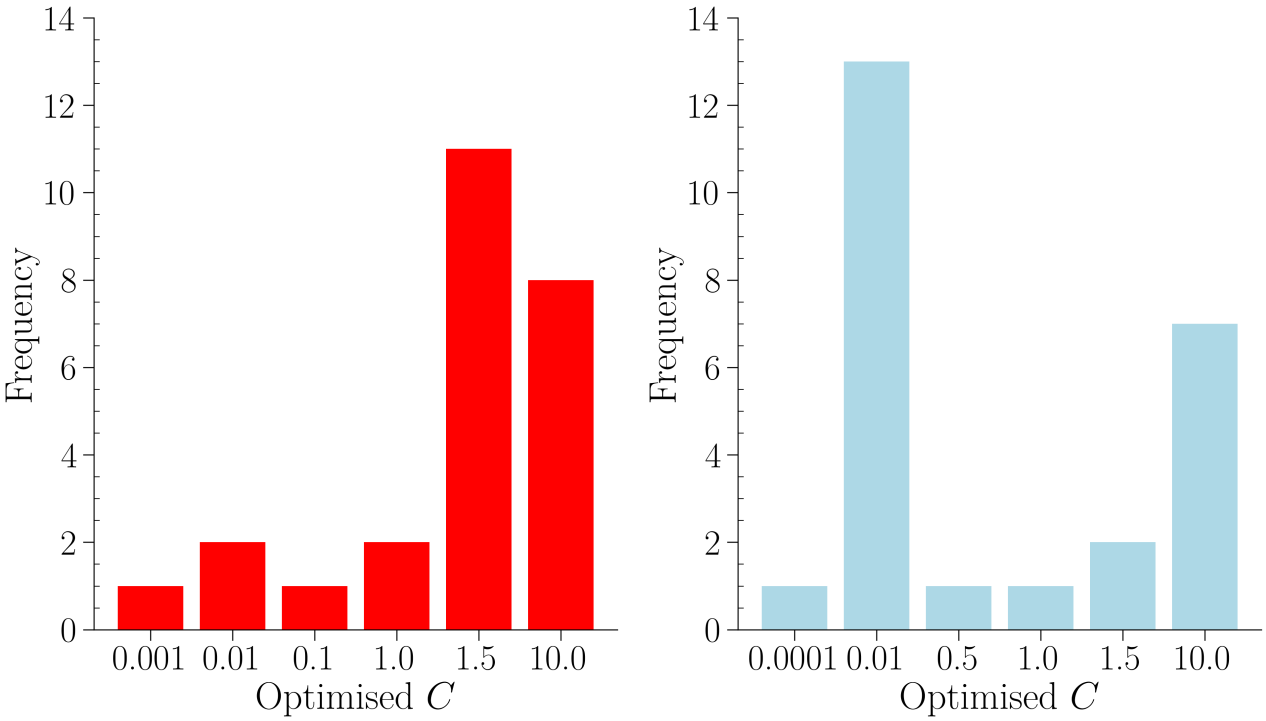


Figure 8.2: Distribution of A_e of decision tree, K-NN, Gaussian NB, logistic regression, random forest and linear SVM classifiers with TF-IDF and S-BERT feature-representation methods. A_e are obtained using nested cross-validation.

To understand the significant variance of A_e observed in Figure 8.2, the hypothesis that variation in the selection of hyperparameters in the inner loop of cross-validation as the predominant cause of model variance is investigated.

The example of linear SVM is considered. The frequency of hyperparameter C's selection on the inner loop of nested cross-validation is displayed in Figure 8.3.



(a) S-BERT (b) TF-IDF

Figure 8.3: Frequency of selection of the hyperparameter C as a result of hyperparameter optimisation in the inner loop of nested cross-validation over the hyperparameter grid [0, 1, 0.5, 1.5, 0.1, 0.01, 0.001, 0.0001, 5] for a linear SVM classifier.

Figure 8.3 shows a significant variation in the hyperparameters selected as part of the inner loop of the nested cross-validation for both S-BERT and TF-IDF feature-representation methods. Despite observing more variation in the distribution of A_e for TF-IDF than S-BERT in Figure 8.2, Figure 8.3 shows very similar levels of variation in hyperparameter selection. This disproves the hypothesis that variation in the selection of hyperparameters in the inner loop of cross-validation is the predominant cause of the model variance. However, as mentioned previously, due to TF-IDF fitting not being included as part of the nested cross-validation, there is an unaccounted for cause of error.

Training a classifier on a small dataset such as the one in this study will likely result in significant

variances. Here, 5-fold cross-validation was performed on a training set of 118 documents. This means the model performance was validated for 23-24 documents for each fold of the cross-validation algorithm. The inner fold used approximately 75 documents for hyperparameter optimisation and around 18 documents for testing performance. This is a tiny test size. Whilst the ability of the data to generalise to new data is being measured, at such a small sample size, the presence of outliers could affect results considerably. This explains the significant variance observed in Figure 8.2 and the instability in the selected hyperparameters discussed previously.

The results in Table 8.3 are used to identify overfitting for the supervised classification methods by comparing the Estimated A with the model's accuracy on the independent test set (A). If A is within one standard deviation of the Estimated A it is assumed the classifier has not overfit. Logistic regression and random forest classifiers were found to overfit. It is unsurprising that overfitting is observed particularly for complex algorithms such as random forest at such a small training size.

Overall, the supervised classifiers performed better using the transfer-learning technique S-BERT for feature representation. This supports the hypothesis that transfer learning can improve performance for smaller datasets. However, it is noted that when using TF-IDF, the vectors are fit before splitting the data. This could have resulted in data leakage, which would have led to biased results. Therefore, future implementations of this method should perform the TF-IDF fitting after splitting data into training and test sets and as part of the cross-validation process. Furthermore, feature selection is not included as part of the nested cross-validation process. In this study, feature selection was performed arbitrarily. Therefore, it is unclear what effect this has on algorithm performance. There are various feature-selection methods available in Python, so it would be interesting to see how these other methods impact the results for algorithm performance. Ref. [461] demonstrated that using feature selection in nested cross-validation can improve performance. Further work should incorporate feature selection as part of the nested cross-validation.

The best-performing supervised classifier can be identified from Table 8.3. When considering all metrics, linear SVM and NB with S-BERT performed the best with an A of 0.90. NB slightly outperforms linear SVM as it has a lower error in its Estimated A and a higher ROC AUC score of 0.91. Whilst these results appear positive, in Figure 8.2 these algorithms were observed to have a large distribution of A_e, which indicates a lack of model stability.

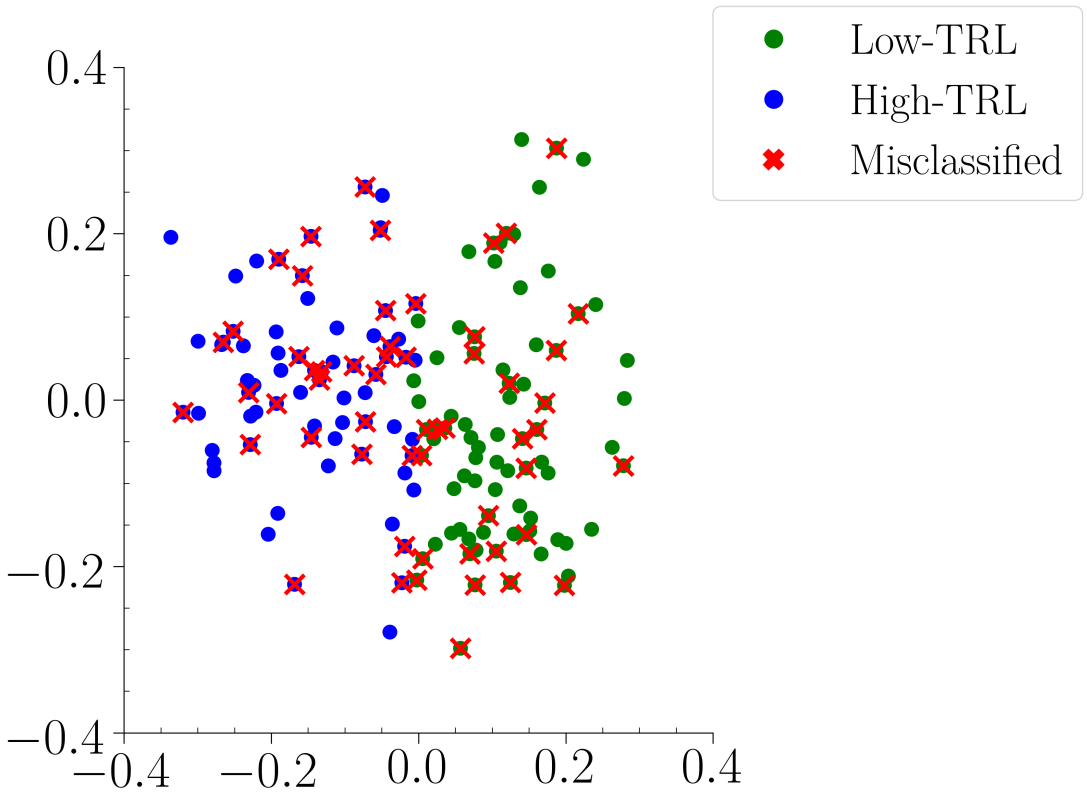
The results of the K-means clustering approach to classification described in Section 8.2.5 will now be discussed. In order to visualise algorithm performance, the vector representations of the documents in the test set were transformed to 2D coordinates using the feature-reduction technique Principal Component Analysis (PCA)[5]. The generated 2D coordinates for S-BERT and TF-IDF feature-representation methods were then plotted in Figures 8.4 (a) and (b) respectively.

Figures 8.4 (a) and (b) show the clusters are not clearly defined for both vector representations with Figure 8.4 (b) showing overlapping clusters. Values of S_c were calculated, and values of 0.14 and 0.02 for TF-IDF and S-BERT were obtained. These low values are further evidence of the poor quality clusters.

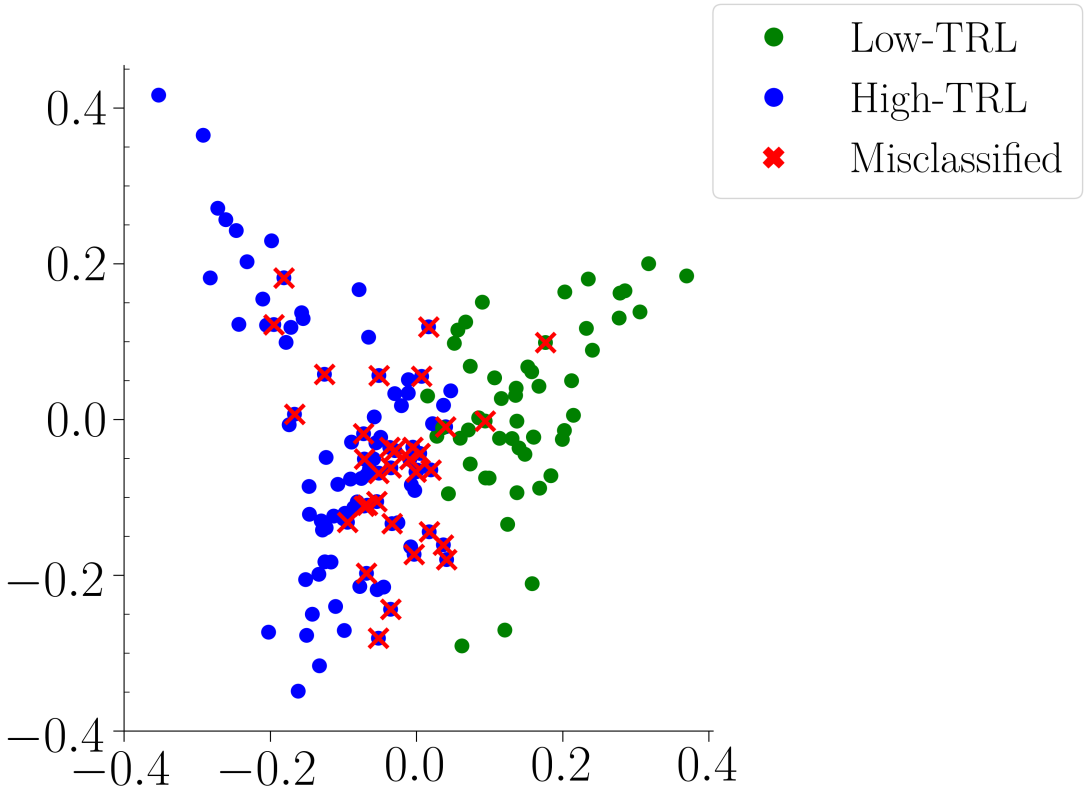
The suboptimal performance illustrated in Figures 8.4 is further observed in Table 8.3. The algorithm's performance on the test set shows feature representation with TF-IDF outperforms feature representation with S-BERT across all metrics. This is in line with the values of S_c, which also indicated TF-IDF as a feature-representation method generated better-defined clusters. It is noted this may be because the TF-IDF vectors are fit before splitting the data. As previously explained, this could have resulted in data leakage, leading to improved performance. As previously noted for supervised techniques, the TF-IDF fitting should be done after splitting the data into training and test sets.

Generally speaking, the supervised classification algorithms outperformed K-means clustering with TF-IDF. K-means clustering with TF-IDF outperforms the baseline of the keyword-matching algorithm and finetuning Sci-BERT, with $A = 0.78$. In contrast, K-means clustering with S-BERT outperforms all other examined approaches, obtaining $A = 0.62$ (which is low).

[5]See Ref. [484] for a detailed overview of PCA.



(a) Feature representation with S-BERT.



(b) Feature representation with TF-IDF.

Figure 8.4: Visualisation of K-means clustering algorithm performance. Each document in the test set is transformed to 2D coordinates using PCA. Documents labelled as High-TRL are shown in blue and documents labelled as Low-TRL as shown in green. Misclassified documents have a superimposed red cross.

The results appear promising, with most approaches exceeding the baseline value of $A = 0.73$. K-means clustering with S-BERT was the worst-performing algorithm of those examined. The best-performing algorithm that did not overfit was Gaussian NB. This algorithm achieved an A of 0.90 and a F_1 score of 0.90, indicating good classifier performance for such a small dataset. Whilst these results indicate the algorithms perform relatively well. It is noted that with such a small sample size for evaluation, the results should be taken with a pinch of salt.

8.4 Conclusions of preliminary study

This chapter aimed to compare the use of supervised and unsupervised techniques for document classification on a small dataset of NTREM journal articles and evaluate the usefulness of the preprocessing pipeline developed in Chapter 7.

In Section 8.1, an overview of the relevant theoretical information for document classification was presented. An overview of feature representation with TF-IDF and transfer learning was given in Section 8.1.1, and Section 8.1.2 gave an overview of classification algorithms. Section 8.1.3 introduced the problem with model validation for small datasets. A review of the literature identified nested cross-validation as a potential solution. In Section 8.1.4, evaluation metrics were introduced.

In Section 8.2, the methodology used in this study was described. Section 8.2.1 detailed the process used to develop the dataset and described how the preprocessing methodology developed in Chapter 7 was used in this study. Downstream preprocessing methods were also described. Sections 8.2.2 and 8.2.3 described a simple keyword-matching algorithm used as a baseline measure of performance and the transfer-learning technique of finetuning Sci-BERT. Sections 8.2.5 - 8.2.6 described the methods used for K-means clustering and supervised classifiers with two different feature-representation methods, TF-IDF and S-BERT.

The results of this study were presented and discussed in Section 8.3. The results showed the preprocessing pipeline designed in Chapter 7 effectively facilitated document classification. The results of the keyword-matching algorithm yielded a suboptimal $A = 0.73$. Suggestions to improve algorithm performance are made, e.g. by expanding the list of keywords. The transfer-learning technique of finetuning Sci-BERT achieved $A = 0.77$, improving upon the performance

of the keyword-matching algorithm. However, the method used in this approach did not capture the instability of finetuning. The model was trained on a very small sample size therefore this model was likely overfitting.

A range of supervised classification algorithms was examined with two different feature-representation methods (TF-IDF and the transfer-learning technique S-BERT). Nested cross-validation was used for hyperparameter optimisation and to estimate classifier accuracy. The generated estimated accuracies were observed to be of high variance. It was hypothesised that variation in the selection of hyperparameters in the inner loop of cross-validation is the predominant cause of the model variance. Although this hypothesis was disproven, it was concluded that model variance occurred predominately due to the small sample size.

The estimated accuracies were compared with performance on an independent test set to identify overfitting. Logistic regression and random forest were the only two algorithms found to overfit. The majority of these supervised algorithms outperformed finetuning Sci-BERT and the keyword-matching algorithm on the test set. The best-performing algorithm overall was the supervised approach of Gaussian NB with S-BERT for feature representation. This algorithm achieved $A = 0.90$ and $F_1 = 0.90$, indicating good classifier performance for such a small dataset. For all supervised classifiers, the performance was higher when using S-BERT as a feature-representation method. This implies transfer learning can improve classifier performance at small sample sizes. However, it was noted this might be because the TF-IDF vectors were generated before splitting the data into training and test sets. This could have resulted in data leakage leading to biased performance. This also may explain the more significant variance in the estimated accuracies of classifiers using S-BERT as TF-IDF fitting not being included as part of the nested cross-validation is an unaccounted cause of error.

This same error in methodology may also explain the K-means clustering results. The algorithm's performance on the test set shows feature representation with TF-IDF outperforms the feature representation with S-BERT across all metrics. This may be because the fitting of TF-IDF vectors was performed before splitting the data. This could result in data leakage, which could have led to improved performance.

It is noted that the majority of examined approaches saw a high R and lower P, indicating that, whilst the algorithms could return the majority of High-TRL documents, a significant portion of these documents were incorrectly labelled. The dataset has a slight class imbalance that

could cause this as there are a more significant number of Low-TRL documents to misclassify and fewer High-TRL documents to learn from. Techniques to mitigate the effects of a class imbalance will be investigated in Chapter 9.

Overall, the initial results are promising. However, these promising results should be viewed with caution due to the small sample size for evaluation. Further work will involve collecting more data to see how the performance of these models changes with larger sample size. The desired outcome of this study was a series of recommendations to guide future studies. These recommendations are summarised as

- Feature selection or feature reduction should be included as part of the cross-validation process. This will allow for a more accurate generalisation error to be calculated and for optimisation of the number of dimensions of the input-document vectors.
- Generating TF-IDF vectors should be performed separately for the test and training datasets. It should also be performed as part of the inner loop cross-validation process for supervised techniques. This will reduce model bias and lead to a more realistic calculation of generalisation error.
- The dataset used in this study had a slight class imbalance. Techniques to mitigate the effects of a class imbalance should be investigated.

These recommendations will be explored further in Chapter 9[6].

[6] An additional recommendation would be to explore other finetuning approaches such as using ULMFit and long-former. However, time did not permit the investigation of these methods.

Chapter 9

Comparing experts' and NLP tools' classification of journal articles by TRL

This chapter is the second part of a two-chapter study addressing the limitations of using NLP tools in a small-data environment. In Chapter 8, a preliminary study was conducted comparing supervised and unsupervised techniques for document classification on a small dataset of NTREM journal articles. In this chapter, recommendations from Chapter 8 will be applied to refine the methodology presented in Chapter 8 in order to compare the ability of supervised and unsupervised classifiers to classify the TRL of a small dataset of NTREM articles hand-labelled by an expert. The flexibility of the classifiers will be tested by examining their ability to classify a dataset of HEMCE articles. In Chapters 4, 5 and 6, the interface between using NLP tools to either supplement or replace human analysis in the energetic-materials domain was examined. The work in these studies identified little agreement among experts and raised the point that if experts don not agree on basic points then the reliability of human-annotated data used to train algorithms is brought into question. This chapter will explore this further by presenting a methodology to examine the reliability of the data used to train the aforementioned classifier.

Chapter achievements include:

- Examination of the effectiveness of a range of techniques to mitigate the negative effects of a class imbalance on supervised classifiers in a data-limited environment.
- Exploration of the impact of PCA transformations on classifier performance.

- Comparison of supervised, unsupervised and statistical methods on long text classification by TRL for small datasets.
- Demonstration of the importance of identifying the correct agreement metric for a specific dataset.
- Presentation of methodology used to assess the reliability of human annotations.

In Section 9.1, a literature review of standard methods for mitigating the effects of a class imbalance will be given. In Section 9.2, an overview of the methodology used in the present study is outlined. In Section 9.3 results are presented and discussed. In Section 9.4, the conclusions of this study are discussed and suggestions for further work are made.

9.1 Class imbalance

In Chapter 8, a limitation of using real-world data, class imbalances, were identified as problematic, particularly for small datasets. As a result, a recommendation from the aforementioned chapter was to investigate methods that can mitigate the negative effects on classifier performance of a class imbalance. This section will give an overview of the theory behind such techniques.

Class imbalance, where the frequency of one class is much higher than the frequency of another, is a common problem presented by many real-world datasets [485]. Examples of datasets that commonly experience class imbalances include those developed for medical diagnosis [486, 487], fault diagnosis and anomaly detection [488]. A classifier trained on an imbalanced dataset will be biased towards the major class, thus resulting in poor performance on the minority class. This is a well-studied problem [489]. Techniques for mitigating the effects of a class imbalance can be divided into data-driven and algorithmic.

The most common approach to mitigating the effects of a class imbalance are data-driven sampling approaches [485]. Sampling approaches aim to balance the classes by either increasing the frequency of members in the minority class (oversampling), decreasing the number of members in the majority class (undersampling), or employing a combination of undersampling and oversampling techniques.

Randomised oversampling is a simplistic oversampling approach. This is the process of increasing the frequency of the minority class by duplicating randomly selected members of the class [490]. One limitation of this approach is it increases the probability of overfitting [465, 491]. Randomised undersampling (RUS) is a basic undersampling technique. It randomly selects members of the majority class and deletes them from the dataset. This approach removes potentially valuable data from the dataset, which can negatively impact classifier performance. This approach is helpful for large datasets, but with a small dataset, removing any training examples is generally not advisable as this will likely lower the classifier's overall performance [492].

Synthetic minority oversampling technique (SMOTE) is an oversampling technique developed to avoid the overfitting associated with random oversampling. The most simplistic version of SMOTE creates synthetic instances of the minority class by selecting a member of the minority class and randomly interpolating together k_n of its nearest neighbours to create a new minority class datapoint [493]. More complex versions of the SMOTE algorithm include SMOTE-SVM [494] and MWMOTE [495]. SMOTE has been used widely to counteract class imbalance problems [465] including in text-classification problems [496, 497].

Synthetic labelling is another data-driven approach to mitigating class imbalances. This technique is used when there is an unlabelled dataset and an imbalanced labelled dataset. Items in the unlabelled dataset are labelled using an unsupervised labelling technique and added to the labelled dataset. There are a range of task-specific approaches to synthetic labelling. Examples of synthetic labelling for text classification include clustering [498] and keyword-based approaches.

Algorithmic approaches to class imbalance include modifying the sensitivity of classification algorithms so that errors in minority classes are more costly than in the majority class. This technique was used in Chapter 8, whereby the class-weighting parameter of a range of different supervised classifiers were modified to enable mistakes in under-represented classes to be penalised to a higher degree.

9.2 Methodology

The methodology used in this study consists of two parts: (1) comparing the ability of NLP tools' to classify journal articles by TRL and (2) measuring agreement among experts for classification of journal articles by TRL. First, the two primary datasets used in this study will be described (Section 9.2.1). Second, the methodology for comparing the ability of NLP tools to classify journal articles by TRL will be presented in Section 9.2.2. Finally, the method for measuring agreement among experts for classification of journal articles by TRL will be outlined in Section 9.2.3.

In this study, the definitions of High-TRL and Low-TRL specified in Section 8.2.1 will be followed throughout. In addition, the same lists of High-TRL and Low-TRL keywords specified in Table 8.2 will be used.

9.2.1 Dataset production and preprocessing

The two primary datasets used in this study were produced by the same expert used in the study in Chapter 8, the expert will be referred to as Participant B for the purposes of this study. In addition, these datasets were elicited using the same format as in Section 8.2.1, a questionnaire in the form of a two-column spreadsheet consisting of document titles and a blank column for Participant B to assign the TRL.

The dataset described in Section 8.2.1 was expanded by asking Participant B to assign a TRL to an additional 294 papers. This expanded dataset, referred to as the NTREM dataset, consists of 442 labelled NTREM papers from NTREM 2012, 2015 and 2017-2019. This dataset has a significant class imbalance with 63% of documents labelled as Low-TRL and 37% labelled as High-TRL. This imbalance is expected as most papers published in an academic journal are of Low-TRL content.

This dataset is split into an NTREM training dataset of 353 documents and an NTREM independent test set of 89 documents for the supervised classifiers examined in this study. The NTREM training and NTREM independent test datasets maintain the balance of the classes found in the dataset as a whole. For the unsupervised methods examined, the entire dataset

of labelled NTREM papers will be used to test the performance of the algorithms. A total of 756 unlabelled NTREM papers will be used to train the unsupervised algorithms.

The performance of the trained algorithms will also be evaluated on an additional dataset of 209 HEMCE journal articles. As discussed in Chapter 7, the topic of HEMCE and NTREM papers are similar, but the style in which the journal articles are written varies considerably. The classes of the the HEMCE dataset, like the NTREM dataset, are unbalanced (60% are of Low-TRL and 40% are of High-TRL).

9.2.2 Comparing the ability of different NLP tools to classify journal articles by TRL

This section will describe the method used to compare the ability of supervised and unsupervised classifiers to classify journal articles by TRL. This methodology consists of three parts. First, the best-performing supervised classifier will be identified, then the best-performing unsupervised classifier will be identified, and finally, the best-performing supervised and unsupervised classifiers will be compared.

As in the preliminary study (Chapter 8) two different feature-representation methods, TF-IDF and S-BERT, will be compared. The preprossessing methodology specified in Section 8.2.1 will be followed with one amendment (suggested in Section 8.4): the generation of vector representations of the documents will not be included as part of the preprocessing. This allows for TF-IDF fitting to be performed separately for the test and training sets, avoiding data leakage which can result in biased performance. Generating S-BERT vectors for documents is independent of the wider corpus so fitting the training and test set together does not result in data leakage.

As illustrated by Section 9.2.1, there is a significant class imbalance in the NTREM and HEMCE datasets. In Section 8.1.4, the metric F_1, an adaptation of F-measure for classification, was introduced whereby positive classes were defined as High-TRL. Here, the metric F_0 is introduced. This is an adaptation of Eq. (3.6) using the following values:

- True positives (T_P) are the number of items correctly labelled as the positive class (Low-TRL).

- True negatives (T_N) are the number of items correctly labelled as the negative class (High-TRL).
- False positives (F_P) are the number of items incorrectly labelled as the positive class (Low-TRL).
- False negatives (F_N) the number of items incorrectly labelled as the negative class (High-TRL).

Due to the significant class imbalance, F_1 and F_0 will be skewed. Therefore, a third metric is introduced, $F_{<01>}$, which is defined as the mean of F_1 and F_0 and gives a measure of overall classifier performance. The metrics AUC ROC (Section 8.1.4), S_c (Eq. (8.4)) and A (Eq. (8.3)) will also be used.

Identifying the best supervised classifier

In this study, the following supervised classifiers are examined: Linear Support Vector Machine (SVM), Gaussian Naive Bayes (NB), decision tree, logistic regression and random forest (see Section 8.1.2 for descriptions) for feature representation using S-BERT[1].

First, an overview of the method used for supervised classifiers in this study will be specified. As in Chapter 8, model validation will be performed using nested cross-validation, and the inner and outer loops will consist of 5-fold repeated cross-validation. Section 8.4 recommended that feature selection be included as part of the cross-validation process for a more accurate generalisation error to be calculated and to optimise the number of dimensions (d_n) of the input document vectors. In Chapter 8, KBest was used for feature selection. In this study, KBest for feature selection is replaced with Principal Component Analysis (PCA) as PCA is significantly less computationally expensive[2]. The number of dimensions (d_n) will be optimised over the following grid: [50, 60, 70, 80, 90, 100]. Hyperparameter optimisation is also performed as part

[1]Results were initially generated using both S-BERT and TF-IDF feature-representation methods. However, the results generated using TF-IDF overfit for every classifier. Therefore, with the fact that, in Section 8.3, TF-IDF was outperformed by S-BERT for every supervised classifier, it was decided not to include the results of the TF-IDF feature-representation method in this thesis. Therefore, in this section, the method for S-BERT is discussed only.

[2]PCA rotates and transforms data by keeping the features with the highest variance. Further details on PCA can be found in Ref. [484]. This change was observed to have a limited impact on classifier performance.

of the inner loop. The hyperparameters optimised in this study are unchanged from Section 8.1.3. Model validation will generate an Estimated A with an associated generalisation error.

Following this, document vectors for the NTREM training dataset will be generated using S-BERT. Next, optimisation of the d_n for PCA and hyperparameters is performed using grid-search with 5-fold repeated cross-validation. These optimised values are then input into the classifier, which is then trained on the whole training set. Then, document vectors for the NTREM test set are generated, and classifier performance on the test set is evaluated.

The first experiment with the supervised classifiers will examine the impact of including PCA in the nested cross-validation on the performance of supervised classifiers. This experiment conducts a direct comparison of the effect on algorithm performance between PCA being included in the nested cross-validation (d_n is optimised) and not included in the nested cross-validation (d_n is set arbitrarily to 50[3]).

The second experiment will compare the three best-performing supervised classifiers from the first experiment using various techniques to mitigate the effects of class imbalance. This is a recommendation for further research given in Chapter 8. For each classifier, four different approaches to mitigating a class imbalance will be examined and compared to the classifier's performance with no class-imbalance-mitigation techniques. The four examined approaches are: the algorithmic approach of altering the class-weighting parameter to exaggerated values, RUS, SMOTE, and generating synthetic labels. These approaches are introduced in more detail in Section 9.1. RUS and SMOTE will be included as part of the nested cross-validation. The synthetic-labelling method is independent of all other documents in the dataset, so it is not included as part of the nested cross-validation.

The labelling method used the the keyword-matching algorithm to label the unlabelled documents as High and Low-TRL. Documents labelled as High-TRL were then ranked using the following metric

$$D_C = C_H - C_L. \tag{9.1}$$

This is the difference between the number of counts of High-TRL words (C_H) and Low-TRL words (C_L) in a document. The top 114 scoring documents were identified and added to the training dataset to balance the classes. The desired outcome of this experiment will be the

[3]As explained in Chapter 7, this value was selected as it is the smallest number of dimensions commonly observed in the literature [122] for text representation.

identification of the best-performing supervised classifier and a refinement of the methodology introduced in Chapter 8.

Identifying the best unsupervised classifier

Identifying the best unsupervised classifier will consist of two experiments.

The first experiment for unsupervised classification will compare the performance of K-means clustering classifiers using two different feature-representation methods: S-BERT and TF-IDF. The method described in Section 8.1.2 will be followed with one notable exception: TF-IDF fitting will be applied to the training and test sets separately. Algorithms will be trained on the unlabelled NTREM dataset of 756 documents and evaluated on the labelled NTREM dataset using $F_{<01>}$ and AUC ROC. Silhouette coefficients (S_c) (Eq. (8.4)) will be used to evaluate the quality of generated clusters. The best-performing feature-representation method will be identified in this experiment.

The second experiment will examine the impact of applying PCA transforms prior to clustering on the performance of the K-means clustering classifier with the best-performing feature-representation method. Recommendations for further work in Section 8.4 were to explore using a smaller d_n to perform K-means clustering. K-means clustering algorithms perform better in lower dimensions due to the curse of high dimensionality[4]. As mentioned in Chapter 3, featureless metrics such as the normalised web distance have been developed to resolve this problem [171]. However, reducing the dimensions of the data using techniques such as PCA is a proven way to improve the performance of clustering algorithms [501, 499]. To examine this, PCA transforms to d_n in the range of 2 to 718[5] will be fit to the unlabelled NTREM training set in feature space. The data will then be used to train a K-means classifier. The classifier's performance will then be evaluated on the NTREM dataset using S_c, $F_{<01>}$, A and AUC ROC.

The desired outcome of these experiments is the identification of the best-performing unsupervised classifier and a refinement of the methodology introduced in Chapter 8.

[4]This is the theory that objects become almost equidistant in high dimensions, and metrics such as cosine similarity become meaningless [499]. Furthermore, as the d_n increases, the distance between a data point and its nearest neighbour tends toward the distance to its furthermost point [500].

[5]The maximum d_n that a PCA transform can be applied to is equal to the number of samples in the dataset.

Comparing supervised and unsupervised classifiers

The performance of the best-performing supervised and K-means clustering classifiers described above will be compared to the keyword-matching algorithm (Eq. 8.7) for the NTREM dataset. This process will then be repeated for a dataset of HEMCE journal articles.

9.2.3 Measuring agreement among experts' classifications of journal articles by TRL

This section will present the method used to examine the reliability of the data used to train the NLP tools described in Section 9.2.2. The reliability of the data will be examined by measuring the extent to which a group of experts agree with Participant B (inter-participant agreement) and the extent to which Participant B agrees with their own assigned labels when asked to complete the same task at a later time (test-retest agreement).

The participants for this study are technically trained. They all possess a STEM degree and have several years of experience in the field of energetic materials. The identity of the participants will be kept anonymous. Therefore, participants are referred to by the letters A - D. Some participants (Participants C and D) were paid for their time contribution to this project. Participants A and B were recruited voluntarily. All participants completed the study independently. No background information on the participants was collected as the sample size was too small for any meaningful comparison by background information factors to be conducted. However, it is noted that Participants C and D are from the same organisation.

Data collection was performed over three stages. The first stage was to collect the data used for the NLP analysis in Chapters 8 and 9. Section 9.2.1 describes how the dataset developed in Section 8.2.1 is expanded to 442 labelled NTREM papers from NTREM 2012, 2015, 2017, 2018 and 2019. These datasets were labelled by Participant B.

The second stage generated data for inter-participant comparison. The procedure to obtain TRL labels described in Section 9.2.1 was repeated. Participant A completed the exercise for 104 documents from NTREM 2019. Participants C and D completed the exercise for the same 104 documents from 2019 and 148 documents from 2018.

The third stage generated data for test-retest comparison. The ratings that Participant B generated formed the basis of two studies. It was therefore decided to evaluate the test-retest agreement of this data. This was done by asking the participant to repeat the exercise for 20 randomly selected NTREM documents.

An inter-group comparison will be conducted in 2 stages. Using participant assignments of TRL from NTREM 2019 only for all four participants and using participant ratings from NTREM 2019 and 2018 for Participants A, C and D. Section 6.1 gave an overview of inter-rater reliability metrics. In this study, a comparison of two inter-rater reliability metrics, Cohen's kappa κ and paradox-proof Gwets $AC1$ $\hat{\kappa}_G$, will be given. Test-retest agreement will be measured using $\hat{\kappa}_G$. As discussed in Section 6.1, various benchmark scales are used for interpreting agreement metrics. The Landis and Koch scale will be used in this study. Gwet's statistical approach for interpreting $\hat{\kappa}_G$, described in Section 6.1, will also be used to interpret $\hat{\kappa}_G$.

9.3 Results and discussion

The results will be presented in four sections. First, the results for comparing the ability of NLP tools to classify journal articles by TRL will be presented. In Section 9.3.1, a series of experiments will be conducted to identify the best-performing supervised classifier. Then, in Section 9.3.2, experiments will be conducted to identify the best-performing unsupervised classifier. In Section 9.3.3, the best-performing unsupervised and supervised classifier will be compared to a keyword-matching algorithm. Finally, Section 9.3.4 will present the results of evaluation of reliability of the data.

9.3.1 Identifying the best-performing supervised classifier

As detailed in Section 9.2.2, the method used to identify the best-performing supervised classifier consists of two experiments: examining the impact of including PCA in the nested cross-validation and mitigating the effects of class imbalance on supervised classifiers. The results of these experiments will be presented in the aforementioned order.

Examining the impact of including PCA in nested cross-validation

This experiment examines the impact of including PCA in the nested cross-validation with optimisation of the d_n for PCA, using a training dataset of 353 documents and a test set of 89 documents.

Table 9.1 displays the results of the performance of supervised classifiers with S-BERT for feature representation on the extended study dataset. For each supervised classifier, the table displays the classifier performance with PCA included as part of the nested cross-validation and without.

Algorithm	PCA in CV	Evaluation Metrics			
		$F_{<01>}$	AUC ROC	A	Estimated A
Linear SVM	Yes	0.66	0.67	0.67	0.52 ± 0.05
	No	0.65	0.65	0.66	0.55 ± 0.07
NB	Yes	0.67	0.65	0.66	0.56 ± 0.05
	No	0.49	0.49	0.56	0.56 ± 0.05
Decision tree	Yes	0.58	0.58	0.61	0.55 ± 0.07
	No	0.58	0.58	0.61	0.55 ± 0.06
Logistic regression	Yes	0.56	0.58	0.65	0.62 ± 0.02
	No	0.56	0.58	0.65	0.62 ± 0.02
Random forest	Yes	0.50	0.54	0.64	0.62 ± 0.04
	No	0.42	0.52	0.64	0.63 ± 0.01

Table 9.1: Performance of supervised classifiers using S-BERT for feature representation with PCA included as part of the nested cross-validation (CV) and without. Evaluation is conducted using the metrics: $F_{<01>}$, AUC ROC and A. Estimated A is generated using nested cross-validation.

The results in Table 9.1 show minor differences in the performance of the classifiers with PCA included as part of the nested cross-validation and without. Including PCA in the nested cross-validation should increase the generalisation error (error in estimated A) as the error will now include the variance generated by the different PCA transformations and the additional optimisation. An increased generalisation error for PCA in the nested cross-validation is observed for the decision tree and random forest classifiers. Unsurprisingly, this pair of algorithms are affected similarly since, as described in Section 8.1.2, these algorithms are heavily related. The generalisation error of the linear SVM classifier decreased from 0.07 to 0.05 for PCA being in and not in the nested cross-validation. This was because the stability of hyperparameter optimisations was higher with the PCA in the nested cross-validation. With PCA in the nested

cross-validation, optimisation of the d_n selected 100 on 72% of the loops and optimising the parameter C selected 10 on 60% of the loops. When PCA was not included as part of the nested cross-validation, there was considerably more variation in the selection of C. There was not a single dominant value.

For the logistic regression classifier, no changes to the generalisation error were observed because the optimised d_n was most frequently selected to be 50. Since 50 is the d_n set arbitrarily when PCA is not part of the nested cross-validation, there is no observed change to the generalisation error. This was also observed for the NB classifier.

There are only minor changes to the performance of the classifiers on the test set. It would be expected that including PCA in the nested cross-validation would allow for increased performance as the d_n would be optimised. This is observed for linear SVM, NB, and the random forest classifier. For the decision tree and logistic regression classifiers, there is no difference in the algorithm's performance on the test set between PCA being in the nested cross-validation and not being in the nested cross-validation. This is because, again, the optimised d_n selected by the cross-validation loop was 50, the same d_n set arbitarily when PCA is not part of the nested cross-validation.

To identify if a classifier has overfit, the Estimated A is compared with the A on the test set. If A lies above the error range of the Estimated A, then the algorithm can be identified as overfitting. Including PCA in the nested cross-validation had no impact on whether or not a classifier overfits for all classifiers except for the NB classifier. For the NB with PCA not in the nested cross-validation, $A = 0.56$ which is within the error range of the Estimated $A = 0.56 \pm 0.05$. However, with PCA in the nested cross-validation, the Estimated $A = 0.56 \pm 0.05$ but A increases to 0.66, above the error range of Estimated A. The NB classifier has hence overfit. This classifier selected 100 as the optimal d_n for the PCA. This is a high number and could be the cause of the overfitting. As mentioned in Chapter 8, classifiers trained on small datasets of high-dimensional data are likely to overfit.

To summarise, the method that optimises d_n for PCA lead to improved estimation of the generalisation error. As a result, this method will be applied for all future experiments with supervised classifiers.

The performance of the classification algorithms (with PCA in the nested cross-validation)

will now be discussed. The results in Table 9.1 indicate the linear SVM, NB, and the logistic regression classifiers have overfit. For linear SVM and Gaussian NB, the classifiers were trained on data of 100 dimensions. It is, therefore, unsurprising that they overfit. To avoid overfitting in future experiments, the grid used to optimise the number of output dimensions of PCA should be altered to remove high dimensions. In contrast to linear SVM and NB, the logistic regression classifier's $A = 0.65$ is only slightly larger than the Estimated A plus the generalisation error (0.62+0.02).

Overall, the performance of the classifiers examined here is poor. The values of A are in the range of 0.54 to 0.67. Achieving an $A = 0.50$ is no different from selecting the TRL of the journal papers at random. The classifier that performed the best was the decision tree classifier, where an AUC ROC and $F_{<01>}$ of 0.58 was obtained. These values are suboptimal. There are a range of plausible reasons for the poor performance. However it is hypothesised that this poor peformance is due to the datasets' significant class imbalance. This pronounced class imbalance could result in poor overall performance as the classifier cannot accurately identify members of the under-represented class. The next experiment in this study will address this hypothesis by investigating techniques to mitigate the effects of the class imbalance on the three best-performing classifiers from this study. The random forest and decision tree classifiers did not overfit, and the logistic regression classifier overfit less drastically than the other two classifiers; therefore, the three best-performing classifiers are identified as logistic regression, decision tree and random forest.

Mitigating the effects of class imbalance on supervised classifiers

This experiment will investigate a range of techniques to mitigate the effects of a class imbalance on the three best-performing classifiers from the previous experiment: decision tree, random forest and logistic regression.

Four approaches to mitigating the effects of a class imbalance were examined: RUS, adjusting the class-weighting parameter, SMOTE and synthetic labelling. Results are compared to classifier performance without using any of these techniques. Table 9.2 summarises results.

Table 9.2 shows RUS decreased the overall performance of the decision tree classifier across all metrics. This is unsurprising as RUS removes items from the dataset, giving the classifier

Algorithm	Class Imbalance Method	Evaluation Metrics			
		$F_{<01>}$	AUC ROC	A	Estimated A
Decision tree	None	0.58	0.58	0.61	0.55 ± 0.07
	RUS	0.54	0.55	0.54	0.51 ± 0.05
	Class weighting	0.60	0.60	0.64	0.51 ± 0.05
	Synthetic labels	0.56	0.56	0.56	0.53 ± 0.05
	SMOTE	0.56	0.57	0.57	0.53 ± 0.06
Random forest	None	0.50	0.54	0.64	0.62 ± 0.04
	RUS	0.60	0.61	0.60	0.51 ± 0.07
	Class weighting	0.58	0.58	0.64	0.59 ± 0.05
	Synthetic labels	0.58	0.58	0.58	0.57 ± 0.05
	SMOTE	0.66	0.65	0.70	0.60 ± 0.05
Logistic regression	None	0.54	0.56	0.65	0.62 ± 0.02
	RUS	0.63	0.63	0.64	0.51 ± 0.05
	Class weighting	0.56	0.58	0.66	0.62 ± 0.02
	Synthetic labels	0.65	0.65	0.65	0.61 ± 0.05
	SMOTE	0.63	0.64	0.65	0.63 ± 0.05

Table 9.2: Performance of decision tree, logistic regression and random forest classifiers with and without techniques to mitigate the class imbalance (RUS, class weighting, synthetic labels and SMOTE). Evaluation is conducted using the metrics: $F_{<01>}$, AUC ROC and A. Values of Estimated A are generated using nested cross-validation

fewer examples to learn from and therefore negatively impacting classifier performance. The logistic regression and random forest classifiers saw a decrease in A but an increase in $F_{<01>}$ and AUC ROC. This implies that, whilst the classifier's ability to identify elements of the majority class has decreased (resulting in a lower overall A), its ability to identify members of the minority class has improved (resulting in higher $F_{<01>}$ and AUC ROC). This can be verified by examining the individual classes' $F_{<01>}$ scores. The F_1 for the minority classes (High-TRL) increased when RUS was applied (0.31 to 0.56 and 0.24 to 0.56 for the logistic regression and random forest classifiers respectively). Conversely, the F_0 for the majority classes (Low-TRL) decreased when RUS was applied (0.77 to 0.70 and 0.76 to 0.63 for the logistic regression and random forest classifiers respectively). This result highlights the inability of A to capture classifier performance on datasets with a class imbalance[6]. Whilst the classifiers' ability to distinguish between items has improved, overall A has decreased.

The Estimated A is compared to A for each classifier to identify overfitting. The logistic regression and random forest classifiers achieved A on the test set higher than the Estimated

[6] A is the ratio of correctly classified instances to all instances in a dataset. To illustrate how misleading it is in situations with a class imbalance, consider the following example of a dataset of 10 documents: 9 are in Class A, and 1 is in Class B. A classifier classes them all as Class A. The A in this case would be 0.9, a misleadingly high value.

A plus the generalisation error. This indicates overfitting. RUS decreases the size of the dataset by downsizing the majority class; with less data to train the model on there is a higher risk of overfitting. An increase in the generalisation error was also observed. This reflects the randomised nature of the RUS, which introduces further variance into the classification models.

Altering the class-weighting parameter to exaggerated values in favour of the minority class increased the performance of all examined classifiers. The $F_{<01>}$ and AUC ROC increased for all classifiers indicating an improved ability for the classifiers to distinguish between the classes. For the logistic regression classifier, F_0 does not change, but F_1 increases from 0.31 to 0.35. A similar pattern is observed for the decision tree and random forest classifiers. All classifiers for the altering the class-weighting parameter method did not overfit. Overall, these results indicate that altering of the class-weighting parameter to exaggerated values in favour of the minority class improves classifier performance and does not result in overfitting.

Generating synthetic labels was found to have a varied impact on classifier performance. The logistic regression and random forest classifiers values of $F_{<01>}$ and AUC ROC increased. The overall ability of these classifiers to distinguish between the two classes and classify items correctly improved. The A of the logistic regression classifier remained constant, but the A of the random forest classifier decreased. The individual classes' $F_{<01>}$ scores are examined. The F_0 for the random forest classifier decreased from 0.76 to 0.59, and the minority class F_1 score increased from 0.24 to 0.57. The $F_{<01>}$ scores of the two classes are significantly more balanced, indicating a better classifier that is less likely to be overfitting on the majority class. The majority class has more instances. Therefore, the lower F_0 explains why A of the model has decreased. Table 9.2 indicates the random forest classifier is not overfitting.

The logistic regression classifier's F_1 increased from 0.31 to 0.68, and the majority class F_0 decreased from 0.77 to 0.68. Again, the ability of the classifier to classify individual classes is significantly more balanced, indicating a better overall classifier. Results for A indicate the logistic regression classifier is not overfitting[7].

An increase in the classification algorithms' overall performance is expected as generating synthetic labels results in a significant increase in the number of samples the algorithm is trained on. This gives the classifiers more examples to learn from, which should thus result in improved

[7] A comparison between $A = 0.65$ and the Estimated $A = 0.61 \pm 0.05$ indicates the logistic regression classifier is not overfitting.

performance and a reduced risk of overfitting.

The decision tree classifier's performance decreases across all three examined metrics when synthetic labels are used. The majority class F_0 score fell from 0.69 to 0.57 and the minority class F_1 score increased from 0.46 to 0.56. Whilst using synthetic labels balances out the $F_{<01>}$ scores, meaning the model is less likely to be overfitting on the majority class, as discussed for the random forest classifier, this balancing can decrease the model's performance. Furthermore, synthetic labelling is only effective if the synthetic labels are accurate. Whilst efforts were made to generate high-quality synthetic labels, the methodology may be error-prone. This may have resulted in errors in the performance of the decision tree classifier, which typically exhibits higher variance than random forest classifiers [425]. A comparison between the synthetic-labelling method and the participants' own labels should be conducted to explore this hypothesis further.

SMOTE increased the performance of the random forest classifier across all metrics. The value of F_1 increased from 0.24 to 0.54 indicating SMOTE reduced the impact of the class imbalance. A value of $A = 0.70$ was obtained, which is out of range of the Estimated $A = 0.60 \pm 0.05$ and is a strong indication of overfitting. The optimisation determined $d_n = 100$. Such a high value of d_n is error prone [502], which could have resulted in the observed overfitting. Unlike the synthetic-labelling method, there is no way to evaluate the quality of the synthetically generated points.

The logistic regression classifier's overall performance increased when SMOTE was used. $F_{<01>}$ increased from 0.54 to 0.63, AUC ROC increased from 0.56 to 0.64, and A remained constant. The F_1 increased from 0.31 to 0.55, and F_0 decreased from 0.77 to 0.69. The F scores for the majority and minority classes are not balanced. This may be an indication that there is some overfitting on the majority class. However, a comparison between A and Estimated A does not support this conclusion.

It is expected that using SMOTE would increase classifier performance as there is an increase in the size of the data that the models are trained on. Whilst this increase in performance is observed for the logistic regression and random forest classifiers for the decision tree classifier, the overall performance decreased. $F_{<01>}$ decreased from 0.58 to 0.56, AUC ROC decreased from 0.58 to 0.57 and A fell from 0.61 to 0.57. The F_0 decreased from 0.69 to 0.63, and F_1 increased from 0.46 to 0.49. The ratio of decrease in the majority class to increase in the

minority class is high. This would significantly impact the classifier's performance as there are significantly more instances of the majority class. This model does not appear to be overfitting.

Table 9.2 shows the best-performing algorithm when no sampling method is applied is the decision tree classifier. The generalisation error in this classifier is on the order of 10%, which is similar to the preliminary results seen in Chapter 8. From the Table 9.2, the best-performing approach that does not show clear indications of overfitting is the logistic regression classifier with SMOTE. This achieved an overall $A = 0.65$, which is poor.

To summarise the above, RUS decreased the performance of all classifiers. Adjusting the class-weighting parameter to exaggerated values in favour of the minority class is found to positively impact algorithm performance without resulting in overfitting for all examined approaches. SMOTE and synthetic labelling were found to have a varied impact on classifier performance. The best-performing algorithm that did not overfit was logistic regression using synthetic labels.

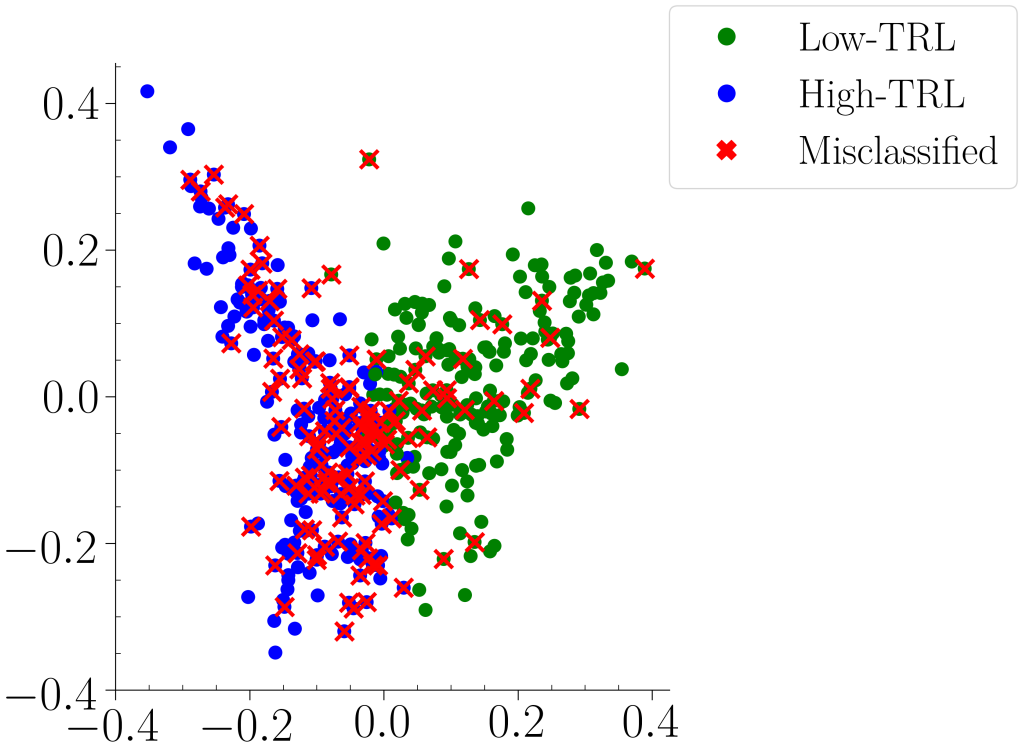
9.3.2 Identifying the best-performing unsupervised classifier

As described in Section 9.2.2, the method used to identify the best-performing unsupervised classifier consists of two experiments: comparing S-BERT and TF-IDF feature-representation methods for a K-means clustering classifier and examining the impact of applying PCA transforms prior to clustering on the performance of a K-means clustering classifier. The results of these experiments will be presented in the order above.

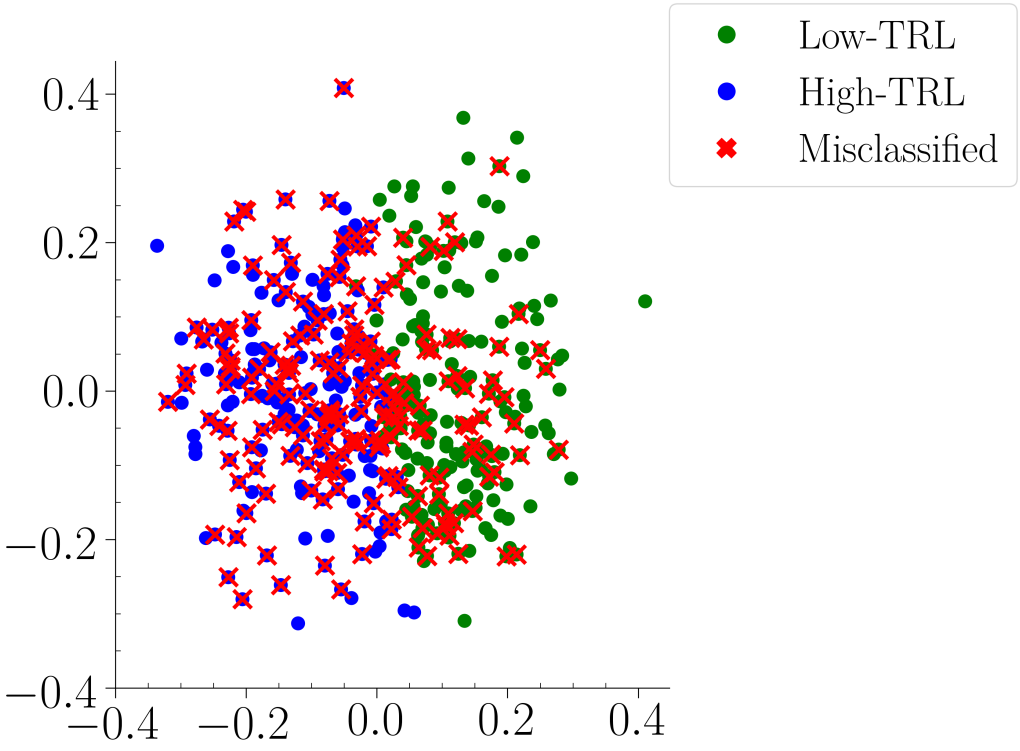
Unsupervised clustering classifier: S-BERT versus TF-IDF

The K-means clustering approach described in Section 9.2.2 is fit to a dataset consisting of document vectors of 718 unlabelled NTREM papers. The algorithm's performance is then evaluated on 442 labelled NTREM papers. Two different feature-representation approaches were examined; S-BERT and TF-IDF. Figure 9.1 shows a visualisation of the results of the K-means clustering classifier, whereby PCA is used to generate 2D representations of the document vectors.

Figures 9.1 (a) and (b) show 2D representations of the document vectors generated using TF-IDF and S-BERT respectively. As with the results presented in Section 8.3 for both feature-



(a) TF-IDF document vectors



(b) S-BERT document vectors

Figure 9.1: Visualisation of K-means clustering classifier performance with feature representation of TF-IDF (a) and S-BERT (b). Each document in the test set is transformed to 2D coordinates using PCA. Documents labelled as High-TRL are shown in blue and those labelled as Low-TRL are shown in green. Misclassified documents have a superimposed red cross.

representation methods, the clusters are not clearly defined. The TF-IDF and S-BERT representation methods generated S_c values of 0.02 and 0.07 respectively. These scores are very close to 0, implying the clusters are ill-defined and overlapping. This indicates a poorly defined decision boundary and hence poor classifier performance.

The performance of the K-means classifier for both TF-IDF and S-BERT feature-representation methods is summarised in Table 9.3.

Algorithm	$F_{<01>}$	AUC ROC	A
K-means TF-IDF	0.64	0.69	0.69
K-means S-BERT	0.54	0.55	0.55

Table 9.3: Performance of the K-means clustering classifier for S-BERT and TF-IDF feature-representation methods. Evaluation is conducted using the metrics: $F_{<01>}$, AUC ROC and A.

K-means clustering using S-BERT did not perform well on the test set. A value of $A = 0.55$ was obtained, which is marginally better than guessing. To further understand the algorithm's poor performance, the F-measure of the classes are examined individually. Values of $F_0 = 0.59$ and $F_1 = 0.49$ were obtained. The difference in these scores shows the class imbalance affects the algorithm. This is in line with results in the literature that show K-means clustering is an algorithm that is affected by imbalanced classes [503]. K-means clustering is subject to the 'uniform effect', whereby the algorithm is prone to developing uniform clusters even if the data is skewed [504]. K-means clustering approaches are prone to assigning members of a majority class to the minority class in order to balance the size of the clusters. Ref. [504] provides a comprehensive overview of the uniform effect in the context of K-means clustering.

In contrast, K-means clustering using TF-IDF showed better performance with $A = 0.69$. This value is higher than the best-performing supervised approach. Values of $F_{<01>} = 0.64$ and AUC ROC $= 0.69$ were also obtained. These values indicate the K-means classifier with TF-IDF classifier can better distinguish between classes than the K-means classifier with S-BERT. Furthermore, $F_0 = 0.64$ and $F_1 = 0.65$ were obtained. These scores are relatively balanced indicating the K-means clustering algorithm is less affected by the class imbalance in TF-IDF feature-representation space.

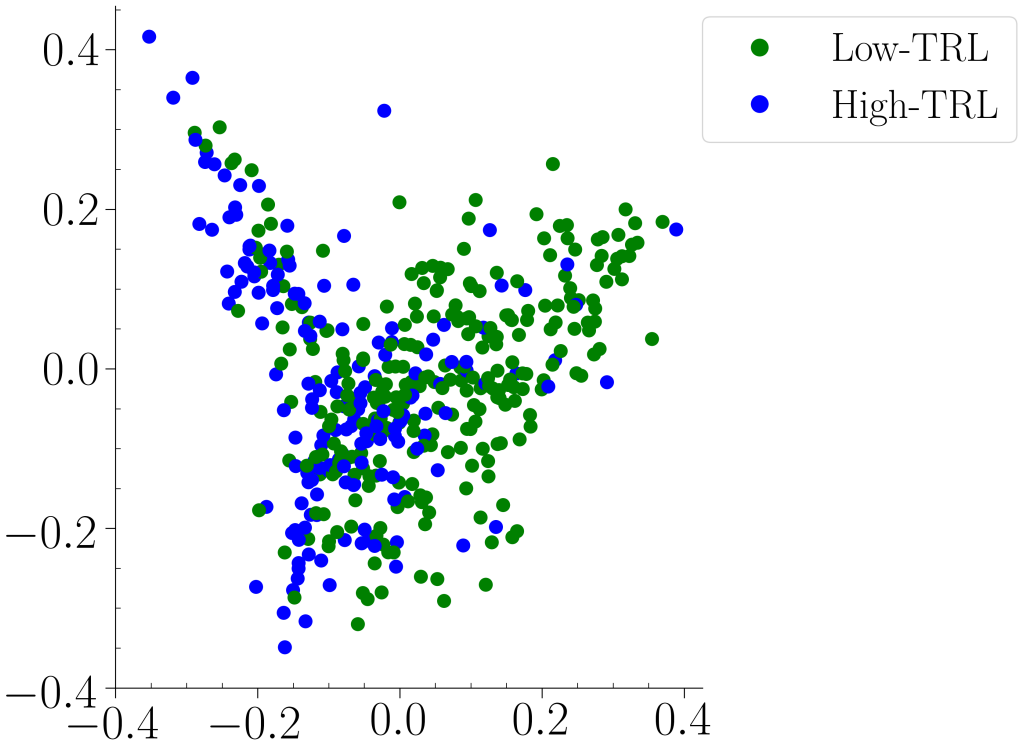
Whilst the K-means clustering with TF-IDF approach performed the best (obtaining $A = 0.65$), its performance is too poor for a system using this method to be implemented reliably. It may be the case that K-means clustering is not the most appropriate clustering algorithm. Therefore,

it is interesting to consider what algorithms may have been a more appropriate choice. One technique to identify other appropriate clustering algorithms is to visualise the labelled dataset and see if any obvious visual patterns emerge that may be more suited to specific clustering algorithms. Figure 9.2 shows 2D PCA transformations of TF-IDF and S-BERT document vectors labelled according to TRL.

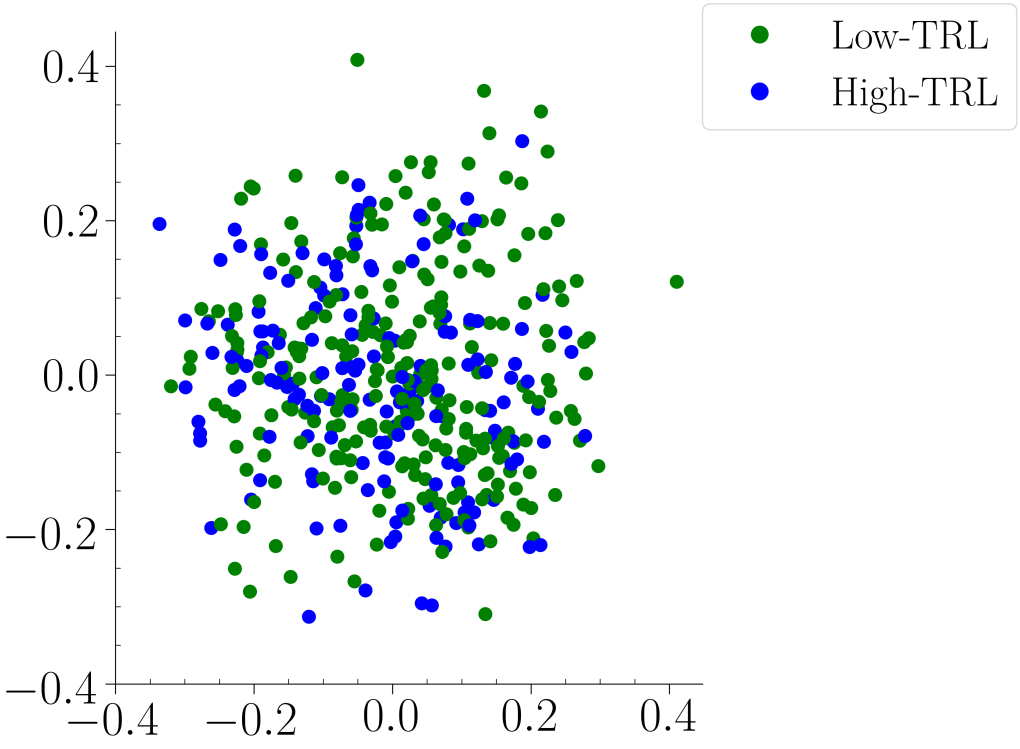
Qualitative analysis of Figures 9.2 (a) and (b) show very undefined decision boundaries between the High-TRL and Low-TRL document vectors for both feature-representation methods. Figure 9.2 (a) shows the majority of High-TRL documents are concentrated on the left of the plot, and Low-TRL documents are concentrated on the right-hand side of the plot. This suggests that a density-based clustering algorithm may yield better results, e.g. DBSCAN [505].

Figure 9.2 (c) shows no clear patterns or decision boundary. This indicates that no one specific clustering algorithm may produce better results. It is noted this analysis is limited in its approach, as Figures 9.2 (a) and (b) show 2D representations of the document vectors. It may be the case that specific patterns may become clearer in higher dimensions. In addition, the choice of algorithm to investigate is somewhat limited by the need for the number of clusters to be a parameter of the algorithm. Clustering algorithms are not designed for classification. The K-means clustering algorithm is one of the few clustering implementations on scikit-learn [425] that has the number of clusters as an input parameter for this parameter.

To conclude, K-means clustering with TF-IDF outperformed K-means clustering with S-BERT. Qualitative evaluation of 2D projections of the labelled data revealed that a density-based algorithm might be a more appropriate clustering algorithm for TF-IDF. However, the number of clusters needs to be an input parameter for the clustering algorithm in order for it to be applied to classification; therefore, exploring this further was ruled out. No noticeable patterns were observed for S-BERT. K-means clustering algorithms are distance-based due to the curse of high dimensionality. Therefore, they tend to perform better in lower dimensions. The next experiment in this section will examine the impact of applying PCA transforms prior to clustering on the performance of the K-means clustering classifier with TF-IDF.



(a) TF-IDF document vectors



(b) S-BERT document vectors

Figure 9.2: 2D PCA representation of document vectors labelled by their assigned TRL. Documents labelled as Low-TRL are shown in green, and documents labelled as High-TRL are shown in blue.

PCA for K-means clustering with TF-IDF

The results of the previous experiment showed the K-means clustering classifier performed best using the TF-IDF feature-representation method. This experiment will examine the impact of applying PCA transforms prior to clustering on the performance of the K-means clustering classifier with TF-IDF.

PCA transforms to d_n from 2 to 718 are fit to the unlabelled NTREM training set in TF-IDF feature-space. This data is then used to train the classifier.

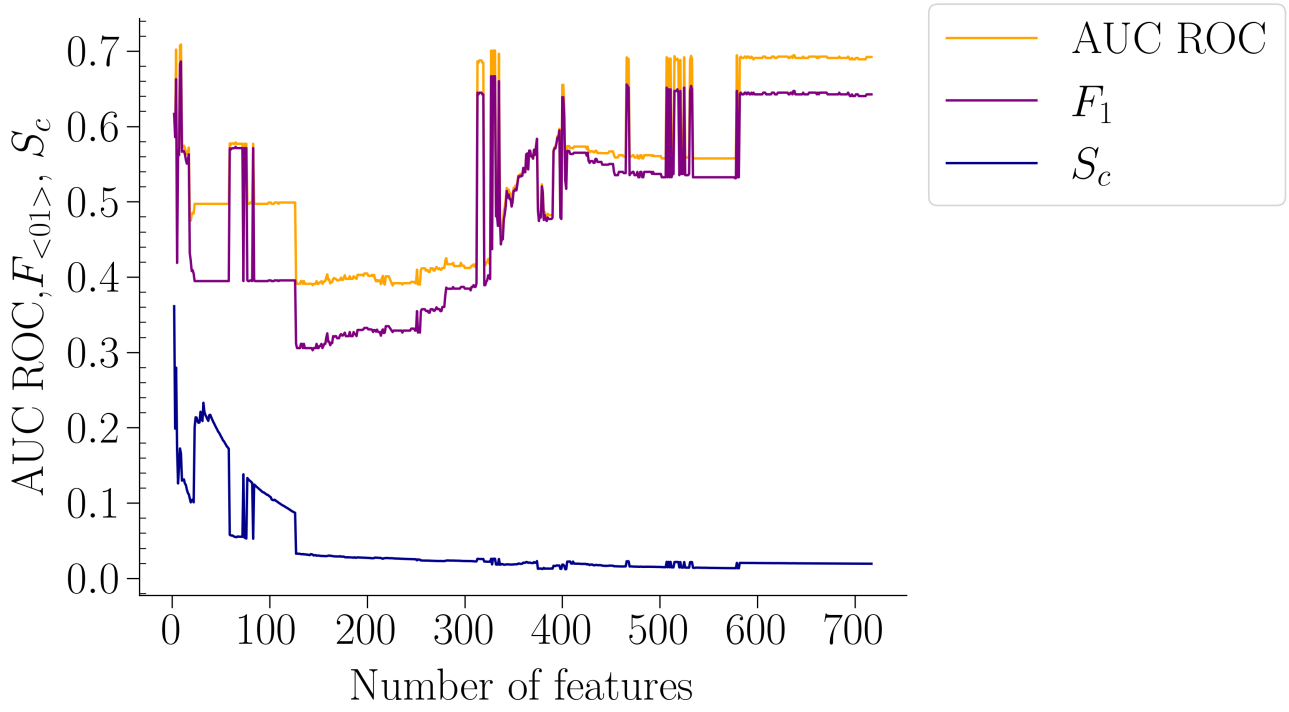


Figure 9.3: AUC ROC, $F_{<01>}$ and S_c of K-means clustering classifier with TF-IDF whereby a PCA transform has been applied versus the d_n after reduction.

Figure 9.3 shows that, as the dimensions of the data decrease, the S_c (Eq. (8.4)) decreases. This is expected as the S_c uses the mean distances between clusters to measure how well defined the clusters are and, at higher dimensions, distance is less defined. The S_c decreases with d_n rapidly. For a d_n higher than 60, the S_c is below 0.20. The effect of the d_n on $F_{<01>}$ and AUC ROC is less clear. For numbers of features less than 300, lower values of $F_{<01>}$ and AUC ROC are observed. This is somewhat expected as the loss of information to transform data to these dimensions is significant. There is, however, some exception to this observation: for dimensions 8 and 9, high values for $F_{<01>}$, AUC ROC and S_c are obtained. This may be due to random variation in the PCA algorithm or because these are the most optimal values of

d_n where all information necessary for classification is retained. For d_n above 400, the values of $F_{<01>}$ and AUC ROC remain constant. For $F_{<01>}$ and AUC ROC evaluation, it is expected the performance of the clustering algorithm will decrease as the d_n decreases as information is lost. It may be the case that for a d_n above 400 all necessary information for performing a classification is retained. PCA is an algorithm with inherent variance. The variance observed in Figure 9.3 reflects this.

From Figure 9.3, the optimal d_n can be identified by examining the peak values of AUC ROC, $F_{<01>}$ and S_c. The optimal d_n is 9. At this d_n, values of 0.71, 0.69 and 0.17 are obtained for the AUC ROC, $F_{<01>}$, and S_c respectively. This minor increase in performance from the results in Table 9.3 demonstrates that using PCA as part of the classification process can increase the model's performance if the PCA is performed to an optimised d_n. This conclusion is in agreement with other studies in the literature [501, 499].

To conclude, the best-performing unsupervised algorithm was K-means clustering using TF-IDF for feature representation and performing a PCA transform to 9 dimensions. There are alternative methods of feature reduction and feature selection that may prove to be more fruitful; examples include non-negative matrix factorisation [506] and linear discriminant analysis [507]. Identification of the best feature-reduction method is normally performed on a trial-and-error basis. Future studies should explore a range of different feature reduction/selection methods.

9.3.3 Comparing supervised and unsupervised classifiers

In this section, the performance of the best-performing supervised and unsupervised algorithms from Sections 9.3.1 and 9.3.2 for classification of journal articles by TRL are compared. Section 9.3.1 examined a range of supervised classification algorithms and approaches to tackling class imbalances. This work identified the best-performing supervised classifier as a logistic regression classifier using S-BERT for feature representation and the synthetic-labelling approach to mitigate the class imbalance. This algorithm will henceforth be referred to as 'logistic regression (S-BERT SL)'. In Section 9.3.2, unsupervised clustering algorithms were examined. This analysis concluded the best-performing unsupervised algorithm was K-means clustering using TF-IDF for feature representation and performing a PCA transform to 9 dimensions. This

algorithm will henceforth be referred to as 'K-means clustering (TF-IDF 9)'.

The performance of K-means clustering (TF-IDF 9), logistic regression (S-BERT SL) and a baseline performance measure (the keyword-matching algorithm) on the NTREM dataset is summarised in Table 9.4.

Algorithm	$F_{<01>}$	AUC ROC	A	Estimated A
Logistic regression (S-BERT SL)	0.65	0.65	0.65	0.61 $\pm$0.05
K-means clustering (TF-IDF 9)	0.69	0.71	0.69	
Keyword matching	0.58	0.62	0.58	

Table 9.4: Results of K-means clustering (TF-IDF 9), logistic regression (S-BERT SL) and a keyword-matching algorithm for document classification by TRL on the NTREM dataset.

Table 9.4 shows the keyword-matching algorithm is outperformed by both the supervised and unsupervised algorithms. In Chapter 8, the performance of the keyword-matching algorithm was evaluated on a smaller dataset of 148 documents. The results can be found in Table 8.3. The performance of the keyword-matching on this larger dataset dropped considerably, achieving an AUC ROC of 0.73 in the Chapter 8 study compared to 0.62 in the present study. This could be because keywords are a good indicator of TRL for some but not all journal articles so on a larger dataset the performance of the model drops. This ties into the discussion in Chapter 4 that different conceptual ideas can be portrayed using similar words, as using keywords alone is insufficient to identify if a particular concept is being discussed. As suggested in Chapter 8, one method of improving this algorithm would be to expand the keywords list. Alternatively, given the process of assigning TRL is subjective and the labels are human-generated, the labels generated for this dataset may be inconsistent or erroneous. The quality of the dataset will be evaluated in Section 9.3.4. The keyword-matching algorithm was unaffected by the class imbalance since F_0 and F_1 were equal values. This is an entirely unsupervised statistical algorithm. It is therefore unaffected by the class imbalance.

K-means clustering (TF-IDF 9) achieved the best performance of the algorithms investigated, obtaining values of 0.69, 0.71 and 0.69 for $F_{<01>}$, AUC ROC and A respectively. Whilst these values demonstrate this algorithm is better than random guessing, the overall performance is still suboptimal and error-prone, meaning it cannot be reliably deployed in a real-world application. Values of 0.68 and 0.69 were obtained for F_1 and F_0 respectively, indicating the K-means clustering (TF-IDF 9) algorithm is affected minimally by the class imbalance as this is not the cause of the poor performance.

As discussed in Section 9.3.1, the logistic regression (S-BERT SL) classifier was largely unaffected by the class imbalance. However, the model's overall performance is suboptimal, obtaining $A = 0.65$. The logistic regression (S-BERT SL) classifier did not overfit. However, the obtained generalisation error was approximately 8%. This is on the order of the results in Table 8.3. To understand the cause of this generalisation error, the stability of the hyperparameters and the d_n optimised as part of the inner loop of the nested cross-validation was investigated. Logistic regression (S-BERT SL) optimised one hyperparameter C. Surprisingly C was stable for all optimisations performed. However, the d_n was found to vary significantly. Figure 9.4 displays the frequency of selection of d_n for PCA as a result of optimisation in the inner loop of nested cross-validation.

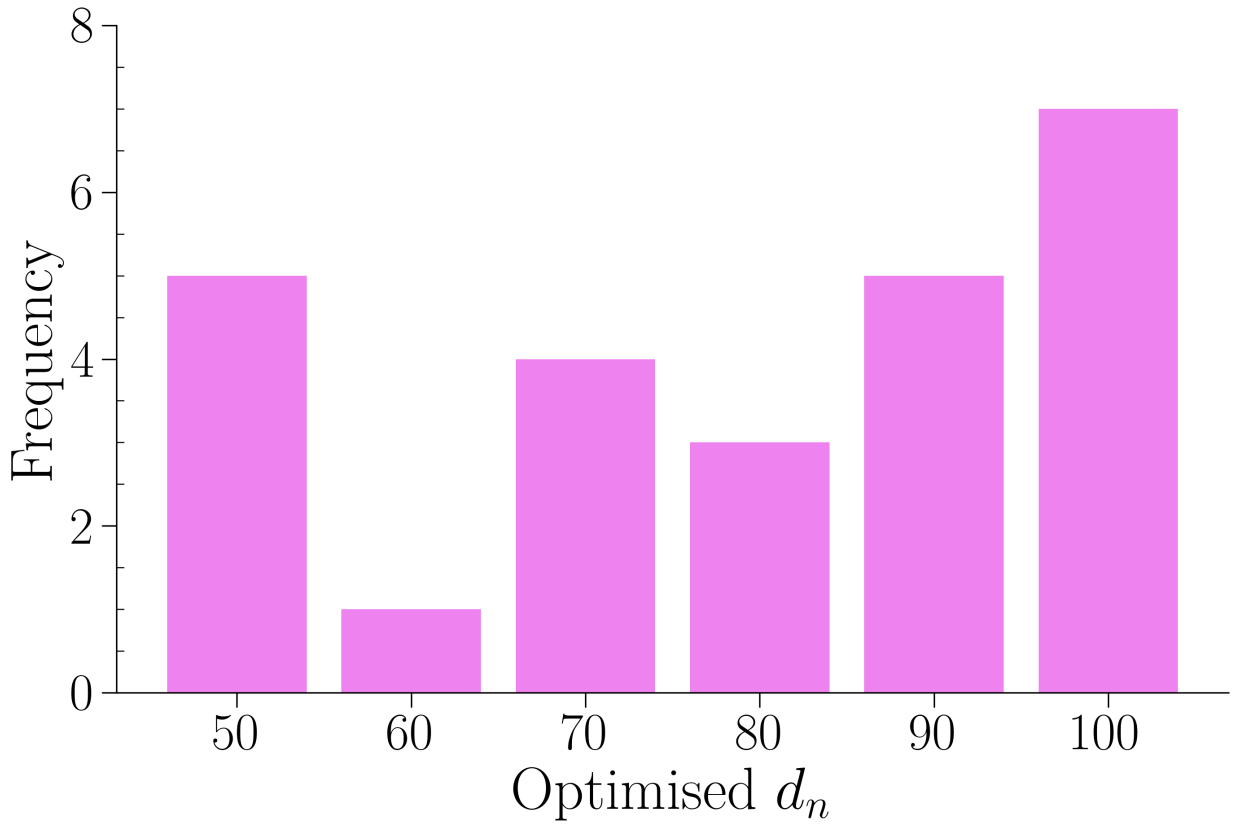


Figure 9.4: Frequency of selection of d_n for PCA as a result of optimisation in the inner loop of nested cross-validation over the grid [50,60,70,80,90,100], for logistic regression (S-BERT SL).

Figure 9.4 shows a range of values of d_n were deemed optimal depending on what portion of the data the algorithm was trained on. This variance is unsurprising for a small dataset. This observed lack of stability in d_n is likely a key cause of the generalisation error. This is a clear indication that feature-reduction methods can be a significant cause of variance and therefore impact the overall performance of an algorithm.

The performance of the classifiers in this study are suboptimal. When the values of A and AUC ROC in Table 9.4 are compared to those obtained by the study conducted in Chapter 8 (Table 8.3), there is a significant difference in performance. One possible reason for this is that the supervised results in Table 8.3 exhibited severe overfitting that was not picked up by the cross-validation, possibly due to feature-selection not being included as part of the nested cross-validation. The decrease in performance of the K-means clustering classifiers may be because the algorithm is fitted on fewer documents in Table 9.4. The performance of the keyword-matching algorithm also decreased. The overall decrease in performance could be due to the increased class imbalance (though analysis completed earlier in this section suggests otherwise) or due to errors in labelling the dataset.

The flexibility of the K-means clustering (TF-IDF 9), logistic regression (S-BERT SL) and keyword-matching classifiers will be examined by testing their performance on the HEMCE dataset decribed in Section 9.2.1.

Algorithm	$F_{<01>}$	AUC ROC	A	Estimated A
Logistic regression (S-BERT SL)	0.55	0.55	0.54	0.61 $\pm$0.05
K-means clustering (TF-IDF 9)	0.61	0.61	0.62	
Keyword-matching	0.48	0.58	0.50	

Table 9.5: Results of K-means clustering (TF-IDF 9), logistic regression (S-BERT SL) and keyword-matching classifiers on the HEMCE dataset.

Table 9.5 shows the performance of the K-means clustering (TF-IDF 9) and logistic regression (S-BERT SL) classifiers decreased for the HEMCE dataset. The algorithms were fit/trained on NTREM papers. The performance on this dataset shows a more diverse training dataset is needed to generate an algorithm flexible enough to perform classification on different journals.

In addition, Table 9.5 shows the performance of the keyword-matching algorithm declined from an $A = 0.58$ on the NTREM dataset to $A = 0.50$ on the HEMCE dataset. An A of 0.50 is no better than flipping an unbaised coin to assign labels. This decline in the performance of the keyword-matching algorithm is examined further by considering the performance of individual classes. Values of 0.35 and 0.60 were obtained for F_0 and F_1 respectively. This implies the Low-TRL keyword list may be less effective than the High-TRL keyword list for HEMCE papers. This could be because the experts who generated the keyword lists developed the lists with NTREM in mind or because HEMCE uses different terminology than NTREM.

Despite the class imbalance being less pronounced for the HEMCE dataset, it is found to significantly impact the performance of both the logistic regression (S-BERT SL) and K-means clustering (TF-IDF 9) classifiers. The logistic regression (S-BERT SL) classifier obtained values of 0.57 and 0.50 for F_0 and F_1 respectively. The logistic regression (S-BERT SL) classifier used a synthetic-labelling method to label High-TRL documents and add them to the dataset to mitigate the effects of the class imbalance. As shown in Eq. (9.1), the synthetic-labelling method is based upon the keyword-matching algorithm, which, as explained above, is less accurate for the HEMCE dataset. This, therefore, explains the decline in the performance of the algorithm. K-means clustering (TF-IDF 9) obtained an F_0 of 0.68 and an F_1 of 0.54. Overall the K-means clustering (TF-IDF 9) algorithm performed the best on the HEMCE dataset. K-means clustering (TF-IDF 9) and logistic regression (S-BERT SL) outperformed the statistical keyword-matching algorithm indicating that, even in small-data environments, ML approaches can yield better results than statistical algorithms.

To conclude, the performance of these algorithms was suboptimal. The best-performing algorithm on both datasets was K-means clustering (TF-IDF 9). Both ML approaches, K-means clustering (TF-IDF 9) and logistic regression (S-BERT SL), outperformed the statistical keyword-matching algorithm on both datasets. The performance of all three algorithms declined from the results shown in Chapter 8. This could be due to errors in labelling the dataset. Section 9.3.4 will investigate this further.

9.3.4 Results of comparing experts' classifications of journal articles' TRL

This section will examine the reliability of the data used to train the NLP tools in previous sections. The method for this analysis is presented in Section 9.2.3.

Inter-participant agreement

Table 9.6 displays the inter-participant agreement between all four experts on their classification of 108 NTREM articles by TRL. Cohen's Kappa (κ) is used to calculate agreement, and the Landis and Koch benchmark scale is used for interpretation (see Table 6.2).

Pair	κ	Interpretation
A & B	0.08	Slight
A & C	0.33	Fair
A & D	0.33	Fair
B & C	0.20	Slight
B & D	0.33	Fair
C & D	0.39	Fair

Table 9.6: κ for all possible pairings of experts on their classification of 108 NTREM articles by TRL. Interpretation of κ is completed using Landis and Koch benchmark scales.

Table 9.6 shows only slight agreement between A & B and B & C. Fair agreement is observed between pairs C & A, A & D, B & D, and D & C. Classifying a document by TRL is a subjective process, and different experts, even when presented with the same set of keywords and definitions for High-TRL and Low-TRL, will likely have different approaches to evaluating the TRL of a document. However, given the MoD has nine TRL levels, it was expected that reducing this to a two-point scale would reduce the number of instances where experts would disagree on TRL assignments. As a result, the low levels of agreement observed here are unexpected.

However, as mentioned in Section 6.1, there are certain conditions under which κ is an invalid metric. This is referred to as the kappa paradox, which can lead to low κ for high agreement when the marginals of a contingency table are unbalanced (bais) or asymmetric (prevalence). To examine if this is the cause of the low observed κ, contingency tables are obtained for each pair and values of R_{1A}, R_{2A}, R_{1B}, and R_{2B} are computed. These values are then compared to check for bias and prevalence. This is displayed in Table 9.7.

Pair	R_{1A}	R_{2A}	R_{1B}	R_{2B}	Bias	Prevalence
A & B	76	28	94	10	✗	✓
C & B	65	39	94	10	✗	✓
D & B	74	30	94	10	✗	✓
A & C	76	28	65	39	✗	✓
A & D	76	28	74	30	✗	✓
C & D	65	39	74	30	✗	✓

Table 9.7: Marginals for all possible pairings of experts on their classification of 108 NTREM articles by TRL. The marginals are examined for bias and prevalence.

Table 9.7 shows that for all combinations of pairs give symmetric but unbalanced marginals, i.e. no bias but prevalence. This is an example of the second kappa paradox introduced in Section 6.1. It is noted some combinations of pairs, notably C & D, A & D and A & C, have less

unbalanced marginals. These pairings resulted in the three highest κ scores given in Table 9.6. Observing unbalanced marginals is not surprising since the NTREM datasets have a clear class imbalance. This implies the results in Table 9.6 do not reflect the true agreement. Therefore, the analysis will be repeated using the more paradox-resistant Gwet's $\hat{\kappa}_G$.

Agreement metrics are now recalculated using $\hat{\kappa}_G$. Interpretation of $\hat{\kappa}_G$ and obtaining an error in $\hat{\kappa}_G$ (SE) is completed using the statistical approach described in Section 6.1 with Landis and Koch benchmark scales. Results are displayed in Table 9.8 alongside the interpretation of the κ score shown in Table 9.7.

Pair	$\hat{\kappa}_G$	SE	Interpretation of $\hat{\kappa}_G$	Interpretation of κ
A & B	0.59	0.08	Moderate	Slight
A & C	0.50	0.09	Fair	Fair
A & D	0.69	0.07	Moderate	Fair
B & C	0.47	0.09	Fair	Slight
B & D	0.55	0.08	Moderate	Fair
C & D	0.50	0.09	Fair	Fair

Table 9.8: $\hat{\kappa}_G$ and associated error (SE) for all possible pairings of experts on their classification of 108 NTREM articles by TRL. Interpretation of $\hat{\kappa}_G$ is completed using a statistical approach described in Section 6.1 with Landis and Koch benchmark scales. Interpretation of κ using Landis and Koch benchmark scales is computed for each pair.

The results in Table 9.8 show that for all pairings, except A & C and C & D, κ generated underestimates of the agreement between individuals. The reasons for this can be established by examining Table 9.7. All pairings showed an imbalance in the marginal distributions. Therefore, it is expected that κ would provide underestimates of agreement between pairs. Pairings A & C and C & D show the least imbalance in their marginals. Therefore, these pairs should see a minor variation between the interpretation of κ and $\hat{\kappa}_G$. The pairing with the most unbalanced marginals is A & B. This is predictably the pairing in which there is the most change in the interpreted agreement between the two metrics (from moderate to slight). This analysis has shown the importance of evaluating the appropriateness of an agreement metric before using it.

Participants A, B and D are the most 'agreeable' participants, each having a moderate agreement with two other experts and fair agreement with one other expert. Since the majority of analysis in this chapter is based on the labels generated by Participant B, this is positive. However, overall, participant B's agreement with the other participants is moderate at best, indicating the data used to train and evaluate the performance of the algorithms is not reliable.

Three participants (including Participant B) classified a total of 252 NTREM documents by TRL. The process of identifying an appropriate agreement metric by examining prevalence and bias in the values of R_{1A}, R_{2A}, R_{1B}, and R_{2B} is repeated. The results are displayed in Table 9.9.

Pair	R_{1A}	R_{2A}	R_{1B}	R_{2B}	Bias	Prevalence
C & B	161	91	157	95	✗	✓
D & B	190	62	157	95	✗	✓
D & C	190	62	161	91	✗	✓

Table 9.9: Marginals for all possible pairings of experts on their classification of 252 NTREM articles by TRL. The marginals are examined for bias and prevalence.

Table 9.9, similarly to the results in Table 9.7, shows symmetrical unbalanced marginals for all pairs of participants. This indicated the data is subject to the second kappa paradox (prevalence). Therefore, $\hat{\kappa}_G$ will be used to calculate the agreement among participants. These results are displayed in Table 9.8 along with the agreement calculated using κ for comparison. The standard error in the obtained $\hat{\kappa}_G$ is also shown. Interpretation of $\hat{\kappa}_G$ is completed using a statistical approach described in Section 6.1 with Landis and Koch benchmark scales.

Pair	$\hat{\kappa}_G$	SE	Interpretation of $\hat{\kappa}_G$	κ	Interpretation of κ
B & C	0.42	0.06	Fair	0.34	Fair
B & D	0.51	0.06	Moderate	0.37	Fair
D & C	0.57	0.05	Moderate	0.42	Moderate

Table 9.10: $\hat{\kappa}_G$ and associated error (SE) for all possible pairings of experts on their classification of 108 NTREM articles by TRL. Interpretation of $\hat{\kappa}_G$ is completed using a statistical approach described in Section 6.1 with Landis and Koch benchmark scales. For comparison, agreement using κ is calculated. Interpretation of κ using Landis and Koch benchmark scales is computed for each pair.

Table 9.10 shows that, whilst for the pairs B & C and D & C there is no difference in the interpretation of the agreement when using $\hat{\kappa}_G$ and κ, this is not the case for B & D. Using κ for B & D would lead to an underestimate of the agreement. Table 9.9 shows this is the pair with the most unbalanced marginals; therefore, observing a difference between the interpretation of $\hat{\kappa}_G$ and κ is expected. If the agreement between pairs on the smaller dataset (Table 9.8) is compared to the agreement on this larger dataset, it is observed that for pairs B & C and B & D, there is no change in the interpreted agreement. However, the agreement of the pair D & C increased from fair to moderate. This indicates there were higher levels of agreement between the participants on the 2018 proceedings as opposed to the 2019 proceedings. Again, the agreement between

Participant B (who labelled the NTREM datasets) and the other participants is lower than expected.

Thus far, the agreement between pairs of participants on assigning the TRL of journal articles has been calculated. Here, the agreement across participants will be calculated using $\hat{\kappa}_G$ for the dataset of 104 documents and participants A, B, C & D and the dataset of 252 documents and participants B, C & D. The results are displayed in Table 9.11.

Dataset	Participants	$\hat{\kappa}_G$	Interpretation of $\hat{\kappa}_G$
252 documents	B,C and D	0.49847 ± 0.04381	Moderate
104 documents	A, B, C and D	0.55131 ± 0.06087	Moderate

Table 9.11: $\hat{\kappa}_G$ for the dataset of 104 documents and participants A, B, C, D)and the dataset of 252 documents and Participants B, C, D.

Table 9.11 shows the agreement across all participants for both datasets can be interpreted as 'moderate'. The moderate agreement may be because the definition of TRL given to the participants may not have been specific enough. Alternatively, there may have been several papers in the proceedings that were on the boundary between Low-TRL and High-TRL so were difficult to assign a TRL to. The feedback from the participants was that some articles were more straightforward to assign a TRL to than others, where it was more subjective. Future studies could investigate this further by asking participants to give a 1-5 confidence score in their assignment of TRL. This would not only allow those documents that are difficult to classify to be identified but enable the uncertainty in the participant's assignment of a label to be measured.

The results must be considered in the context of the study's limitations. Firstly, there was a limited sample size of participants and documents to assign a TRL. The results would be more accurate if a larger sample size were used. Future studies of this nature should gather more data. Secondly, some participants were paid for their participation, and others were recruited voluntarily. This may explain the low agreement as it may have been the case that those that were paid assigned more time to the task than those that were not and ultimately approached the task using different methods on account of this time allowance. Future studies should avoid this.

Test-retest

In order to assess the reliability of the data, Participant B was asked to complete a retest study. This was done on a separate occasion over a month after prior data collection had been completed on a sample size of 20 documents. $\hat{\kappa}_G$ was used to measure reliability. A value of $\hat{\kappa}_G = 0.55 \pm 0.06$ was obtained. Interpretation of this value is completed using a statistical approach detailed in Section 6.1. This can be interpreted as moderate agreement to a confidence level of 95%. Whilst this restest study was conducted on a small sample size, this result implies a lack of consistency in Participant B's judgement of document TRL. This lack of consistency in the participant's judgements could be because their understanding of TRL evolved with time or because the subset of documents chosen were difficult to classify. Regardless of the reason, the lack of consistency in the labelling of Participant B may explain the poor performance of the algorithms in Sections 9.3.1, 9.3.2 and 9.3.3. This could also explain why the results in Chapter 8 were more promising than the results in this chapter: unlike the study in Chapter 8, this study used a dataset generated over several months.

9.4 Conclusions and further work

This chapter presented the second part of a two-chapter study addressing the limitations of using NLP tools in a small-data environment. Recommendations to refine the Chapter 8 methodology were applied to refine the methodology presented in Chapter 8. The ability of supervised and unsupervised classifiers to classify the TRL of a small dataset of NTREM articles hand-labelled by an expert was compared. In addition, the reliability of the data used to train these classifiers was examined.

In Chapter 8, a limitation of using real-world data, class imbalance, was identified as problematic, particularly for small datasets. It was therefore recommended to investigate methods that can mitigate the adverse effects on classifier performance of a class imbalance. Section 9.1 provided an overview of standard approaches to dealing with class imbalances that are suitable for data-limited environments.

Section 9.2 presented an overview of the methodology used in this study. Metrics to examine the performance of each class in the dataset were introduced (F_0 and F_1). A description of the

primary datasets NTREM and HEMCE was given. Then an overview of the method used to compare the ability of NLP tools to classify journal articles by TRL was given. This consisted of three parts. The first part was the method for identifying the best-performing supervised classifier (comprised of two experiments: examining the impact of including PCA in nested cross-validation and various techniques to mitigate the effects of class imbalance).

The first experiment compares the performance of linear SVM, naive bayes (NB), decision tree, logistic regression and random forest classifiers using a small dataset of 442 NTREM papers. S-BERT was used for feature representation. As this is a small dataset, nested cross-validation was used to optimise hyperparameters and generate estimates of the model's performance. This experiment examined the impact of including optimisation of the number of dimensions (d_n) and therefore PCA as part of the nested cross-validation by comparing it to an implementation whereby PCA was performed arbitrarily to $d_n = 50$ outside of the nested cross-validation. This was a recommendation from Section 8.4 as it was expected that including PCA optimisation as part of nested cross-validation would generate a more accurate generalisation error and increase the model's performance. The second experiment examined the impact of a range of techniques to mitigate the effects of a class imbalance (introduced in Section 9.1) on the performance of supervised classifiers. The investigated techniques are random undersampling (RUS), adjusting the class-weighting parameter, Synthetic minority over sampling technique (SMOTE) and a synthetic-labelling algorithm based upon a keyword-matching algorithm. Performance was evaluated on the NTREM dataset.

The second part of the study methodology was to identify the best-performing unsupervised classifier. This also consisted of two experiments (comparing the performance of K-means classifiers using two different feature-representation methods, S-BERT and TF-IDF, and examining the impact of applying PCA transforms prior to clustering on the performance of the K-means clustering classifier). The first experiment compared the performance of a K-means clustering classifier using two different feature-representation methods, S-BERT and TF-IDF. The algorithm was trained using a dataset of 718 unlabelled NTREM articles and tested on 442 labelled NTREM articles. K-means clustering algorithms are distance-based algorithms. Therefore, they suffer from the curse of high dimensionality and tend to perform better in lower dimensions. Therefore, the second experiment sought to optimise the performance of the best-performing K-means clustering classifier by examining the effect of applying PCA transforms

to values of d_n in the range of 2 to 718 on classifier performance.

The third part of the study methodology was to compare the performance of the best-performing supervised and unsupervised classifiers on the NTREM and HEMCE datasets. The final part of this section introduced a methodology to compare experts' classifications of journal articles' TRL. This aimed to investigate the reliability of the data used to train the algorithms in the previous sections. Three additional experts were asked to classify a series of NTREM papers by TRL, and inter-participant agreement was computed. The expert who generated the datasets (Participant B) was asked to repeat the classification task for a small subset of documents. This data was then used to measure participant B's agreement with their labels. A comparison of two agreement metrics was conducted: Gwets $AC1$ $\hat{\kappa}_G$ and Cohen's kappa (κ). Interpretation of these values was performed using the Landis and Koch benchmark scales.

Section 9.3 presented the results. The first set of results (Section 9.3.1) presented a series of experiments to determine the best-performing supervised classifier. First, the results of examining the impact of including PCA in nested cross-validation were presented. Despite the expectation that including PCA in the nested cross-validation would increase the performance of the classifiers, only marginal changes were observed. For the linear SVM, NB, and random forest classifiers, the model's performance increased with PCA in the nested cross-validation loop. However, for the decision tree and logistic regression classifiers, no difference between the two methods was observed, as for the method where d_n was optimised, d_n was determined to be 50, the same d_n set arbitrarily for the method where PCA is not part of the nested cross-validation. More accurate generalisation errors were observed for the majority of classification algorithms. Therefore, it was determined the method that used optimisation of d_n for PCA in the nested cross-validation loop should be used going forward. Values of A in the range of 0.67 to 0.54 were observed, which is suboptimal. It was hypothesised that the poor classifier performance was due to the class imbalance.

The three best-performing classifiers from this experiment: logistic regression, decision tree and random forest, are used to investigate this hypothesis in the second experiment. This experiment examined the impact of techniques (RUS, adjusting the class-weighting parameter, synthetic labelling and SMOTE) to mitigate the effects of a class imbalance on classifier performance. Results indicated that altering the class-weighting parameter to exaggerated values in favour of the minority class increased the performance of all examined classifiers without

overfitting. RUS decreased the performance of all classifiers. SMOTE varied on classifier performance: increasing the performance of the random forest and logistic regression classifiers but decreasing performance for the decision tree classifier. Logistic regression with SMOTE was found to overfit. Using synthetic labels also had a varied impact on classifier performance, significantly increasing the performance of the logistic regression algorithm and reducing the performance of the decision tree and random forest algorithms. The best-performing algorithm was logistic regression using synthetic labels. This obtained $A = 0.65$.

In Section 9.3.2, the second set of results were presented. This consisted of two experiments to identify the best-performing unsupervised classifier. The first of these experiments compared the performance of the K-means clustering classifier using TF-IDF and S-BERT for feature representation. The results indicated using TF-IDF outperformed using S-BERT. Using S-BERT, the classifier obtained $A = 0.55$. This is marginally better than random guessing. Using TF-IDF, a value of $A = 0.69$ was obtained, outperforming the best-performing supervised algorithm. It was noted that K-means clustering algorithms are prone to the 'uniform effect', which can be problematic when the data has unbalanced classes. Despite this, the performance of the K-means clustering method using TF-IDF on each class was relatively balanced, obtaining F_0 and F_1 of 0.64 and 0.65 respectively. Qualitative evaluation of 2D projections of the labelled data revealed a density-based algorithm might be a more appropriate clustering algorithm for TF-IDF. However, the number of clusters needs to be an input parameter for the clustering algorithm in order for it to apply to clustering. Therefore, such an approach was ruled out. No noticeable patterns were observed for S-BERT. The second experiment examined the effect of applying PCA transforms to d_n in the range of 2 to 718 on classifier performance. This analysis determined that performing PCA transforms can optimise classifier performance. The best-performing value of d_n was 9, and this obtained a $F_{<01>} = 0.71$.

In Section 9.3.3, the results of comparing the best-performing supervised and unsupervised classifiers identified in the studies in Sections 9.3.1 (logistic regression (S-BERT SL)) and 9.3.2 (K-means clustering (TF-IDF 9)) to a keyword-matching algorithm was presented. First, the results were analysed on the NTREM dataset. The keyword-matching algorithm was outperformed by both Logistic regression (S-BERT SL) and K-means clustering (TF-IDF 9). The logistic regression (S-BERT SL) obtained a large generalisation error of 8%. This was attributed to a lack of stability in the optimisations of d_n. This further indicates that feature

selection/reduction and its optimisation should be included as part of cross-validation. K-means clustering (TF-IDF 9) was the best-performing method obtaining $A = 0.69$. The performance of the examined classifiers is suboptimal. There is a notable drop in the performance of the classifiers in this chapter compared to the results in Chapter 8. There is a range of plausible reasons, such as the supervised results in Section 8.3 exhibiting severe overfitting that was not picked up by the nested cross-validation. The decrease in performance of the clustering algorithms may be due to a decrease in the number of documents on which the algorithm is fitted. These reasons do not explain why the performance of the keyword-matching algorithm decreased. In Chapter 8, the keyword-matching algorithm obtained ROC AUC = 0.73, which contrasts with the ROC AUC = 0.62 obtained in this study. This could be because keywords are a good indicator of TRL for some but not all journal articles, so the algorithm's performance dropped on a larger dataset. As suggested in Chapter 8, one method of improving this algorithm would be to expand the keywords list. Alternatively, given the process of assigning TRL is subjective and the labels are human-generated, the labels generated for this dataset may be inconsistent or erroneous.

The flexibility of the logistic regression (S-BERT SL), K-means clustering (TF-IDF 9) and keyword-matching classifiers was then analysed by evaluating the performance on a dataset of HEMCE articles (a similar journal in the domain of energetic materials). The performance of all the algorithms dropped. Despite the class imbalance being less pronounced for the HEMCE dataset, it is found to significantly impact the performance of both the logistic regression (S-BERT SL) and K-means clustering (TF-IDF 9) classifiers. The synthetic-labelling approach used for the Logistic regression classifier is based upon the keyword-matching algorithm. Since the performance of the keyword matching algorithm dropped, it is unsurprising the performance of the logistic regression (S-BERT SL) classifier dropped. Overall the K-means clustering (TF-IDF 9) algorithm performed the best on the HEMCE dataset. This result shows ML approaches can yield better results even in small-data environments than standard statistical algorithms.

In Section 9.3.4, the reliability of the data used to train the algorithms was examined. Inter-participant agreement was computed using $\hat{\kappa}_G$ and κ. Analysis showed that for data with a class imbalance, the second kappa paradox is satisfied and using κ can lead to under estimated agreement levels. In such situations, the paradox-resistant $\hat{\kappa}_G$ should be used for analysis. The inter-participant agreement was fair to moderate. Agreement across all participants was

moderate. Given TRL is normally on a 9-point scale, and for this study a two-point scale was used, this level of agreement is lower than expected. This could be because the definition of TRL was not specific enough or because TRL assignment is inherently subjective. Alternatively, this could be because several articles were on the boundary between High-TRL and Low-TRL. Feedback from the participants was that some articles were more straightforward to assign a TRL to than others, where it was more subjective. A recommendation for future work is to investigate this further by asking participants to give a 1-5 confidence score in their assignment of TRL. This would allow documents that are difficult to classify to be identified and enable the uncertainty in the participant's assignment of a label to be measured. The reasons why specific documents are hard to classify should also be elicited using a similar format to the interview conducted in Chapter 5. It is noted the results here must be considered in the context of the study's limitations, namely the small sample size of participants and the variation in recruitment methods (some participants were paid).

The results of the test-retest study showed participant B had a moderate agreement level with their classification of document TRL. Whilst this is a low sample size, this result indicates a lack of consistency in their assignment of TRL. There are many reasons, such as the expert's understanding of TRL evolving with time. However, this lack of consistency in the data may ultimately explain the poor performance of the algorithms in this section. It is therefore recommended that future studies using human-generated labels for subjective concepts conduct test-retest studies to check the consistency in the produced dataset. In addition, future work should also conduct test-retest studies for several participants.

Overall, the work in this chapter has refined the methodology introduced in Chapter 8 for classification of journal articles in the energetic-materials domain by TRL. The importance of evaluating the reliability of data used to train a ML model, and determining the most appropriate metric to do this with, is highlighted by this work.

Chapter 10

Conclusion

This chapter will summarise the key findings in relation to the aims of this thesis in Section 10.1, the impact of the research conducted in this thesis in Section 10.2, and future work in Section 10.3.

10.1 Research summary

The underlying theme of this thesis was the limitations of NLP tools in small-data environments in the energetic-materials domain. The aims of this thesis were to identify the challenges of using NLP tools to develop an expert system in the energetic-materials domain and examine the interface between using NLP tools to supplement or replace human analysis.

In Chapters 2 and 3, a two-chapter literature review identified the challenges of using NLP tools to develop an expert system in the energetic-materials domain. The novel work in these chapters is the first of its kind in the energetic-materials domain. Ontologies were identified as a popular knowledge-representation technique widely implemented in many other scientific domains. The absence of domain resources, such as pretrained language models, and labelled datasets were identified as significant barriers for ontology construction. The most fundamental barrier to the successful implementation of an ontology learning system was identified as the preprocessing and extraction of text from text sources. In Chapter 7, this limitation was addressed through the development of a novel general-purpose preprocessing pipeline for the energetic-materials

domain. The development of this pipeline will facilitate the adoption of NLP in the energetic-materials domain and other related fields. The effectiveness of the pipeline was demonstrated in Chapter 8 by its ability to classify long text.

In Chapters 4, 5 and 6, examination of the interface between using NLP tools in data-limited environments to either supplement or replace human analysis was conducted for the subjective concept of 'importance'. Chapter 5 presented the first open-source investigation into the elicitation of knowledge in the domain of energetic materials. The studies conducted in Chapters 4 and 6, comparing the ability of NLP and experts to identify important points in a scientific text, are the first in the domain of energetic materials. Chapter 6 presented a novel methodology allowing for direct comparison of the ability of NLP tools and experts to identify important points in the text. The overall conclusions of these studies are that, in a data-limited environment, the NLP algorithms used here cannot effectively identify important points in a document or generate a summary akin to the way an expert can. There was considerable variation between the participants in even simple tasks such as identifying important points in a single technical document. Agreement between the NLP, expert and participants tended to be on general or conclusion-style statements only. However, as a group, the participants agreed with the expert. It was further concluded that the participants tended to agree with each other less on the reasons why a particular statement is important than they do on how important a statement is. The research conducted in these studies indicated there is a link between background factors and agreement among participants.

Chapters 8 and 9 addressed the limitations of using NLP tools in data-limited environments by developing a methodology for the classification of journal articles in the energetic-materials domain by TRL. The work in these chapters presents the first ever study into NLP-based document classification by TRL. Challenges with using real-world data, such as class imbalances, were addressed and approaches to overcome these challenges were investigated. Chapter 9 also examined the interface between using NLP tools in data-limited environments to supplement or replace human analysis by presenting a methodology to evaluate the reliability of human annotations. The conclusion of this analysis was there was a lack of agreement and consistency on expert evaluation of document TRL.

A summary of the key findings of this thesis are:

- Ontologies were identified as a popular knowledge-representation technique widely implemented in many other scientific domains.

- Barriers for ontology construction in the energetic-materials domain include an absence of domain-specific resources, extracting valuable information from diagrams, figures and equations, and preprocessing of text.

- Extractive-summarisation algorithms cannot generate summaries akin to an expert. The NLP generated summaries in this thesis lifted sentences directly from the text and could not capture complex concepts described over several sentences. They often lacked cohesion and flow.

- In a data-limited environment, extractive-summarisation algorithms cannot effectively identify important points in a domain-specific document in the same way an expert can. There was no statistically significant difference in how important the participants as a group found sentences the NLP identified and did not identify as important. However, every participant identified at least one of the same sentences as the NLP. This was due to the NLP returning conclusion-style sentences, which tended to be identified as important by the participants.

- The participants found conceptual points that the expert/NLP found to be important as more important than those they did not. Further analysis found that the majority of these points were either very general statements on the topic of the text or summary-style sentences that state the conclusions of the text.

- On average, the readers exhibited higher conceptual agreement with the expert than with the NLP/other readers.

- Transfer-learning techniques will not enhance the performance of an algorithm in all instances. In small-data environments for niche domains, traditional feature-representation methods such as TF-IDF can sometimes outperform state-of-the-art pretrained language models.

- Experts do not tend to agree on what is important in a domain-specific document, there was considerable variation in the specific points that experts found important in the text.

- As a group, the participants agreed with the expert on how important and the reason why statements about the text are important. The participants tended to agree with each other less on the reasons why a particular statement is important than they do on how important a statement is. Analysis determined there is a link between background factors and agreement between participants.

- Results indicate novices have varied approaches to evaluating the importance of technical information. One observed approach was to conduct analysis on the basis of surface features, this yielded poor agreement with the expert. Another observed approach was to use semantic clues in the text to identify which specific sentences in the text are important. The novice that used this approach, free from their own biases on the topic, obtained the highest agreement with the expert over all other participants on identification and rating of important points. However, this novice could not identify why a specific point is important akin to an expert, as this requires a complex content schema that a novice will not possess. These results imply there are specific tasks where it may be more productive to consult a group of bias-free novices than a group of experts.

- Results indicate there is a link between background factors such as: the number of years of experience in academic, commercial and government research; total number of years experience and highest education qualification obtained, with the level of agreement between participants on analysis of text by importance.

- Core reasons why the expert found specific aspects of the text were elicited. These reasons are: theoretical importance, domain overview, demonstrated use in an application, practicality of methodology, results show clear quantification of effects, advancement of the field and novelty.

- Future studies wishing to replicate the methodology in Chapter 6 should make the following amendments to the presented methodology as to avoid the limitations of the work presented here: ask specific questions about the reader's knowledge of and experience in energetic materials as a discipline in addition to their general experience. In addition it is recommended for a future study to recruit:

  - A larger group of participants.

  - Participants with experience in government policy as to get a full picture of how individuals in defence and security-related fields interpret importance.
  - A larger group of novices.

- The presence of chemical names, equations and diagrams in text in the energetic-materials domain creates extra complications for preprocessing steps such as text extraction, tokenisation and normalisation. Pre-existing tools such as GROBID and Chemdataextractor are proven to be useful for developing a preprocessing pipeline in the aforementioned domain.

- Nested cross-validation is a useful technique for model-validation in small-data environments. This method generates an estimate of unbiased generalization performance of a classification methodology. Processes such as PCA and TF-IDF fitting should be included as part of the cross-validation.

- Supervised and unsupervised ML techniques can outperform standard statistical approaches even in data-limited environments.

- Techniques such as synthetic labelling and adjusting class weightings can be useful for small datasets with a class imbalance.

- The reliability of human annotations should be checked before training an algorithm using inter-participant agreement and test-retest agreement.

- When calculating agreement, checks for prevalence and bias should be performed to ascertain if the correct metric is used. Steps should be taken to avoid the limitations of the data-generation method used in this study - low sample size and inconsistency in recruitment methods (some participants were paid).

10.2 Research impact

In the era of big data, harnessing the power of NLP to obtain actionable insights from text is a growing area of research that is already transforming other scientific domains. There has been limited work on the use of NLP in the energetic-materials domain. In addition,

there has been limited exploration of the performance of state-of-the-art techniques in data-limited environments. The work done in this thesis can facilitate the use of NLP techniques in the energetic-materials domain and other data-limited domains. For instance, the novel preprocessing pipeline developed in Chapter 7 can be adapted by other NLP practitioners to conduct NLP analysis on related domains. Techniques for downstream NLP analysis tasks in data-limited environments have been detailed in Chapters 4, 8 and 9. Deployment of these techniques in the field of Energetic Materials will enable analysts to identify and summarise relevant information efficiently, facilitating more effective analysis. These techniques can be adapted for other data-limited environments.

This thesis identified loss-of-knowledge due to the retirement of key personnel as a critical problem in defence and security-related fields such as energetic materials. Expert systems and other knowledge-representation methods may be a solution to this problem. Chapters 2 and 3 conducted a comprehensive overview of ontology learning from text, focusing on tools that are of use to small datasets and a first-of-its-kind review into ontology construction in the domain of energetic materials. Identifying the challenges that ontology construction in the energetic-materials domain pose through this work will help to facilitate the development of such a system. Developing an ontology for the energetic materials domain will facilitate knowledge sharing by encouraging interoperability and the standardisation of domain-specific terms.

It was identified that in defence and security-related fields, NLP tools have been used to supplement human analysis instead of replacing human analysis. The methodologies for comparing NLP tools and expert analysis of text presented in Chapters 4 and 6 can be adapted for other similar studies. The work in these studies identified the importance of evaluating the reliability of the data used to train algorithms and proposed a methodology. This is useful for ML practitioners that use human-annotated datasets. The results indicated NLP algorithms could not replicate expert interpretation of the text in data-limited environments. A lack of agreement among the experts was also observed even for simple concepts such as importance.

The lack of agreement observed in the aforementioned studies leads to two fundamental points. Firstly, as pointed out in Ref. [375] policymakers should rethink how they view expert opinions and apply the same critical analysis that must be applied to data. Secondly, if NLP tools are to be used to replace expert analysis eventually, then the following questions need to be considered: whose analysis are we trying to replace? Moreover, what data should we be using to train our

algorithms? Currently, using group consensus among experts is a common way of generating labels for training algorithms but, as demonstrated in this study, experts tend to agree only on very basic statements. Moreover, since algorithms learn from the examples we give, using group consensus to replicate expert opinion severely limits the algorithm's usefulness.

10.3 Future work

Throughout this thesis suggestions for further work have been made. These suggestions are summarised here

- Explore alternative methods of modifying the TextRank algorithm. For instance, using the S-BERT architecture [301] and other methods of generating sentence embeddings from Word2Vec models, such as using TF-ISF weightings in the averaging process.
- Expansion of the capabilities of the preprocessing pipeline described in Chapter 7, e.g. using more advanced chemical entity detection methods; extraction of text from other text formats; developing hand-curated dictionaries to improve the pipeline's handling of abbreviations; and implementation of chemistry-aware spell checkers.
- Exploration of finetuning approaches such as using ULMFit and long-former for long-text classification by TRL.
- Understanding why experts disagree on TRL assignment. Ask a group of experts to assign a 1-5 confidence score in their assignment of TRL. This would allow documents that are difficult to classify to be identified and enable the uncertainty in the participant's assignment of a label to be measured. The reasons why specific documents are hard to classify should also be elicited using a similar format to the interview conducted in Chapter 5. Test-retest studies should also be conducted for a limited subset of participants.

This thesis examined the performance of using NLP tools in data-limited environments. In recent years the use of generative techniques to develop synthetic data sets has emerged as a potential solution for a lack of training data in data-limited fields [508, 509]. The use of generative techniques in this context was not considered in this thesis, however, there has been a recent surge in the use and performance of generative techniques for NLP [510, 511].

As a result, it is recommended that future studies should investigate the possibility of using generative techniques to develop synthetic data sets for downstream NLP tasks.

In this thesis, the question of what data should be used to train our algorithms if experts do not agree was raised. As pointed out previously, if only data points which experts agree on are used, then the usefulness of the algorithm is severely limited. In the literature, when disagreements between labels are observed, ground-truth labels can be generated using methods such as majority voting [512], or adjudicating by an 'expert' [513, 514]. However, sometimes, particularly when labels are being assigned for subjective concepts, there is not a single correct answer. In addition, using a ground-truth label will neglect the different interpretations of concepts and lead to inconsistent labels [514]. Recently, a limited number of studies have used annotator disagreement in model training to obtain better estimates in the uncertainty of model predictions [515, 514, 516, 517]. These approaches have again been developed for large-labelled datasets. A recommendation for future work is to explore the limitations of such techniques in a data-limited environment such as the energetic-materials domain.

Bibliography

[1] Sinead O'Brien, William G Proud, and Margaret A Wilson. Elicitation of knowledge from a defence expert. In *Journal of Physics: Conference Series*, volume 1507, page 102027. IOP Publishing, 2020.

[2] Salvatore Parise, Rob Cross, and Thomas H Davenport. Strategies for preventing a knowledge-loss crisis. *MIT Sloan Management Review*, 47(4):31, 2006.

[3] Thomas J Calo. Talent management in the era of the aging workforce: The critical role of knowledge transfer. *Public Personnel Management*, 37(4):403–416, 2008.

[4] Sarah Harper, James Banks, Paul Boyle, Tom Kirkwood, and Martin Knapp. Future of an ageing population. Government office for science report, UK Government Office for Science, London, 2016.

[5] Moria Levy. Knowledge retention: minimizing organizational business loss. *Journal of Knowledge Management*, 15(4):582–600, 2011.

[6] Mark Lavelle and Margaret Wilson. Exploring human and machine-led natural language capabilities across technical disciplines. *Imperial College London*, 2018.

[7] Marcel J Rijckaert, V Debroey, and W Bogaerts. Expert systems: The state of the art. In *Mathematical models for decision support*, pages 487–517. Springer, 1988.

[8] Bruce G Buchanan. Expert systems: working systems and the research literature. *Expert systems*, 3(1):32–50, 1986.

[9] Jean-Marc David, Jean-Paul Krivine, and Reid Simmons. Second generation expert systems: A step forward in knowledge engineering. In *Second generation expert systems*, pages 3–23. Springer, 1993.

[10] Frederick Hayes-Roth, Donald A Waterman, and Douglas B Lenat. Building expert system. 1983.

[11] William P Wagner. Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies. *Expert systems with applications*, 76:85–96, 2017.

[12] Peter Gordon Roetzel. Information overload in the information age: a review of the literature from business administration, business psychology, and related disciplines with a bibliometric approach and framework development. *Business research*, 12(2):479–522, 2019.

[13] Alexander Babuta, Marion Oswald, and Ardi Janjeva. Artificial intelligence and uk national security: policy considerations. 2020.

[14] Ahmed Abbasi, Suprateek Sarker, and Roger HL Chiang. Big data research in information systems: Toward an inclusive research agenda. *Journal of the Association for Information Systems*, 17(2):I, 2016.

[15] Paulo Cortez, Sérgio Moro, Paulo Rita, David King, and Jon Hall. Insights from a text mining survey on expert systems research from 2000 to 2016. *Expert Systems*, 35(3):e12280, 2018.

[16] KR1442 Chowdhary. Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649, 2020.

[17] Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174, 2016.

[18] Fusataka Kuniyoshi, Jun Ozawa, and Makoto Miwa. Analyzing research trends in inorganic materials literature using nlp. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 319–334. Springer, 2021.

[19] Dmitry Tworowski, Alessandro Gorohovski, Sumit Mukherjee, Gon Carmi, Eliad Levy, Rajesh Detroja, Sunanda Biswas Mukherjee, and Milana Frenkel-Morgenstern. Covid19 drug repository: text-mining the literature in search of putative covid19 therapeutics. *Nucleic acids research*, 49(D1):D1113–D1121, 2021.

[20] Daniel W Otter, Julian R Medina, and Jugal K Kalita. A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):604–624, 2020.

[21] Sendong Zhao, Chang Su, Zhiyong Lu, and Fei Wang. Recent advances in biomedical literature mining. *Briefings in Bioinformatics*, 22(3):bbaa057, 2021.

[22] Berry De Bruijn and Joel Martin. Getting to the (c) ore of knowledge: mining biomedical literature. *International journal of medical informatics*, 67(1-3):7–18, 2002.

[23] Takeshi Sekimizu, Hyun S Park, and Jun'ichi Tsujii. Identifying the interaction between genes and gene products based on frequently seen verbs in medline abstracts. *Genome informatics*, 9:62–71, 1998.

[24] Wahiba Ben Abdessalem Karaa, Eman H Alkhammash, and Aida Bchir. Drug disease relation extraction from biomedical literature using nlp and machine learning. *Mobile Information Systems*, 2021, 2021.

[25] Marc Weeber, Henny Klein, Alan R Aronson, James G Mork, LT De Jong-van Den Berg, and Rein Vos. Text-based discovery in biomedicine: the architecture of the dad-system. In *Proceedings of the AMIA Symposium*, page 903. American Medical Informatics Association, 2000.

[26] H. Warrell. Nato allies need to speed up ai defence co-operation. *Financial Times*, 2021.

[27] Injy Sarhan and Marco Spruit. Can we survive without labelled data in nlp? transfer learning for open information extraction. *Applied Sciences*, 10(17):5758, 2020.

[28] Peter Schirmer, Amber Jaycocks, Sean Mann, William Marcellino, Luke J Matthews, John Parsons, and David Schulker. *Natural Language Processing: Security-and Defense-related Lessons Learned*. RAND, 2021.

[29] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *AI magazine*, 14(1):17–17, 1993.

[30] Rinke Hoekstra. *Ontology Representation: design patterns and ontologies that make sense*, volume 197. IOS Press, 2009.

[31] Hector J Levesque. Knowledge representation and reasoning. *Annual review of computer science*, 1(1):255–287, 1986.

[32] Lexico dictionaries — english. https://www.lexico.com/definition/artificial_intelligence, Accessed on 1st May, 2020.

[33] John Durkin. 4 expert system. *The Handbook of Applied Expert Systems*, 4:15, 1997.

[34] Rajendra Akerkar and Priti Sajja. *Knowledge-based systems*. Jones & Bartlett Publishers, 2009.

[35] B Buchanan, Georgia Sutherland, and EA Feigenbaum. Heuristic dendral: A program for generating explanatory hypotheses. *Organic Chemistry*, 1969.

[36] William Van Melle. Mycin: a knowledge-based consultation program for infectious disease diagnosis. *International journal of man-machine studies*, 10(3):313–322, 1978.

[37] Marc A Tischler and Edward A Fox. An expert system for selecting liquid chromatographic separation methods. *Computers & chemistry*, 11(4):235–240, 1987.

[38] Bob J Wielinga, A Th Schreiber, and Jost A Breuker. Kads: A modelling approach to knowledge engineering. *Knowledge acquisition*, 4(1):5–53, 1992.

[39] Alison Kidd. *Knowledge acquisition for expert systems: A practical handbook*. Springer Science & Business Media, 2012.

[40] Lucas Drumond and Rosario Girardi. A survey of ontology learning procedures. *WONTO*, 427:1–13, 2008.

[41] Joost Breuker. A cognitive science perspective on knowledge acquisition. *International Journal of Human Computer Studies*, 71(2):177–183, 2013.

[42] Nigel Shadbolt and A Mike Burton. The empirical study of knowledge elicitation techniques. *ACM SIGART Bulletin*, (108):15–18, 1989.

[43] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

[44] Andrew Thomas Bimba, Norisma Idris, Ahmed Al-Hunaiyyan, Rohana Binti Mahmud, Ahmed Abdelaziz, Suleman Khan, and Victor Chang. Towards knowledge modeling and

manipulation technologies: A survey. *International Journal of Information Management*, 36(6):857–871, 2016.

[45] Carlos Ramirez and Benjamin Valdes. A general knowledge representation model of concepts. *Advances in knowledge representation*, page 272, 2012.

[46] Paul Harmon. *Business process change: a business process management guide for managers and process professionals*. Morgan Kaufmann, 2019.

[47] Stephen I Gallant. Connectionist expert systems. *Communications of the ACM*, 31(2):152–169, 1988.

[48] William J Clancey. The knowledge level reinterpreted: Modeling socio-technical systems. *International journal of intelligent systems*, 8(1):33–49, 1993.

[49] C Welty, F Lehmann, G Gruninger, and M Uschold. Ontology: Expert systems all over again. In *Invited panel at AAAI-99: The National Conference on Artificial Intelligence. Austin, Texas*, 1999.

[50] NF Nabila, Nulida Basir, and Mustafa Mat Deris. Non-taxonomic relation extraction using probability theory. In *The World Congress on Engineering and Computer Science*, pages 287–301. Springer, 2017.

[51] Robert J Rovetto, TS Kelso, and Daniel A O'Neil. Orbital debris ontology, terminology, and knowledge modeling. 2019.

[52] Alan Jovic, Marin Prcela, and Dragan Gamberger. Ontologies in medical knowledge representation. In *2007 29th International Conference on Information Technology Interfaces*, pages 535–540. IEEE, 2007.

[53] Jonathan BL Bard and Seung Y Rhee. Ontologies in biology: design, applications and future challenges. *Nature Reviews Genetics*, 5(3):213–222, 2004.

[54] Janna Hastings, Despoina Magka, Colin Batchelor, Lian Duan, Robert Stevens, Marcus Ennis, and Christoph Steinbeck. Structure-based classification and ontology in chemistry. *Journal of cheminformatics*, 4(1):8, 2012.

[55] Ayesha Ameen, Khaleel Ur Rahman Khan, and B Padmaja Rani. Creation of ontology in education domain. In *2012 IEEE Fourth International Conference on Technology for Education*, pages 237–238. IEEE, 2012.

[56] Mirjana Ivanović and Zoran Budimac. An overview of ontologies and data resources in medical domains. *Expert Systems with Applications*, 41(11):5158–5166, 2014.

[57] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl_1):D258–D261, 2004.

[58] Suzanna E Lewis. The vision and challenges of the gene ontology. *The Gene Ontology Handbook*, page 291.

[59] Luis M Sarro and Rafael Martínez. First steps towards an ontology for astrophysics. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 1389–1395. Springer, 2003.

[60] Miguel Francisco Miravite Remolona. *HOLMES: A Hybrid Ontology-Learning Materials Engineering System*. PhD thesis, Columbia University, 2018.

[61] Robert J Rovetto. An ontological architecture for orbital debris data. *Earth Science Informatics*, 9(1):67–82, 2016.

[62] John F Sowa. Ontology, metadata, and semiotics. In *International conference on conceptual structures*, pages 55–81. Springer, 2000.

[63] Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[64] Shu-Hsien Liao. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications*, 28(1):93–103, 2005.

[65] Allen Newell et al. The knowledge level. *Artificial intelligence*, 18(1):87–127, 1982.

[66] Nicola Guarino and Pierdaniele Giaretta. Ontologies and knowledge bases. *Towards very large knowledge bases*, pages 1–2, 1995.

[67] Deryle Lonsdale, David W Embley, Yihong Ding, Li Xu, and Martin Hepp. Reusing ontologies and language components for ontology generation. *Data & Knowledge Engineering*, 69(4):318–330, 2010.

[68] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001.

[69] Mark A Musen. Dimensions of knowledge sharing and reuse. *Computers and biomedical research*, 25(5):435–467, 1992.

[70] Thomas R Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–221, 1993.

[71] Tom Gruber. Ontology. http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, 2018.

[72] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2):72–79, 2001.

[73] W Kunierczyk. Nontological engineering. formal ontology in information systems. In *Proceedings of the 4th International Conference FOIS*, pages 39–50, 2006.

[74] Willem Nico Borst. Construction of engineering ontologies for knowledge sharing and reuse. 1999.

[75] Fabian Neuhaus. On the definition of'ontology'. In *JOWO*, 2017.

[76] Michael R Genesereth and Nils J Nilsson. Logical foundations of. *Artificial Intelligence. New York: Morgan Kaufmann Publishers*, 1987.

[77] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer, 2009.

[78] ACM Press. *Ontology: Towards a new synthesis*, volume 10, 2001.

[79] Fabian Neuhaus. On the Definition of 'Ontology'. In *JOWO*, 2017.

[80] Tom Gruber, In Ling Liu Ontology, and M Tamer Özsu. Encyclopedia of database systems. *Ontology*, 2009.

[81] Christina Feilmayr and Wolfram Wöß. An analysis of ontologies and their success factors for application to business. *Data & Knowledge Engineering*, 101:1–23, 2016.

[82] Amal Zouaq. An overview of shallow and deep natural language processing for ontology learning. In *Ontology learning and knowledge discovery using the web: Challenges and recent advances*, pages 16–37. IGI Global, 2011.

[83] Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R Swartout. Enabling technology for knowledge sharing. *AI magazine*, 12(3):36–36, 1991.

[84] Dragan Gaševic, Dragan Djuric, and Vladan Devedžic. *Model driven architecture and ontology development*. Springer Science & Business Media, 2006.

[85] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, 44(4):1–36, 2012.

[86] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientfic American*, 284(5):34—-43, 2001.

[87] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.

[88] Grimm st Stephan, Hitzler st Pascal, and Abecker st Andreas. Knowledge representation and ontologies. *Semantic Web Services: Concepts, Technologies, and Applications*, pages 51–105, 2007.

[89] Rdf. =https://www.w3.org/RDF/, Accessed on 15th of June 2020.

[90] Peter F Patel-Schneider, Patrick Hayes, Ian Horrocks, and Frank van Harmelen. Web ontology language (owl) abstract syntax and semantics. *W3C recommendation*, 2004.

[91] definition of predicate. https://www.merriam-webster.com/dictionary/predicate, Accessed 20th July 2020.

[92] Rdf triple. https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/#section-triple, Accessed 1st July 2020.

[93] W3C. RDF Schema 1.1. Technical report, 2014.

[94] JRG Pulido, MAG Ruiz, Roberto Herrera, Enrique Cabello, Steve Legrand, and Dave Elliman. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*, 19(7):489–497, 2006.

[95] Rdfs. https://www.w3.org/wiki/RDFS, Accessed 20th July 2020.

[96] Jorge Cardoso. Discovering semantic web services with and without a common ontology commitment. In *2006 IEEE Services Computing Workshops*, pages 183–190. IEEE, 2006.

[97] Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. *Handbook of knowledge representation*. Elsevier, 2008.

[98] Deborah L McGuinness and Frank van Harmelen. OWL Web Ontology Language Overview, 2004.

[99] Owl. https://www.w3.org/OWL/, Accessed on 20th of June 2020.

[100] Nicola Guarino. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*, volume 46. IOS press, 1998.

[101] Elena Beisswanger, Stefan Schulz, Holger Stenzhorn, and Udo Hahn. Biotop: An upper domain ontology for the life sciences. *Applied Ontology*, 3(4):205–212, 2008.

[102] Adam Pease and Ian Niles. Ieee standard upper ontology: a progress report. *The Knowledge Engineering Review*, 17(1):65–70, 2002.

[103] Farhad Ameri and Debasish Dutta. An upper ontology for manufacturing service description. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 42578, pages 651–661, 2006.

[104] Roberto Navigli and Paola Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151–179, 2004.

[105] Sinéad Boyce and Claus Pahl. Developing domain ontologies for course content. JSTOR, 2007.

[106] Ste en Schulze-Kremer. Ontologies for molecular biology. *Computer and Information Science*, 6(21), 2001.

[107] Lynda Temal, Pascal Lando, Bernard Gibaud, Michel Dojat, Gilles Kassel, and Anne Lapujade. Ontoneurobase: a multi-layered application ontology in neuroimaging. In *Second Workshop: Formal Ontologies Meet Industry (FOMI 2006)*, 2006.

[108] Jose LV Mejino Jr, Daniel L Rubin, and James F Brinkley. Fma-radlex: An application ontology of radiological anatomy derived from the foundational model of anatomy reference ontology. In *AMIA Annual Symposium Proceedings*, volume 2008, page 465. American Medical Informatics Association, 2008.

[109] Erick Antezana, Mikel Egaña, Ward Blondé, Aitzol Illarramendi, Iñaki Bilbao, Bernard De Baets, Robert Stevens, Vladimir Mironov, and Martin Kuiper. The cell cycle ontology: an application ontology for the representation and integrated analysis of the cell cycle process. *Genome Biology*, 10(5):R58, 2009.

[110] Dirk Marwede, Matthew Fielding, and Thomas Kahn. Radio: a prototype application ontology for radiology reporting tasks. In *AMIA Annual Symposium Proceedings*, volume 2007, page 513. American Medical Informatics Association, 2007.

[111] Michael Uschold, Michael Gruninger, et al. Ontologies: Principles, methods and applications. *TECHNICAL REPORT-UNIVERSITY OF EDINBURGH ARTIFICIAL INTELLIGENCE APPLICATIONS INSTITUTE AIAI TR*, 1996.

[112] Fatima N. Al-Aswadi, Huah Yong Chan, and Keng Hoon Gan. Automatic ontology construction from text: a review from shallow to deep learning trend. *Artificial Intelligence Review*, pages 1—-28, 2019.

[113] Roberto Navigli, Paola Velardi, and Aldo Gangemi. Ontology learning and its application to automated terminology translation. *IEEE Intelligent systems*, 18(1):22–31, 2003.

[114] Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. A survey of ontology learning techniques and applications, 2018.

[115] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. Ontology Learning from Text : An Overview. *Ontology learning from text: Methods, evaluation and applications*, 123:3—-12, 2005.

[116] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.

[117] Agnieszka Konys. Knowledge systematization for ontology learning methods. *Procedia computer science*, 126:2194–2207, 2018.

[118] Philipp Cimiano, Alexander Mädche, Steffen Staab, and Johanna Völker. Ontology learning. In *Handbook on ontologies*, pages 245–267. Springer, 2009.

[119] Zhichao Liu, Ruth A Roberts, Madhu Lal-Nag, Xi Chen, Ruili Huang, and Weida Tong. Ai-based language models powering drug discovery and development. *Drug Discovery Today*, 26(11):2593–2607, 2021.

[120] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[121] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[122] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[123] Faiza Khan Khattak, Serena Jeblee, Chloé Pou-Prom, Mohamed Abdalla, Christopher Meaney, and Frank Rudzicz. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics*, 100:100057, 2019.

[124] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[125] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[126] Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting contextual word embeddings: Architecture and representation. *arXiv preprint arXiv:1808.08949*, 2018.

[127] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021.

[128] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[129] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[130] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, pages 22–es, 2004.

[131] Christopher A Brewster. Mind the gap: Bridging from text to ontological knowledge. 2008.

[132] Merriam webster defintion of homograph. https://www.merriam-webster.com/dictionary/homograph, 2020.

[133] https://nlp.stanford.edu/ir-book/html/htmledition/tokenization-1.html, 2019.

[134] Nikita A Astrakhantsev, Denis G Fedorenko, and D Yu Turdakov. Methods for automatic term recognition in domain-specific text collections: A survey. *Programming and Computer Software*, 41(6):336–349, 2015.

[135] Michael Krauthammer and Goran Nenadic. Term identification in the biomedical literature. *Journal of biomedical informatics*, 37(6):512–526, 2004.

[136] John S Justeson and Slava M Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1(1):9–27, 1995.

[137] Sree Harissh Venu, Vignesh Mohan, Kodaikkaavirinaadan Urkalan, and TV Geetha. Unsupervised domain ontology learning from text. In *International Conference on Mining Intelligence and Knowledge Exploration*, pages 132–143. Springer, 2016.

[138] Kevin Meijer, Flavius Frasincar, and Frederik Hogenboom. A semantic approach for extracting domain taxonomies from text. *Decision Support Systems*, 62:78–93, 2014.

[139] Mariam Tariq Lee Gillam and Khurshid Ahmad. Terminology and the construction of ontology. *Application-Driven Terminology Engineering*, 2:49, 2007.

[140] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[141] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.

[142] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[143] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[144] Shahzad Qaiser and Ramsha Ali. Text mining: use of tf-idf to examine the relevance of words to documents. *International Journal of Computer Applications*, 181(1):25–29, 2018.

[145] Bo Wang, Jun Luo, and Shuyuan Zhu. Research on domain ontology automation construction based on chinese texts. In *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, pages 425–430, 2019.

[146] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Determining termhood for learning domain ontologies using domain prevalence and tendency. In *Proceedings of the sixth Australasian conference on Data mining and analytics-Volume 70*, pages 47–54. Citeseer, 2007.

[147] Paola Velardi, Michele Missikoff, and Roberto Basili. Identification of relevant terms to support the construction of domain ontologies. In *Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management*, 2001.

[148] Kiruparan Balachandran and Surangika Ranathunga. Domain-specific term extraction for concept identification in ontology construction. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 34–41. IEEE, 2016.

[149] Roberto Navigli and Paola Velardi. Semantic interpretation of terminological strings. In *Proc. 6th Int'l Conf. Terminology and Knowledge Eng*, pages 95–100, 2002.

[150] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International journal on digital libraries*, 3(2):115–130, 2000.

[151] E Milios, Y Zhang, B He, and L Dong. Automatic term extraction and document similarity in special text corpora. In *Proceedings of the sixth conference of the pacific association for computational linguistics*, pages 275–284. Citeseer, 2003.

[152] Efthymios G Drymonas. Ontology learning from text based on multiword term concepts: the ontogain method. 2009.

[153] Sunyang Fu, David Chen, Huan He, Sijia Liu, Sungrim Moon, Kevin J Peterson, Feichen Shen, Liwei Wang, Yanshan Wang, Andrew Wen, et al. Clinical concept extraction: a methodology review. *Journal of biomedical informatics*, 109:103526, 2020.

[154] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[155] Miran Seok, Hye-Jeong Song, Chan-Young Park, Jong-Dae Kim, and Yu-seop Kim. Named entity recognition using word embedding as a feature. *International Journal of Software Engineering and Its Applications*, 10(2):93–104, 2016.

[156] Jingchi Jiang, Yi Guan, and Chao Zhao. Wi-enre in clef ehealth evaluation lab 2015: Clinical named entity recognition based on crf. In *CLEF (Working Notes)*, 2015.

[157] Abdulrahman Khalifa, Sumithra Velupillai, and Stephane Meystre. Utahbmi at semeval-2016 task 12: extracting temporal information from clinical text. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1256–1262, 2016.

[158] Yaoyun Zhang, Min Jiang, Jingqi Wang, and Hua Xu. Semantic role labeling of clinical text: comparing syntactic parsers and features. In *AMIA Annual Symposium Proceedings*, volume 2016, page 1283. American Medical Informatics Association, 2016.

[159] Philipp Cimiano and Johanna Völker. A framework for ontology learning and data-driven change discovery. In *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 227–238. Springer, 2005.

[160] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[161] Donald AB Lindberg, Betsy L Humphreys, and Alexa T McCray. The unified medical language system. *Methods of information in medicine*, 32(4):281, 1993.

[162] Zellig Harris. Mathematical structures of language. *Interscience tracts in pure and applied mathematics*, 1968.

[163] Kaihong Liu, William R Hogan, and Rebecca S Crowley. Natural language processing methods and systems for biomedical ontology learning. *Journal of biomedical informatics*, 44(1):163–179, 2011.

[164] P Cimiano, A Hotho, and S Staab. Comparing conceptual, divisive and agglomerative clustering for learning taxonomies from text retrieved june 21, 2010, 2004.

[165] Krister Lindén and Jussi Piitulainen. Discovering synonyms and other related words. In *Proceedings of CompuTerm 2004: 3rd International Workshop on Computational Terminology*, pages 63–70, 2004.

[166] Yanen Li, Bo-June Paul Hsu, ChengXiang Zhai, and Kuansan Wang. Mining entity attribute synonyms via compact clustering. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 867–872, 2013.

[167] Li Zhang, Jun Li, and Chao Wang. Automatic synonym extraction using word2vec and spectral clustering. In *2017 36th Chinese Control Conference (CCC)*, pages 5629–5632. IEEE, 2017.

[168] Dongwen Zhang, Hua Xu, Zengcai Su, and Yunfeng Xu. Chinese comments sentiment classification based on word2vec and svmperf. *Expert Systems with Applications*, 42(4):1857–1863, 2015.

[169] Rawan N Al-Matham and Hend S Al-Khalifa. Synoextractor: a novel pipeline for arabic synonym extraction using word2vec word embeddings. *Complexity*, 2021, 2021.

[170] Sunghae Jun, Sang-Sung Park, and Dong-Sik Jang. Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications*, 41(7):3204–3212, 2014.

[171] S Lovelyn Rose and K Chandran. Normalized web distance based web query classification. *Journal of Computer Science*, 8(5):804–808, 2012.

[172] Christian Wartena, Rogier Brussee, and Wout Slakhorst. Keyword extraction using word co-occurrence. In *2010 Workshops on Database and Expert Systems Applications*, pages 54–58. IEEE, 2010.

[173] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[174] Xian-ming YAO, Jian-hou GAN, and XU Jian. Concept extraction based on hybrid approach combined with semantic analysis. *DEStech Transactions on Engineering and Technology Research*, (amma), 2017.

[175] Masahiro Ito, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Association thesaurus construction methods based on link co-occurrence analysis for wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 817–826, 2008.

[176] Patrick Pantel and Dekang Lin. A statistical corpus-based term extractor. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 36–46. Springer, 2001.

[177] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[178] Peter D Turney. Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *European conference on machine learning*, pages 491–502. Springer, 2001.

[179] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211, 1997.

[180] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[181] Roberto Basili, Maria Teresa Pazienza, and Paola Velardi. An empirical symbolic approach to natural language processing. *Artificial Intelligence*, 85(1-2):59–99, 1996.

[182] Marti A Hearst. Automated discovery of wordnet relations. *WordNet: an electronic lexical database*, 2, 1998.

[183] Rion Snow, Daniel Jurafsky, and Andrew Y Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. Association for Computational Linguistics, 2006.

[184] Neha Kaushik and Niladri Chatterjee. Automatic relationship extraction from agricultural text for ontology construction. *Information processing in agriculture*, 5(1):60–73, 2018.

[185] Chao Zhang, Fangbo Tao, Xiusi Chen, Jiaming Shen, Meng Jiang, Brian Sadler, Michelle Vanni, and Jiawei Han. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2701–2709, 2018.

[186] Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. A phrase mining framework for recursive construction of a topical hierarchy. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 437–445, 2013.

[187] Mahmood Neshati and Leila Sharif Hassanabadi. Taxonomy construction using compound similarity measure. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 915–932. Springer, 2007.

[188] Gilles Bisson, Claire Nédellec, and Dolores Canamero. Designing clustering methods for ontology building-the mo'k workbench. In *ECAI workshop on ontology learning*, volume 31. Citeseer, 2000.

[189] Sharon A Caraballo. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 120–126, 1999.

[190] Jeroen De Knijff, Flavius Frasincar, and Frederik Hogenboom. Domain taxonomy learning from text: The subsumption method versus hierarchical clustering. *Data & Knowledge Engineering*, 83:54–69, 2013.

[191] Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, 1999.

[192] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. Automatic taxonomy construction from keywords. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1433–1441, 2012.

[193] Luis Espinosa-Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. Supervised distributional hypernym discovery via domain adaptation. In *Conference on Empirical Methods in Natural Language Processing; 2016 Nov 1-5; Austin, TX. Red Hook (NY): ACL; 2016. p. 424-35.* ACL (Association for Computational Linguistics), 2016.

[194] Junjie Yao, Bin Cui, Gao Cong, and Yuxin Huang. Evolutionary taxonomy construction from dynamic tag space. *World Wide Web*, 15(5-6):581–602, 2012.

[195] Anh Tuan Luu, Yi Tay, Siu Cheung Hui, and See Kiong Ng. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 403–413, 2016.

[196] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. Learning term embeddings for hypernymy identification. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[197] Jing Qiu, Yuhan Chai, Yan Liu, Zhaoquan Gu, Shudong Li, and Zhihong Tian. Automatic non-taxonomic relation extraction from big data in smart city. *IEEE Access*, 6:74854–74864, 2018.

[198] Janardhana Punuru and Jianhua Chen. Learning non-taxonomical semantic relations from domain texts. *Journal of Intelligent Information Systems*, 38(1):191–207, 2012.

[199] Rim Zarrad, Narjes Doggaz, and Ezzeddine Zagrouba. Using heading hierarchy for non-taxonomic relation extraction. In *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pages 1–7. IEEE, 2016.

[200] Jorge Villaverde, Agustin Persson, Daniela Godoy, and Analía Amandi. Supporting the discovery and labeling of non-taxonomic relationships in ontology learning. *Expert Systems with Applications*, 36(7):10288–10294, 2009.

[201] Ivo Serra, Rosario Girardi, and Paulo Novais. Parnt: a statistic based approach to extract non-taxonomic relationships of ontologies from text. In *2013 10th International Conference on Information Technology: New Generations*, pages 561–566. IEEE, 2013.

[202] Mário Ribeiro. Extraction of non-taxonomic relations from texts to enrich a basic ontology. *Unpublished Master Thesis, School of Information Systems and Computer Engineering, University of Lisbon, Portugal*, 2014.

[203] Stefan Bergman. *The kernel function and conformal mapping*, volume 5. American Mathematical Soc., 1970.

[204] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106, 2003.

[205] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 423. Association for Computational Linguistics, 2004.

[206] Hong-Woo Chun, Chang-Hoo Jeong, Sa-Kwang Song, Yun-Soo Choi, Sung-Pil Choi, and Won-Kyung Sung. Relation extraction based on composite kernel combining pattern similarity of predicate-argument structure. In *International Conference on U-and E-Service, Science and Technology*, pages 269–273. Springer, 2011.

[207] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics, 2005.

[208] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th*

Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2, pages 1003–1011. Association for Computational Linguistics, 2009.

[209] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[210] Limin Yao, Sebastian Riedel, and Andrew McCallum. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023. Association for Computational Linguistics, 2010.

[211] Ryan McDonald, Fernando Pereira, Seth Kulick, Scott Winters, Yang Jin, and Pete White. Simple algorithms for complex relation extraction with applications to biomedical ie. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 491–498. Association for Computational Linguistics, 2005.

[212] Sergey Brin. Extracting patterns and relations from the world wide web. In *International Workshop on The World Wide Web and Databases*, pages 172–183. Springer, 1998.

[213] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94, 2000.

[214] Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, 2007.

[215] Ana Oliveira, F Câmara Pereira, and Amílcar Cardoso. Automatic reading and learning from text. In *Proceedings of the international symposium on artificial intelligence (ISAI)*. sn, 2001.

[216] Dekang Lin and Patrick Pantel. Dirt@ sbt@ discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328, 2001.

[217] Elias Zavitsanos, George Paliouras, and George A Vouros. Gold standard evaluation of ontology learning methods through ontology transformation and alignment. *IEEE Transactions on knowledge and data engineering*, 23(11):1635–1648, 2010.

[218] Hlomani Hlomani and Deborah Stacey. Approaches, methods, metrics, measures, and subjectivity in ontology evaluation: A survey. *Semantic Web Journal*, 1(5):1–11, 2014.

[219] Joe Raad and Christophe Cruz. A survey on ontology evaluation methods. 2015.

[220] Janez Brank, Marko Grobelnik, and Dunja Mladenic. A survey of ontology evaluation techniques. In *Proceedings of the conference on data mining and data warehouses (SiKDD 2005)*, pages 166–170. Citeseer Ljubljana, Slovenia, 2005.

[221] Alexander Maedche and Steffen Staab. Measuring similarity between ontologies. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 251–263. Springer, 2002.

[222] Robert Porzel and Rainer Malaka. A task-based approach for ontology evaluation. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*, pages 1–6. Citeseer, 2004.

[223] Christopher Brewster, Harith Alani, Srinandan Dasmahapatra, and Yorick Wilks. Data driven ontology evaluation. 2004.

[224] Adolfo Lozano-Tello and Asunción Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management (JDM)*, 15(2):1–18, 2004.

[225] Jinie Pak and Lina Zhou. A framework for ontology evaluation. In *Workshop on E-Business*, pages 10–18. Springer, 2009.

[226] Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jos Lehmann. Modelling ontology evaluation and validation. In *European Semantic Web Conference*, pages 140–154. Springer, 2006.

[227] Denny Vrandečić. Ontology evaluation. In *Handbook on ontologies*, pages 293–313. Springer, 2009.

[228] Peter Haase and York Sure. Usage tracking for ontology evolution. *EU IST Project SEKT Deliverable D*, 3.

[229] Chintan Patel, Kaustubh Supekar, Yugyung Lee, and EK Park. Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In *Proceedings of the 5th ACM international workshop on Web information and data management*, pages 58–61, 2003.

[230] David Faure, Claire Nédellec, and Céline Rouveirol. Acquisition of semantic knowledge using machine learning methods: The system" asium". In *Universite Paris Sud*. Citeseer, 1998.

[231] David Faure and Thierry Poibeau. First experiments of using semantic knowledge learned by asium for information extraction task using intex. Citeseer.

[232] Christiane Fellbaum. Wordnet. *The encyclopedia of applied linguistics*, 2012.

[233] Mehrnoush Shamsfard and Ahmad Abdollahzadeh Barforoush. Learning ontologies from natural language texts. *International journal of human-computer studies*, 60(1):17–63, 2004.

[234] Philipp Cimiano, Andreas Hotho, and Steffen Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of artificial intelligence research*, 24:305–339, 2005.

[235] Lillian Lee. Measures of distributional similarity. *arXiv preprint cs/0001012*, 2000.

[236] Xing Jiang and Ah-Hwee Tan. Crctol: A semantic-based domain ontology learning system. *Journal of the American Society for Information Science and Technology*, 61(1):150–168, 2010.

[237] Satanjeev Banerjee and Ted Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *International conference on intelligent text processing and computational linguistics*, pages 136–145. Springer, 2002.

[238] Euthymios Drymonas, Kalliopi Zervanou, and Euripides GM Petrakis. Unsupervised ontology acquisition from plain texts: the ontogain system. In *International Conference on Application of Natural Language to Information Systems*, pages 277–287. Springer, 2010.

[239] Daniel C Elton, Dhruv Turakhia, Nischal Reddy, Zois Boukouvalas, Mark D Fuge, Ruth M Doherty, and Peter W Chung. Using natural language processing techniques to extract information on the properties and functionalities of energetic materials from large text corpora. *arXiv preprint arXiv:1903.00415*, 2019.

[240] Masakazu Suzuki, Fumikazu Tamari, Ryoji Fukuda, Seiichi Uchida, and Toshihiro Kanahori. Infty: an integrated ocr system for mathematical documents. In *Proceedings of the 2003 ACM symposium on Document engineering*, pages 95–104, 2003.

[241] Igor V Filippov and Marc C Nicklaus. Optical structure recognition software to recover chemical information: Osra, an open source solution, 2009.

[242] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[243] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[244] J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182, 2003.

[245] Takaki Makino, Yoshihiro Ohta, Jun'ichi Tsujii, et al. Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain*, pages 1–8, 2002.

[246] Patrice Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pages 473–474. Springer, 2009.

[247] Matthew C Swain and Jacqueline M Cole. Chemdataextractor: a toolkit for automated extraction of chemical information from the scientific literature. *Journal of chemical information and modeling*, 56(10):1894–1904, 2016.

[248] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In *Mining text data*, pages 43–76. Springer, 2012.

[249] Milan Straka and Jana Straková. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 shared task: Multilingual Parsing from raw text to universal dependencies*, pages 88–99, 2017.

[250] Steven P Abney. Parsing by chunks. In *Principle-based parsing*, pages 257–278. Springer, 1991.

[251] Jasmeet Singh and Vishal Gupta. A systematic review of text stemming techniques. *Artificial Intelligence Review*, 48(2):157–217, 2017.

[252] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

[253] SA Babar and Pallavi D Patil. Improving performance of text summarization. *Procedia Computer Science*, 46:354–363, 2015.

[254] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66, 2017.

[255] Mahmood Yousefi-Azar and Len Hamey. Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68:93–105, 2017.

[256] Cristian Moral, Angélica de Antonio, Ricardo Imbert, and Jaime Ramírez. A survey of stemming algorithms in information retrieval. *Information Research: An International Electronic Journal*, 19(1):n1, 2014.

[257] Liwei Hou, Po Hu, and Chao Bei. Abstractive document summarization via neural model with joint attention. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 329–338. Springer, 2017.

[258] Rui Sun, Zhenchao Wang, Yafeng Ren, and Donghong Ji. Query-biased multi-document abstractive summarization via submodular maximization using event guidance. In *International Conference on Web-Age Information Management*, pages 310–322. Springer, 2016.

[259] Shuai Wang, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. Integrating extractive and abstractive models for long text summarization. In *2017 IEEE International Congress on Big Data (BigData Congress)*, pages 305–312. IEEE, 2017.

[260] Som Gupta and Sanjai Kumar Gupta. Abstractive summarization: An overview of the state of the art. *Expert Systems with Applications*, 121:49–65, 2019.

[261] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[262] Pegasus hugging face transformers. https://huggingface.co/docs/transformers/model_doc/pegasus. Accessed: 2010-09-30.

[263] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[264] Nenkova. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, 101, 2005.

[265] N Moratanch and S Chitrakala. A survey on extractive text summarization. In *2017 international conference on computer, communication and signal processing (ICCCSP)*, pages 1–6. IEEE, 2017.

[266] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[267] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 552–559, 2007.

[268] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[269] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[270] Juan-Manuel Torres-Moreno. *Automatic text summarization*. John Wiley & Sons, 2014.

[271] Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, and Apurba Sarkar. Graph-based text summarization using modified textrank. In *Soft computing in data analytics*, pages 137–146. Springer, 2019.

[272] Federico Barrios, Federico López, Luis Argerich, and Rosa Wachenchauzer. Variations of the similarity function of textrank for automated summarization. *arXiv preprint arXiv:1602.03606*, 2016.

[273] Kang Yang, Kamal Al-Sabahi, Yanmin Xiang, and Zuping Zhang. An integrated graph model for document summarization. *Information*, 9(9):232, 2018.

[274] Daraksha Parveen and Michael Strube. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[275] Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, and Ilyas Cicekli. Text summarization using latent semantic analysis. *Journal of Information Science*, 37(4):405–417, 2011.

[276] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[277] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3), 2010.

[278] Susan T Dumais. Improving the retrieval of information from external sources. *Behavior research methods, instruments, & computers*, 23(2):229–236, 1991.

[279] Makbule Gulcin Ozsoy, Ilyas Cicekli, and Ferda Nur Alpaslan. Text summarization of turkish texts using latent semantic analysis. In *Proceedings of the 23rd international conference on computational linguistics*, pages 869–876. Association for Computational Linguistics, 2010.

[280] Preslav Nakov, Antonia Popova, and Plamen Mateev. Weight functions impact on lsa performance. *EuroConference RANLP*, pages 187–193, 2001.

[281] IV Mashechkin, MI Petrovskiy, DS Popov, and Dmitry V Tsarev. Automatic text summarization using latent semantic analysis. *Programming and Computer Software*, 37(6):299–305, 2011.

[282] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng. Text summarization using a trainable summarizer and latent semantic analysis. *Information processing & management*, 41(1):75–95, 2005.

[283] Josef Steinberger and Karel Ježek. Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275, 2012.

[284] Ahmed El-Refaiy, A R Abas, and I Elhenawy. Review of recent techniques for extractive text summarization. *Journal of Theoretical and Applied Information Technology*, 96:7739–7759, 2018.

[285] Yihong Gong and Xin Liu. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–25. ACM, 2001.

[286] Josef Steinberger and Karel Jezek. Using latent semantic analysis in text summarization and summary evaluation. *Proc. ISIM*, 4:93–100, 2004.

[287] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *Proceedings of the 15th international conference on World Wide Web*, pages 953–954, 2006.

[288] Kamal Al-Sabahi, Zhang Zuping, and Yang Kang. Latent semantic analysis approach for document summarization based on word embeddings. *arXiv preprint arXiv:1807.02748*, 2018.

[289] Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. Recent advances in document summarization. *Knowledge and Information Systems*, 53(2):297–336, 2017.

[290] Narendra Andhale and LA Bewoor. An overview of text summarization techniques. In *2016 international conference on computing communication control and automation (ICCUBEA)*, pages 1–7. IEEE, 2016.

[291] Pierre Lison. An introduction to machine learning, 2015.

[292] You Ouyang, Wenjie Li, Renxian Zhang, Sujian Li, and Qin Lu. A progressive sentence selection strategy for document summarization. *Information Processing & Management*, 49(1):213–221, 2013.

[293] Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679, 2021.

[294] Yogesh Kumar Meena and Dinesh Gopalani. Evolutionary algorithms for extractive automatic text summarization. *Procedia Computer Science*, 48:244–249, 2015.

[295] Elena Lloret and Manuel Palomar. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41, 2012.

[296] Xue Ying. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing, 2019.

[297] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.

[298] Rahim Khan, Yurong Qian, and Sajid Naeem. Extractive based text summarization using k-means and tf-idf. *International Journal of Information Engineering & Electronic Business*, 11(3), 2019.

[299] Akanksha Joshi, Eduardo Fidalgo, Enrique Alegre, and Laura Fernández-Robles. Summcoder: An unsupervised framework for extractive text summarization based on deep auto-encoders. *Expert Systems with Applications*, 129:200–215, 2019.

[300] Dima Suleiman and Arafat A Awajan. Deep learning based extractive text summarization: approaches, datasets and evaluation measures. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 204–210. IEEE, 2019.

[301] Derek Miller. Leveraging bert for extractive text summarization on lectures. *arXiv preprint arXiv:1906.04165*, 2019.

[302] Anil K Jain, Richard C Dubes, et al. *Algorithms for clustering data*, volume 6. Prentice hall Englewood Cliffs, 1988.

[303] Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

[304] Aakanksha Sharaff, Hari Shrawgi, Priyank Arora, and Anshul Verma. Document summarization by agglomerative nested clustering approach. In *2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT)*, pages 187–191. IEEE, 2016.

[305] Ayman El-Kilany and Iman Saleh. Unsupervised document summarization using clusters of dependency graph nodes. In *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 557–561. IEEE, 2012.

[306] Harshal J Jain, MS Bewoor, and SH Patil. Context sensitive text summarization using k means clustering algorithm. *Int. J. Soft Comput. Eng*, 2(2):301–304, 2012.

[307] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[308] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[309] M Indu and KV Kavitha. Review on text summarization evaluation methods. In *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, pages 1–4. IEEE, 2016.

[310] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[311] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[312] https://docs.python.org/2/library/string.html, 2019.

[313] sumy python documentation. https://pypi.org/project/sumy/. Accessed: 2010-09-30.

[314] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer, 2019.

[315] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[316] Documentation of python implementation of rouge metrics via the package rouge-score. https://pypi.org/project/rouge-score/. Accessed: 2019-09-30.

[317] Documentation of python implementation of rouge metrics via the package pyrouge. https://pypi.org/project/pyrouge/. Accessed: 2019-09-30.

[318] Documentation of python implementation of rouge metrics via the package py-rouge. https://pypi.org/project/py-rouge/. Accessed: 2019-09-30.

[319] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[320] Yang Liu. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*, 2019.

[321] Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press, 2017.

[322] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[323] Tom Kenter, Alexey Borisov, and Maarten De Rijke. Siamese cbow: Optimizing word embeddings for sentence representations. *arXiv preprint arXiv:1606.04640*, 2016.

[324] Ye Wang, Zhi Zhou, Shan Jin, Debin Liu, and Mi Lu. Comparisons and selections of features and classifiers for short text classification. In *Iop conference series: Materials science and engineering*, volume 261, page 012018. IOP Publishing, 2017.

[325] Furqan Rustam, Imran Ashraf, Arif Mehmood, Saleem Ullah, and Gyu Sang Choi. Tweets classification on the base of sentiments for us airline companies. *Entropy*, 21(11):1078, 2019.

[326] Tae Kyun Kim and Jae Hong Park. More about the basic assumptions of t-test: normality and sample size. *Korean journal of anesthesiology*, 72(4):331, 2019.

[327] Diana Dee-Lucas and Jill H Larkin. Novice rules for assessing importance in scientific texts. *Journal of Memory and Language*, 27(3):288–308, 1988.

[328] Patrick Royston. Approximating the shapiro-wilk w-test for non-normality. *Statistics and computing*, 2(3):117–119, 1992.

[329] Nornadiah Mohd Razali, Yap Bee Wah, et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of statistical modeling and analytics*, 2(1):21–33, 2011.

[330] Robert V Hogg, Elliot A Tanis, and Dale L Zimmerman. *Probability and statistical inference*. Pearson/Prentice Hall Upper Saddle River, NJ, USA:, 2010.

[331] András Vargha and Harold D Delaney. A critique and improvement of the cl common language effect size statistics of mcgraw and wong. *Journal of Educational and Behavioral Statistics*, 25(2):101–132, 2000.

[332] Patrick Schober, Christa Boer, and Lothar A Schwarte. Correlation coefficients: appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768, 2018.

[333] N Gains. The repertory grid approach. In *Measurement of food preferences*, pages 51–76. Springer, 1994.

[334] John G Gammack. Different techniques and different aspects on declarative knowledge. In *Knowledge acquisition for expert systems*, pages 137–163. Springer, 1987.

[335] Mantz Yorke. *The repertory grid: a critical appraisal*. PhD thesis, University of Nottingham, 1983.

[336] David V Canter. A multiple sorting procedure for studying conceptual systems. Dartmouth Publishing Company, 1996.

[337] Gordon Rugg and Peter McGeorge. The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. *Expert Systems*, 14(2):80–93, 1997.

[338] Sally Fincher and Josh Tenenberg. Making sense of card sorting data. *Expert Systems*, 22(3):89–93, 2005.

[339] Mark NK Saunders. 12 using mixed methods–combining card sorts and in-depth interviews. *Handbook of Research Methods on Trust*, page 134, 2015.

[340] Martyn Hammersley. Troubles with triangulation. *Advances in mixed methods research*, pages 22–36, 2008.

[341] Terri A Scandura and Ethlyn A Williams. Research methodology in management: Current practices, trends, and implications for future research. *Academy of Management journal*, 43(6):1248–1264, 2000.

[342] Michelene TH Chi, Robert Glaser, and Marshall J Farr. *The nature of expertise*. Psychology Press, 2014.

[343] Robert B Kozma and Joel Russell. Multimedia and understanding: Expert and novice responses to different representations of chemical phenomena. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching*, 34(9):949–968, 1997.

[344] IG Borovskoy, AB Vorozhtsov, and AE Salko. Magnetogasdynamic control of burning rate of condensed system. *Defence Science Journal*, 45(1):47–49, 1995.

[345] Igor'G Borovskoj and Alexander B Vorozhtsov. Magnetic field control of burning rate and thrust in solid rocket motors. *Journal of Propulsion and Power*, 11(4):824–829, 1995.

[346] Pe J Mayo, LA Watermeier, and Felix Jiri Winberg. Electrical control of solid propellant burning. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 284(1399):488–498, 1965.

[347] Charles Bazerman. Physicists reading physics: Schema-laden purposes and purpose-laden schema. *Written communication*, 2(1):3–23, 1985.

[348] Alan H Schoenfeld and Douglas J Herrmann. Problem perception and knowledge structure in expert and novice mathematical problem solvers. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 8(5):484, 1982.

[349] Cynthia Shanahan, Timothy Shanahan, and Cynthia Misischia. Analysis of expert readers in three disciplines: History, mathematics, and chemistry. *Journal of Literacy Research*, 43(4):393–429, 2011.

[350] James M LeBreton and Jenell L Senter. Answers to 20 questions about interrater reliability and interrater agreement. *Organizational research methods*, 11(4):815–852, 2008.

[351] Natasa Gisev, J Simon Bell, and Timothy F Chen. Interrater agreement and interrater reliability: key concepts, approaches, and applications. *Research in Social and Administrative Pharmacy*, 9(3):330–338, 2013.

[352] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[353] PF Watson and A Petrie. Method agreement analysis: a review of correct methodology. *Theriogenology*, 73(9):1167–1179, 2010.

[354] Hyunsook Hong, Yunhee Choi, Seokyung Hahn, Sue Kyung Park, and Byung-Joo Park. Nomogram for sample size calculation on a straightforward basis for the kappa statistic. *Annals of Epidemiology*, 24(9):673–680, 2014.

[355] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[356] JL Fleiss, B Levin, and MC Paik. Statistical rates and proportions. hoboken, 2003.

[357] Douglas G Altman. Practical statistics for medical research chapman and hall. *London and New York*, 1991.

[358] Qingshu Xie. Agree or disagree? a demonstration of an alternative statistic to cohen's kappa for measuring the extent and reliability of agreement between observers. In *Proceedings of the Federal Committee on Statistical Methodology Research Conference*, volume 4, 2013.

[359] Joseph L Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

[360] Andrew F Hayes and Klaus Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89, 2007.

[361] Klaus Krippendorff. Computing krippendorff's alpha-reliability. 2011.

[362] John Hughes. krippendorffsalpha: An r package for measuring agreement using krippendorff's alpha coefficient. *arXiv preprint arXiv:2103.12170*, 2021.

[363] Klaus Krippendorff and Joseph L Fleiss. Reliability of binary attribute data, 1978.

[364] Domenic V Cicchetti and Alvan R Feinstein. High agreement but low kappa: Ii. resolving the paradoxes. *Journal of clinical epidemiology*, 43(6):551–558, 1990.

[365] Alvan R Feinstein and Domenic V Cicchetti. High agreement but low kappa: I. the problems of two paradoxes. *Journal of clinical epidemiology*, 43(6):543–549, 1990.

[366] Kilem L Gwet. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC, 2014.

[367] Kilem Li Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008.

[368] Daniel Klein. Implementing a general framework for assessing interrater agreement in stata. *The Stata Journal*, 18(4):871–901, 2018.

[369] Documentation of python implementation of irrcac. https://irrcac.readthedocs.io/en/latest/index.html. Accessed: 2022-09-30.

[370] Christine P Dancey and John Reidy. *Statistics without maths for psychology*. Pearson education, 2007.

[371] Pamela Thibodeau Hardiman, Robert Dufresne, and Jose P Mestre. The relation between problem categorization and problem solving among experts and novices. *Memory & cognition*, 17(5):627–638, 1989.

[372] Marie Delacre, Daniël Lakens, and Christophe Leys. Why psychologists should by default use welch's t-test instead of student's t-test. *International Review of Social Psychology*, 30(1), 2017.

[373] Patrick E McKight and Julius Najab. Kruskal-wallis test. *The corsini encyclopedia of psychology*, pages 1–1, 2010.

[374] Alan C Elliott and Linda S Hynan. A sas® macro implementation of a multiple comparison post hoc test for a kruskal–wallis analysis. *Computer methods and programs in biomedicine*, 102(1):75–80, 2011.

[375] William J Sutherland and Mark Burgman. Policy advice: use experts wisely. *Nature*, 526(7573):317–318, 2015.

[376] Martin Krallinger, Obdulia Rabal, Analia Lourenco, Julen Oyarzabal, and Alfonso Valencia. Information retrieval and text mining technologies for chemistry. *Chemical reviews*, 117(12):7673–7761, 2017.

[377] Hannah Bast and Claudius Korzen. A benchmark and evaluation for text extraction from pdf. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–10. IEEE, 2017.

[378] Github page for pdftotext. https://github.com/jalan/pdftotext. Accessed: 2022-05-23.

[379] Github page for pypdf2. https://github.com/py-pdf/PyPDF2. Accessed: 2022-05-23.

[380] Github page for pdfminer.six. https://github.com/pdfminer/pdfminer.six. Accessed: 2022-05-23.

[381] Pdfminersix.readthedocs.io. https://pdfminersix.readthedocs.io/en/latest/topic/converting_pdf_to_text.html#id2. Accessed: 2022-05-23.

[382] Grobid github. https://github.com/kermitt2/grobid. Accessed: 2022-05-23.

[383] Grobid python client github. https://github.com/kermitt2/grobid_client_python. Accessed: 2022-05-23.

[384] Filipe Rodrigues, Francisco Pereira, and Bernardete Ribeiro. Sequence labeling with multiple annotators. *Machine learning*, 95(2):165–181, 2014.

[385] Grobid documentation. https://grobid.readthedocs.io/en/latest/Principles/. Accessed: 2022-05-23.

[386] Md Saef Ullah Miah, Junaida Sulaiman, Talha Bin Sarwar, Ateeqa Naseer, Fasiha Ashraf, Kamal Zuhairi Zamli, and Rajan Jose. Sentence boundary extraction from scientific literature of electric double layer capacitor domain: Tools and techniques. *Applied Sciences*, 12(3):1352, 2022.

[387] Katrin Tomanek, Joachim Wermter, and Udo Hahn. Sentence and token splitting based on conditional random fields. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 49, page 57. Citeseer, 2007.

[388] Roman Klinger, Corinna Kolářik, Juliane Fluck, Martin Hofmann-Apitius, and Christoph M Friedrich. Detection of iupac and iupac-like chemical names. *Bioinformatics*, 24(13):i268–i276, 2008.

[389] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational linguistics*, 32(4):485–525, 2006.

[390] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, 2012.

[391] Roger Sayle, Paul Hongxing Xie, and Sorel Muresan. Improved chemical text mining of patents with infinite dictionaries and automatic spelling correction. *Journal of chemical information and modeling*, 52(1):51–62, 2012.

[392] David M Jessop, Sam E Adams, Egon L Willighagen, Lezan Hawizy, and Peter Murray-Rust. Oscar4: a flexible architecture for chemical text-mining. *Journal of cheminformatics*, 3(1):1–12, 2011.

[393] Abbas Akkasi, Ekrem Varoğlu, and Nazife Dimililer. Chemtok: A new rule based tokenizer for chemical named entity recognition. *BioMed research international*, 2016, 2016.

[394] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[395] Ariel S Schwartz and Marti A Hearst. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*, pages 451–462. World Scientific, 2002.

[396] Safaa Eltyeb and Naomie Salim. Chemical named entities recognition: a review on approaches and applications. *Journal of cheminformatics*, 6(1):1–12, 2014.

[397] Tsendsuren Munkhdalai, Meijing Li, Khuyagbaatar Batsuren, and KH Ryu. Banner-chemdner: incorporating domain knowledge in chemical and drug named entity recognition. In *Proceedings of the Fourth BioCreative Challenge Evaluation Workshop*, volume 2, pages 135–139. Citeseer, 2013.

[398] Atima Tharatipyakul, Somrak Numnark, Duangdao Wichadakul, and Supawadee Ingsriswang. Chemex: information extraction system for chemical data curation. In *BMC bioinformatics*, volume 13, pages 1–11. Springer, 2012.

[399] Daniel M Lowe and Roger A Sayle. Recognition of chemical entities in patents using leadmine. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, pages 9–11, 2015.

[400] Lezan Hawizy, David M Jessop, Nico Adams, and Peter Murray-Rust. Chemicaltagger: A tool for semantic text-mining in chemistry. *Journal of cheminformatics*, 3(1):1–13, 2011.

[401] Tim Rocktäschel, Michael Weidlich, and Ulf Leser. Chemspot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640, 2012.

[402] Ilia Korvigo, Maxim Holmatov, Anatolii Zaikovskii, and Mikhail Skoblov. Putting hands to rest: efficient deep cnn-rnn architecture for chemical named entity recognition with no hand-crafted rules. *Journal of cheminformatics*, 10(1):1–10, 2018.

[403] Kristina M Hettne, Rob H Stierum, Martijn J Schuemie, Peter JM Hendriksen, Bob JA Schijvenaars, Erik M van Mulligen, Jos Kleinjans, and Jan A Kors. A dictionary to identify small molecules and drugs in free text. *Bioinformatics*, 25(22):2983–2991, 2009.

[404] Chemspider. http://www.chemspider.com/. Accessed: 2022-05-23.

[405] Chemspider python documentation. https://chemspipy.readthedocs.io/en/latest/guide/intro.html. Accessed: 2022-05-23.

[406] Pubchem. https://pubchem.ncbi.nlm.nih.gov/. Accessed: 2022-05-23.

[407] Pubchempy documentation. https://pubchempy.readthedocs.io/en/latest/. Accessed: 2022-05-23.

[408] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.

[409] Tatyana S. Pivina2 Tatyana V. Petukhova, Vyacheslav L. Korolev and Victor P. Ivshin. SIMULATION OF 1,4,5,8-TETRANITRO-1,4,5,8- TETRAAZADECAHYDRONAPHTHALENE DECOMPOSITION MECHANISM. 2007.

[410] Ray Smith. An overview of the tesseract ocr engine. In *Ninth international conference on document analysis and recognition (ICDAR 2007)*, volume 2, pages 629–633. IEEE, 2007.

[411] Andrius Vabalas, Emma Gowen, Ellen Poliakoff, and Alexander J Casson. Machine learning algorithm validation with a limited sample size. *PloS one*, 14(11):e0224365, 2019.

[412] John C Mankins et al. Technology readiness levels. *White Paper, April*, 6(1995):1995, 1995.

[413] Katharina Tomaschek, Alison Olechowski, Steven Eppinger, and Nitin Joglekar. A survey of technology readiness level users. In *INCOSE International Symposium*, volume 26, pages 2101–2117. Wiley Online Library, 2016.

[414] Zamrady Ariffin and Aminuddin Md Arof. A review of technology readiness level (trl) in developing command and control system for maritime operations.

[415] https://www.gov.uk/guidance/defence-and-security-accelerator-terms-and-conditions- Accessed: 04-05-2022.

[416] William G. Proud. O'Brien, Sinead. Predicting the technology readiness level of ntrem papers. In *ew trends in research of energetic materials conference proceedings*, 2022.

[417] Alhanoof Althnian, Duaa AlSaeed, Heyam Al-Baity, Amani Samha, Alanoud Bin Dris, Najla Alzakari, Afnan Abou Elwafa, and Heba Kurdi. Impact of dataset size on classification performance: an empirical evaluation in the medical domain. *Applied Sciences*, 11(2):796, 2021.

[418] Nadir Durrani, Hassan Sajjad, and Fahim Dalvi. How transfer learning impacts linguistic knowledge in deep nlp models? *arXiv preprint arXiv:2105.15179*, 2021.

[419] Aparna Balagopalan, Benjamin Eyre, Frank Rudzicz, and Jekaterina Novikova. To bert or not to bert: comparing speech and language-based approaches for alzheimer's disease detection. *arXiv preprint arXiv:2008.01551*, 2020.

[420] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019.

[421] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*, 2020.

[422] Sentence-transformers documentation. https://www.sbert.net/, 2022. Accessed: 31 January 2022.

[423] Takeru Miyato, Andrew M Dai, and Ian Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

[424] Kamal Nigam, Andrew McCallum, and Tom M Mitchell. Semi-supervised text classification using em., 2006.

[425] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[426] Gilles Gasso. Logistic regression, 2019.

[427] Venkat N Gudivada, MT Irfan, E Fathi, and DL Rao. Cognitive analytics: Going beyond big data analytics and machine learning. In *Handbook of statistics*, volume 35, pages 169–205. Elsevier, 2016.

[428] Aynur Dayanik, David D Lewis, David Madigan, Vladimir Menkov, and Alexander Genkin. Constructing informative prior distributions from domain knowledge in text classification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 493–500, 2006.

[429] Joshua J Levy and A James O'Malley. Don't dismiss logistic regression: the case for sensible extraction of interactions in the era of machine learning. *BMC medical research methodology*, 20(1):1–15, 2020.

[430] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[431] James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R*. Spinger, 2013.

[432] Fabrice Colas and Pavel Brazdil. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer, 2006.

[433] Peng Chen, Bing Wang, Hau-San Wong, and De-Shuang Huang. Prediction of protein b-factors using multi-class bounded svm. *Protein and peptide letters*, 14(2):185–190, 2007.

[434] JL Raheja, Anand Mishra, and Ankit Chaudhary. Indian sign language recognition using svm. *Pattern Recognition and Image Analysis*, 26(2):434–441, 2016.

[435] Sadam Al-Azani and El-Sayed M El-Alfy. Using word embedding and ensemble learning for highly imbalanced data sentiment analysis in short arabic text. *Procedia Computer Science*, 109:359–366, 2017.

[436] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[437] Jair Cervantes, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, 408:189–215, 2020.

[438] Ashis Pradhan. Support vector machine-a survey. *International Journal of Emerging Technology and Advanced Engineering*, 2(8):82–85, 2012.

[439] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD explorations newsletter*, 2(2):1–13, 2000.

[440] Jochen Hartmann, Juliana Huppertz, Christina Schamp, and Mark Heitmann. Comparing automated text classification methods. *International Journal of Research in Marketing*, 36(1):20–38, 2019.

[441] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[442] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14, 2001.

[443] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1):69–90, 1999.

[444] Marco Bressan and Jordi Vitria. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, 2003.

[445] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pages 420–434. Springer, 2001.

[446] Danilo Bzdok, Martin Krzywinski, and Naomi Altman. Machine learning: supervised methods. *Nature methods*, 15(1):5, 2018.

[447] Sonia Singh and Priyanka Gupta. Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 27(27):97–103, 2014.

[448] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1):77–93, 2004.

[449] Bin Li, J Friedman, R Olshen, and C Stone. Classification and regression trees (cart). *Biometrics*, 40(3):358–361, 1984.

[450] Leszek Rutkowski, Maciej Jaworski, Lena Pietruczuk, and Piotr Duda. The cart decision tree for mining data streams. *Information Sciences*, 266:1–15, 2014.

[451] J Ross Quinlan et al. Bagging, boosting, and c4. 5. In *Aaai/Iaai, vol. 1*, pages 725–730, 1996.

[452] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[453] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.

[454] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[455] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE, 2014.

[456] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[457] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[458] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

[459] David G Stork, Richard O Duda, Peter E Hart, and D Stork. Pattern classification. *A Wiley-Interscience Publication*, 2001.

[460] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):1–8, 2006.

[461] Jacques Wainer and Gavin Cawley. Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Systems with Applications*, 182:115222, 2021.

[462] Damjan Krstajic, Ljubomir J Buturovic, David E Leahy, and Simon Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):1–15, 2014.

[463] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

[464] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[465] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[466] Debangana Dey, Thamar Solorio, Manuel Montes y Gómez, and Hugo Jair Escalante. Instance selection in text classification using the silhouette coefficient measure. In *Mexican international conference on artificial intelligence*, pages 357–369. Springer, 2011.

[467] Martin F Porter. An algorithm for suffix stripping. *Program*, 1980.

[468] sentence-transformers/all-distilroberta-v1, hugging face. https://huggingface.co/sentence-transformers/all-distilroberta-v1l, 2022. Accessed: 31 January 2022.

[469] distilroberta-base, hugging face. https://huggingface.co/distilroberta-base, 2022. Accessed: 31 January 2022.

[470] Guangyuan Piao. Scholarly text classification with sentence bert and entity embeddings. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 79–87. Springer, 2021.

[471] Andreas Janecek, Wilfried Gansterer, Michael Demel, and Gerhard Ecker. On the relationship between feature selection and classification accuracy. In *New challenges for feature selection in data mining and knowledge discovery*, pages 90–105. PMLR, 2008.

[472] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference*, pages 372–378. IEEE, 2014.

[473] Tzu-Tsung Wong and Po-Yang Yeh. Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1586–1594, 2019.

[474] Steve R Gunn et al. Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16, 1998.

[475] Shraddha Pandit, Suchita Gupta, et al. A comparative study on distance measuring approaches for clustering. *International Journal of Research in Computer Science*, 2(1):29–31, 2011.

[476] Max Bramer. Avoiding overfitting of decision trees. *Principles of data mining*, pages 119–134, 2007.

[477] Amalia Luque, Alejandro Carrasco, Alejandro Martín, and Ana de Las Heras. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*, 91:216–231, 2019.

[478] Riham H El Khouli, Katarzyna J Macura, Peter B Barker, Mohamed R Habba, Michael A Jacobs, and David A Bluemke. Relationship of temporal resolution to diagnostic performance for dynamic contrast enhanced mri of the breast. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 30(5):999–1004, 2009.

[479] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

[480] Jane V Carter, Jianmin Pan, Shesh N Rai, and Susan Galandiuk. Roc-ing along: Evaluation and interpretation of receiver operating characteristic curves. *Surgery*, 159(6):1638–1645, 2016.

[481] Yong Liu, Pavel Dmitriev, Yifei Huang, Andrew Brooks, and Li Dong. An evaluation of transfer learning for classifying sales engagement emails at large scale. In *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 542–548. IEEE, 2019.

[482] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[483] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[484] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[485] Muhammad Atif Tahir, Josef Kittler, and Fei Yan. Inverse random under sampling for class imbalance problem and its application to multi-label classification. *Pattern Recognition*, 45(10):3738–3750, 2012.

[486] Sara Fotouhi, Shahrokh Asadi, and Michael W Kattan. A comprehensive data level analysis for cancer diagnosis on imbalanced data. *Journal of biomedical informatics*, 90:103089, 2019.

[487] Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural networks*, 21(2-3):427–436, 2008.

[488] Chandresh Kumar Maurya, Durga Toshniwal, and Gopalan Vijendran Venkoparao. Online anomaly detection via class-imbalance learning. In *2015 Eighth International Conference on Contemporary Computing (IC3)*, pages 30–35. IEEE, 2015.

[489] Samir Al-Stouhi and Chandan K Reddy. Transfer learning for class imbalance problems with inadequate data. *Knowledge and information systems*, 48(1):201–228, 2016.

[490] Yanling Li, Guoshe Sun, and Yehang Zhu. Data imbalance problem in text classification. In *2010 Third International Symposium on Information Processing*, pages 301–305. IEEE, 2010.

[491] Nitesh V Chawla. Data mining for imbalanced datasets: An overview. *Data mining and knowledge discovery handbook*, pages 875–886, 2009.

[492] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In

2020 11th international conference on information and communication systems (ICICS), pages 243–248. IEEE, 2020.

[493] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, 2011.

[494] Hien M Nguyen, Eric W Cooper, and Katsuari Kamei. Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, 3(1):4–21, 2011.

[495] Sukarna Barua, Md Monirul Islam, Xin Yao, and Kazuyuki Murase. Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on knowledge and data engineering*, 26(2):405–425, 2012.

[496] Phakhawat Sarakit, Thanaruk Theeramunkong, and Choochart Haruechaiyasak. Improving emotion classification in imbalanced youtube dataset using smote algorithm. In *2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, pages 1–5. IEEE, 2015.

[497] Vaibhav Rupapara, Furqan Rustam, Hina Fatima Shahzad, Arif Mehmood, Imran Ashraf, and Gyu Sang Choi. Impact of smote on imbalanced text features for toxic comments classification using rvvc model. *IEEE Access*, 9:78621–78634, 2021.

[498] Layth Rafea, Abdulrahman Ahmed, and Wisam D Abdullah. Classification of a covid-19 dataset by using labels created from clustering algorithms. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1):164–173, 2021.

[499] Ira Assent. Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):340–350, 2012.

[500] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

[501] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29, 2004.

[502] Lara Lusa et al. Evaluation of smote for high-dimensional class-imbalanced microarray data. In *2012 11th international conference on machine learning and applications*, volume 2, pages 89–94. IEEE, 2012.

[503] Ch N Santhosh Kumar, K Nageswara Rao, A Govardhan, and N Sandhya. Subset k-means approach for handling imbalanced-distributed data. In *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2*, pages 497–508. Springer, 2015.

[504] Junjie Wu. The uniform effect of k-means clustering. In *Advances in K-means Clustering*, pages 17–35. Springer, 2012.

[505] Hans-Peter Kriegel, Peer Kröger, Jörg Sander, and Arthur Zimek. Density-based clustering. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 1(3):231–240, 2011.

[506] Julian Eggert and Edgar Korner. Sparse coding and nmf. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 4, pages 2529–2533. IEEE, 2004.

[507] Aleix M Martinez and Avinash C Kak. Pca versus lda. *IEEE transactions on pattern analysis and machine intelligence*, 23(2):228–233, 2001.

[508] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.

[509] GM Harshvardhan, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020.

[510] Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. Generate, annotate, and learn: Nlp with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842, 2022.

[511] Yiben Yang, Chaitanya Malaviya, Jared Fernandez, Swabha Swayamdipta, Ronan Le Bras, Ji-Ping Wang, Chandra Bhagavatula, Yejin Choi, and Doug Downey. Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546*, 2020.

[512] Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)*, pages 859–866, 2014.

[513] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93, 2016.

[514] Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110, 2022.

[515] Barbara Plank, Dirk Hovy, and Anders Søgaard. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 742–751, 2014.

[516] Trevor Cohn and Lucia Specia. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32–42, 2013.

[517] Huang-Cheng Chou and Chi-Chun Lee. Every rating matters: Joint learning of subjective labels and individual annotators for speech emotion classification. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5886–5890. IEEE, 2019.