Towards a Natural Language Processing Pipeline and Search Engine for Biomedical Associations Derived from Scientific Literature

A dissertation presented to
The Department of Surgery and Cancer

by

Dieter Galea

Submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy in Clinical Research at Imperial College London

June 2019

Declaration of Originality

I certify that this thesis, and the research to which it refers, are the product of my own work, conducted during the Doctorate in Clinical Research at Imperial College London. Any ideas or quotations from the work of other people, published or otherwise, or from my own previous work are fully acknowledged in accordance with the standard referencing practices of the discipline.

The overall proposed pipeline was designed, developed, implemented and optimized by myself. Existing open-source modules that were integrated as part of the pipeline or used in preliminary work are explicitly cited.

Manual extraction of biomarkers (and their statistical significance) from literature, was carried out by Liam Poynter as part of the meta-analysis for "network mapping of molecular biomarkers influencing radiation response in rectal cancer". The analyses, compilation and processing of the results, and designing of the figures was done by myself.

Network propagation work (Section 3.4.1) was carried out as part of the Vodafone DreamLabs project. Data collation, pre-processing and preliminary analysis was carried out by myself, along with the designing of the figure (Figure 21). The algorithm was implemented by Dr Kirill Veselkov. Identification of the drugs with a potential for re-purposing was performed by Guadalupe Gonzalez Pigorini (Section 3.4.1) and were used in this study for 'validation' of the proposed and developed natural language processing platform.

Some of the work leading to this dissertation, or presented in this dissertation, has been or is in the process of being published in a number of journal articles or book chapters, with myself as the primary author or co-author. As such work was carried out as part of this project, this has been included (in text and figures) in this dissertation report, with the appropriate citations included. Specifically, publications include:

- **Galea Dieter**, Laponogov Ivan, & Veselkov Kirill. (2018). Exploiting and assessing multi-source data for supervised biomedical named entity recognition. *Bioinformatics,* 34(14), 2474-2482. Doi:10.1093/bioinformatics/bty152

- **Galea Dieter**, Laponogov Ivan, & Veselkov Kirill. (2018). Sub-word information in pre-trained biomedical word representations: evaluation and hyper-parameter optimization. In *Proceedings of the BioNLP 2018 workshop. pp.* 56-66.
- **Galea Dieter**, Laponogov Ivan, & Veselkov Kirill. (2018). Data-Driven Visualizations in Metabolomic Phenotyping. The Handbook of Metabolic Phenotyping. John C. Lindon, Jeremy K. Nicholson, & Elaine Holmes (Ed.).
- **Galea Dieter**, Inglese Paolo, Cammack Lidia, Strittmatter Nicole, Rebec Monica, Mirnezami Reza, Laponogov Ivan, Kinross James, Nicholson Jeremy, Takats Zoltan, & Veselkov Kirill. (2017). Translational utility of a hierarchical classification strategy in biomolecular data analytics. *Scientific Reports,* 7.
- Poynter Liam, **Galea Dieter**, Veselkov Kirill, Mirnezami Alexander, Kinross James, Takats Zoltan, Nicholson Jeremy, Darzi Ara, Mirnezami Reza. (2019). Network mapping of molecular biomarkers influencing radiation response in rectal cancer. *Clinical Colorectal Cancer*.
- Veselkov Kirill, Gonzalez Pigorini Guadalupe, Aljifri Shahad, **Galea Dieter**, Mirnezami Reza, Youssef Jozef, Bronstein Michael & Laponogov Ivan. (2019). HyperFoods: Machine intelligent mapping of cancer-beating molecules in foods. *Scientific Reports,* 9.
- Laponogov Ivan, Sadawi Noureddin, **Galea Dieter**, Mirnezami Reza, & Veselkov Kirill. ChemDistiller: an engine for metabolite annotation in mass spectrometry. *Bioinformatics*, 34(12) pp. 2096-2102.
- Veselkov Kirill, Sleeman Jonathan, Claude Emmanuelle, Vissers Johannes, **Galea Dieter,** Mroz Anna, Laponogov Ivan, Towers Mark, Tonge Robert, Mirnezami Reza, Takats Zoltan, Nicholson Jeremy, & Langridge James. (2018). BASIS: High-performance bioinformatics platform for processing of large-scale mass spectrometry imaging data in chemically augmented histology. *Scientific Reports,* 8.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution 4.0 International Licence (CC BY).

Under this licence, you may copy and redistribute the material in any medium or format for both commercial and non-commercial purposes. You may also create and distribute modified versions of the work. This on the condition that you credit the author.

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. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

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.

Acknowledgements

Firstly, I would like to express my biggest gratitude and appreciation to Dr Kirill Veselkov and Prof Zoltan Takats for providing me with the opportunity to work closely in their research groups and for their supervision throughout the project;

I am grateful for the funding of this project by Imperial College Stratified Medicine Graduate Training Programme in Systems Medicine and Spectroscopic Profiling (STRATiGRAD) programme and Waters Corporation;

I would like to also thank colleagues and faculty members who have provided their input to improve this work and maximize its utility. Specifically: Dr Ivan Laponogov, Nicolas Ayoub, Guadalupe Gonzalez Pigorini, Shahad Aljifri, Dr Timothy Ebbels and Prof Jeremy Everett.

Finally, I am forever grateful for my family (Helen, Raymond, Raisa, Kaiser, Charlton and Lara) and close friends (Aaron, Andrè, Adrian, Justins, Juan, Keith, Kenneth, Olof and Vincen) for their unconditional and continuous support, companionship, and motivation, and to Liam for helping me balance work and life and staying relatively sane during this doctorate.

Short Abstract

Biomedical research is published at a rapid rate, with PubMed containing over 29 million publications. A natural language processing pipeline (NLP) facilitating information extraction is required. Existing pipelines achieve promising performance, but are often restricted to a small number of bioentities (such as genes and diseases), ignore negative associations, and treat new claims and background sentences equally. Here, different NLP tasks required to develop a scalable and generalizable open source pipeline for biomedical association extraction that tackles these limitations are investigated. In turn, this is used to build a repository of queryable associations.

Starting by optimizing how biomedical language is represented in machine learning (ML) models, state-of-the-art representations are obtained and subsequently used in downstream tasks, including bioentity recognition. Latter work indicates that current recognition models are poorly generalizable, resulting in unrealistic performance when applied at scale. Additionally, it is shown here that acquiring more data does not improve ML-based entity recognition performance. Beyond ML methods, this work presents a number of dictionary-based approaches and graph-based dictionaries for more than 13 sources covering metabolites, genes/proteins, species, chemicals, toxins, drugs, diseases, foods, food compounds and anatomy are compiled. These are used to annotate PubMed for subsequent association extraction.

To achieve a diverse association extraction pipeline for 10 entity types, we attempt to find a balance between generalizable rules and ML models. A neural model is trained to identify novel association claims with 94% accuracy and a rule-based approach to identify negated statements with up to 91% accuracy. A set of rules are devised to define associations. Quantitative evaluation shows promising results, however further work is required. Extracted associations are stored in a graph database, enabling querying for associations reported in literature, as well as discovering new potential indirect linkages. To demonstrate its future use, a frontend proof of concept is presented.

Long Abstract

The rate by which biomedical research is published has increased over the years, with the PubMed repository now containing over 29 million publications. Such rate makes it impossible to keep up with research through manual searches. Additionally, when new findings are considered in isolation, they may be limited by their statistical power, resulting in poor reproducibility. This therefore requires a process of automation – a natural language processing pipeline that facilitates information extraction; such as, potential linkages between bioentities like genes and diseases. Such workflow requires identifying bioentities from unstructured text through named entity recognition and identifying a relationship between entities. This is likely to speed up the clinical validation stage in biomarker discovery due to potentially improved statistical power through discovery cross-publication validation, and inference of new entity linkages.

While existing pipelines achieve highly promising performance, these are often restricted to a small number of bioentities, such as genes and diseases, ignore negative associations, and treat new claim statements and background sentences equally. In this work, different NLP tasks required to develop a scalable and generalizable open source pipeline for biomedical association extraction are investigated. In turn, the developed pipeline is used to build a repository of queryable associations.

Machine learning methods have been researched and used extensively in natural language processing. These approaches require converting text into numerical representations. Distributional representations have been shown to outperform traditional feature engineering-based representations, however, optimizing such representations for biomedical NLP has been minimally explored. By training and optimizing word2vec and fastText models, here state-of-the-art pretrained biomedical embeddings are achieved, that are subsequently used in downstream pipeline tasks.

Latest advancements in machine learning achieve state-of-the-art performance for named entity recognition on benchmark datasets. However, such work is biased by the data used, and through cross-validation such models are identified to be poorly generalizable, therefore the performance metrics are not realistic when such models are applied at scale. Through power

analyses, this work shows that acquiring more training data would generally not improve machine learning-based NER performance. In addition to machine learning approaches, a number of dictionary-based approaches are implemented as part of the developed pipeline and graph-based dictionaries for more 13 sources covering metabolites, genes/proteins, species, chemicals, toxins, drugs, diseases, foods, food compounds and anatomy are compiled. These implementations are used to annotate PubMed for subsequent association extraction.

Relationship extraction is also commonly performed with machine learning methods as well as with rule-based approaches. Similar to NER, one major limitation to scaling these to a diverse number of entity types is generalizability, time-consuming crafting of rules, and/or lack of training data. Aiming towards a diverse pipeline which extracts associations between 10 different entity types, in this work a balance is sought between simple, generalizable rules and machine learning models. A neural model is trained to identify novel claims with 94% accuracy and a rule-based approach to identify negated statements with up to 90.90% accuracy. Finally, a set of rules are defined to identify associations. Quantitative evaluation shows promising results, however devising a fair benchmark dataset is required to evaluate its true performance, as current corpora do not collectively capture: negations, novel associations and false associations. In future work, the association extraction developed here is envisioned to be used as a broad association identifier, with more fine-grained and optimized models used for specific entity class pairs. This information from multiple identifiers can be stored in the devised graph structure of associations and therefore provides flexibility during querying – without requiring to re-run the pipeline with alternative methods. Nonetheless, the current graph enables querying for associations explicitly found in text, as is demonstrated through a frontend proof of concept, as well as traversing through associations to discover new potential indirect linkages.

Table of Contents

Declaration of Originality *3*
Copyright Declaration *5*
Acknowledgements *6*
Short Abstract *7*
Long Abstract *8*
List of Figures *14*
List of Supplementary Figures *20*
List of Supplementary Tables *20*
List of Tables *21*
List of Abbreviations *24*
Chapter 1 - Introduction *26*
1.1 General Introduction **26**
1.2 Scope of this thesis **29**
1.3 Structure of this thesis **32**
2 Chapter 2 – Background and Methods in Natural Language Processing *36*
2.1 General machine learning methods **36**
2.1.1 Linear and Logistic Regression **36**
2.1.2 Support Vector Machines **37**
2.1.3 Naïve Bayes **39**
2.2 Sequential machine learning methods: traditional approaches **40**
2.3 Sequential machine learning methods: Neural network approaches **42**
2.3.1 Recurrent Neural Networks **42**
2.3.2 BiLSTM-CRF architecture **44**
2.4 Word Representations **46**
2.4.1 Word2vec models 49
2.4.2 fastText models 51
2.4.3 Evaluating distributional word embeddings 52
2.4.3.1 Intrinsic evaluation 52
2.4.3.2 Extrinsic evaluation 55
2.5 Biomedical Abbreviation Resolution **55**
2.6 String Matching Algorithms and Data Structures **57**
2.7 Named Entity Recognition **59**
2.7.1 Feature Engineering **60**
2.7.2 Architectures and Performance **63**
2.8 Association Extraction **64**
2.9 Biomedical NLP tools **66**

2.9.1 PolySearch and PolySearch2 67
2.9.2 Arrowsmith 69
2.9.3 BeFree 70
2.9.4 Precompiled association databases 72
2.9.5 Recognizing the gaps 73
3 Chapter 3 – Developing and utilizing methods for the identification of biomedical associations from structured and unstructured data 75
3.1 Abstract 75
3.2 Aims and Objectives 75
3.3 Introduction 76
3.4 Methods 77
3.4.1 Network propagation and supervised classification 77
3.4.2 Hierarchical classification 78
3.4.3 Systematic reviews 81
3.5 Results and Discussion 81
3.5.1 Network propagation 81
3.5.2 Systematic Reviews 83
3.5.3 Hierarchical Classification 85
3.6 Conclusion(s) and Future Direction(s) 88
4 Chapter 4 – Comparing and optimizing biomedical word representations 90
4.1 Abstract 90
4.2 Aims and Objectives 90
4.3 Introduction 91
4.4 Methods 92
4.4.1 Training data pre-processing 92
4.4.2 Embeddings and hyper-parameters 92
4.4.3 Intrinsic Evaluation 93
4.4.4 Extrinsic Evaluation 94
4.4.5 Performance Generalizability 95
4.5 Results and Discussion 95
4.5.1 Corpus 95
4.5.2 General trends: word2vec hyper-parameter selection 95
4.5.3 General trends: fastText hyper-parameter selection 97
4.5.4 Model performance comparison – Intrinsic evaluation 97
4.5.5 Model performance comparison – Extrinsic evaluation 100
4.5.6 Effect of n-grams size 105
4.5.7 Optimized models' performance 106
4.5.8 Generalizability of optimal model performance 108
4.6 Conclusion(s) and Future Direction(s) 110
Chapter 5 - Processing of PubMed articles: Parsing, tokenization, abbreviation resolution, negation detection and sentence classification 111
5.1 Abstract 111
5.2 Aims and Objectives 111
5.3 Introduction 112
5.4 Methods 115
5.4.1 Parsing, Sentence and word tokenization 115
5.4.2 Acronym and abbreviation solver 116
5.4.3 Negation cue detection 116
5.4.4 Negation scope detection 117
5.4.5 Sentence classification 118

5.5 Results and Discussion 119
5.5.1 Acronym and abbreviation solver 119
5.5.2 Negation Cue and Scope Detection 122
5.5.3 Sentence Classification 125
5.6 Conclusion(s) and Future Direction(s) 130

6 Chapter 6 – Implementation of biomedical named entity recognition methods and power analyses 132

6.1 Abstract 132
6.2 Aims and objectives 133
6.3 Introduction 133
6.4 Methods 138
6.4.1 String matching implementations 138
6.4.2 Vocabulary compilation 139
6.4.3 Compiling dictionary graphs 141
6.4.4 Visualizing dictionary graph(s) 143
6.4.5 Building dictionary graph(s) databases 144
6.4.6 Compiling training data 144
6.4.7 Corpora pre-processing 145
6.4.8 Model training and prediction 147
6.4.9 Power analyses 148
6.4.10 Orthographic feature analysis 149
6.5 Results and Discussion 150
6.5.1 Dictionary graph 150
6.5.2 Dictionary visualization 152
6.5.3 Power analyses: Model training strategy 156
6.5.4 Power analyses: Identifying genes and proteins 160
6.5.5 Power analyses: Identifying variants 164
6.5.6 Power analyses: Identifying chemicals, drugs and metabolites 168
6.5.7 Power analyses: Identifying RNA 169
6.6 Conclusion(s) and Future Direction(s) 171

7 Chapter 7 – Biomedical associations: Extraction, database, pipeline and search engine 173

7.1 Abstract 173
7.2 Aims and objectives 174
7.3 Introduction 174
7.4 Methods 176
7.4.1 Relation extraction 176
7.4.2 Database 177
7.4.2.1 Database Management System and Graph Model 177
7.4.2.2 Exporting the graph 178
7.4.2.3 Populating the database 180
7.4.3 Overall Pipeline 180
7.4.3.1 Structure and features 180
7.4.3.2 Quantitative Evaluation 182
7.4.4 Proof of Concept 183
7.5 Results and Discussion 183
7.5.1 Database Management System 183
7.5.2 Graph Model 185
7.5.3 Model Benchmarks 187
7.5.4 Populating the database 189
7.5.5 Overall Pipeline 189
7.5.6 Evaluation 192
7.5.7 Frontend proof of concept 200
7.5.8 Towards automating systematic reviews 202
7.6 Conclusion(s) and Future Direction(s) 203

8 Chapter 8 – Conclusions and future work 205
References 213
Supplementary Tables and Figures 244
Supplementary Methods 253

List of Figures

Figure 1. Diverse potential application of biomarkers by various levels of specificity: from broad screening to subtyping, treatment, and response monitoring. Adapted from: (Chan, Wasinger, & Leong, 2016).....27

Figure 2. Basic natural language processing workflow for information extraction; specifically, extraction of associations/relations from unstructured text. Firstly, text is parsed from the unstructured source files to a standardized format. Pre-processing fragments the text into sentences and subsequently individual words/tokens. Additional pre-processing involves removal of stopwords and normalization through lowercasing, for example. Entities such as genes, chemicals, and diseases are recognized through either dictionary-matching or using machine-learning-based approaches and ultimately relationships between such entities are identified. 28

Figure 3. Overall workflow of the HASKEE pipeline: from parsing, to trivial pre-processing, extension of existing dependencies for improved parsing, abbreviation resolution, article/sentence scoring, named entity recognition and ultimately relation extraction. The output of the pipeline is a set of files that are compatible for importing into a graph database. The location where additional investigations executed as part of this study would fit as part of the pipeline, specifically word embeddings and their use for neural named entity recognition (neural NER), is also indicated for future work. The main existing packages which are used and extended on include: PubMed parser (see Section 5.4.1) (Achakulvisut et al., 2016) and the Schwartz algorithm implementation (see Section 5.4.2) (Schwartz & Hearst, 2003; Gooch, 2017/2018). 32

Figure 4. Structure of the presented thesis. Following an introduction and definition of the scope (Chapter 1), background information and specific methods are introduced (Chapter 2). Approaches for identifying bioassociations are developed and discussed in Chapter 3, and Chapters 4-7 report different parts of the developed pipeline for extraction of biomedical associations and demonstrate a proof of concept for a complimentary frontend. In the final chapter (Chapter 8), the work is concluded, and future directions are discussed. 33

Figure 5. Graph visualization of the comparison between a linear regression model and a logistic model. The linear regression model predicts a continuous target variable whereas the logistic model returns a range between 0-1 representing probability of a sample being in the positive class (in a binary classification problem). Source: (Sayad, 2019). 37

Figure 6. Two-dimensional plots indicating: A) all the possible boundaries that discriminate between 2 classes; and B) a hyperplane which maximizes the margin between the support vectors as in SVM. Source: (Drakos, 2018) 38

Figure 7. Graphic representation of the different models and the relationship between them, where logistic regression is the conditional equivalent of a Naïve Bayes model, linear-chain CRFs are the conditional equivalent to Hidden Markov Models, and the sequential equivalent of logistic regression. Source: (Sutton & McCallum, 2012). 40

Figure 8. Graphical models for different linear-chain CRF variants where transition states Y depend only on the previous state and current observations (A), on the previous state, current

observations and previous observations (B), and all observations (C). Adapted from (Sutton & McCallum, 2012). 42

Figure 9. Basic structure of a recurrent neural network, where x represents the input and h is the neural predicted output. Source: ("Understanding LSTM Networks -- colah's blog," n.d.) 43

Figure 10. Visual example where predicting output h3 requires information from previous timepoints Xo and X1. Source: ("Understanding LSTM Networks -- colah's blog," n.d.)....43

Figure 11. LSTM unit is composed of 4 neural layers: a tanh layer that is also found in a typical recurrent neural network unit, and an additional 3 sigmoid layers that act as gates, controlling which information flows through to the cell state. Source: ("Understanding LSTM Networks -- colah's blog," n.d.)....44

Figure 12. The BiLSTM architecture. Forward and backward LSTM layers are stacked to capture bidirectional information in a sequence labeling task. Source: (Hu, Li, Hu, & Yang, 2018). 45

Figure 13. The BiLSTM-CRF architecture. CRF layer is stacked on top of the BiLSTM layer to learn the label sequence constraints. Source: ("CRF Layer on the Top of BiLSTM - 1," n.d.) 45

Figure 14. Feature sets typically engineered and extracted to represent biomedical words. Adapted from (Campos, Matos, & Oliveira, 2013a). 48

Figure 15. Graphic representation of the continuous bag-of-words (CBOW) and Skip-gram (SG) distributed word vector representation models. In the CBOW architecture, the probability of predicting a word wt given the context words ($wt-2, wt-1, wt+1, wt+2$) is maximized. In the Skip-gram architecture, given a word wt, the probability of predicting the surrounding words $wt-2, wt-1, wt+1, wt+2$ is maximized. Source: (Mikolov et al., 2013). 50

Figure 16. Trie and radix tree data structures. Trie tree (left) and radix tree (right) for the same set of keywords: {plan, play, poll, post}, where radix tree groups nodes with a single descendant; saving on memory and space. Source: ("Radix tree - Swift Data Structure and Algorithms [Book]," n.d.) 58

Figure 17. Text annotated for "person" (PERSON), "organization" (ORG), and "date" (DATE) by a named entity recognition model. Source: spaCy as visualized by displaCy Named Entity Visualizer. 59

Figure 18. Categorization of various features used in biomedical named entity recognition. Source: (Alshaikhdeeb & Ahmad, 2016) 61

Figure 19. Different syntactic structures utilized in relation extraction approaches. A) CoNLL dependency tree; B) PennTreeBank phrase structure tree; C) head dependencies; D) Stanford dependencies; E) predicate-argument structure. Source: (Miyao et al., 2009) 65

Figure 20. Local and global shallow linguistic kernels (B-C) and dependency kernel (D-E) devised by BeFree for association extraction, exemplified by the sentence (A) mentioning the gene EHD3 and disease MDD (Major Depressive Disorder). The local context kernel (B)

exploits orthographic and shallow linguistic features such as POS tags, lemmas and stems for tokens within a window of each entity mention. The global context kernel (D) captures positional and sentence order information, represented by bi-trigrams. “associated” is identified as the least common subsume (LCS). Features considered in the dependency kernel include the token, stem, lemma, POS tag, and the role (disease or gene) (E). Adapted from Àlex Bravo et al. (2015).....72

Figure 21. Graphic overview of the proposed network propagation approach developed and applied for the identification of drugs that may potentially be re-purposed for anti-cancer treatment. Gene-drug interactions are propagated to identify drug-influenced genes (RIGHT). Drugs with a similar mechanism are expected to overlap in the propagated gene profile. In a more patient-based genome approach, patient mutation data can also be propagated through the human interactome network to identify influencer genes (LEFT). The overlap between the propagated networks may indicate drugs that potentially can be more suitable based on a patient’s genomic profile. 80

Figure 22. Network diagram summarizing published biomarkers studied in colorectal cancers. Size of nodes represents number of unique biomarkers for an ontology. Color of a node represents overall statistical significance reported in the original studies. Source: (Poynter et al., 2019).....84

Figure 23. Classification of bacterial spectra into their respective taxonomic classes. a) taxonomic tree of the bacterial species considered, color-coded by their respective genera; b) classification performance for each level of the taxonomic tree (class, order, family, genus and species) and Gram properties; c) novel semi-quantitative visualization of the species classification performance indicating misclassification across the species and genera. Source: (Galea et al., 2017).....85

Figure 24. Classification of cancer genomic data into classes defined in literature. a) Hierarchical classification tree derived from literature and used for supervised training and prediction of different cancer types; b) semi-quantitative hierarchical classification performance of the different cancer subtypes and indication of where misclassification occurs across the same or different cancer types. Source: (Galea et al., 2017) 86

Figure 25. Prediction of the respective cancer type (BLCA, BRCA, GBM, KIRC, KIRP, LAML, LGG, LUAD and PRAD) for a selection of cancer subtypes. Source: (Galea et al., 2017) 87

Figure 26. Word2vec word representations reduced to a three-dimensional space with t-SNE. A selection of words inferred to be related to the word “metabolites” by cosine similarity are highlighted and isolated. In turn, “h.p.l.c” appears to cluster with words relating to the technologies used in metabolic profiling. Figure published in (Galea et al., 2018c). 96

Figure 27. Intrinsic performance of word2vec (w2v) and fastText (FastT) word representations measured by similarity and relatedness for UMNSRS, HDO and XADO datasets, when varying the hyper-parameters: dimensions, negative sampling size and minimum word count. 98

Figure 28. Intrinsic performance of word2vec (w2v) and fastText (FastT) word representations measured by similarity and relatedness for UMNSRS, HDO and XADO

datasets, when varying the hyper-parameters: sub-sampling rate, learning rate (alpha) and window size. 99

Figure 29. Extrinsic performance of word2vec (w2v) and fastText (FastT) word representations measured by named entity recognition accuracy (F-score) on the corpora BC2GM, JNLPBA and CHEMDNER, when varying the hyper-parameters: window size, dimensions, minimum word count, negative sampling size, sub-sampling rate and learning rate (alpha). 101

Figure 30. Training and development sequential sentence classification accuracies under different hyper-parameter and architecture configurations. A) 1 BiLSTM layer each with 50 hidden units with word2vec embeddings; B) 3 BiLSTM layers each with 50 hidden units with fastText embeddings; C) 2 BiLSTM layers with 100 hidden units (as reported by (Reimers & Gurevych, 2017)) using fastText embeddings; and D) 3 BiLSTM layers each with 50 hidden units with fastText embeddings on the ~4 million abstract corpus. A-C were trained on the 200k PubMed RCT corpus....127

Figure 31. Confusion matrix for the sentence classification model trained on the PubMed RCT training set and applied to the PubMed RCT test set. 129

Figure 32. Confusion matrix for the sentence classification model trained on the PubMed structured abstracts training set and applied to the PubMed structured abstracts test set equivalent. 130

Figure 33. Normalized confusion matrix for the sentence classification model trained on the PubMed structured abstracts training set and applied to the PubMed RCT test set. 130

Figure 34. Dictionary graph structure example for the term "caffeine" from CHEBI. String terms are assigned the node type "name" and match the keyword list compiled in Section 6.4.2. This is linked to an "id" node which contains the database accession id for the term through an "ID" edge. Synonymous terms and IUPAC name for "caffeine" are linked through the respective edges. Secondary accession identifiers such as "CHEBI:3295", "CHEBI:41472", "CHEBI:22982" are linked to the primary id through another "ID" edge. Ontologies for the term "caffeine" are assigned by linking the primary accession id through "IS A" relationships to the primary accession id of the ontological term. In this example, the terms "purine alkaloid" and "trimethyl xanthine" are two ontological terms with "CHEBI:26385" and "CHEBI:27134" as their primary identifiers, respectively. These are both ontologies for "CHEBI:27732" (caffeine) and therefore are linked by an "IS_A" relationship....143

Figure 35. Part of the Catalogue of Life dictionary graph showing 5 species (*K. pneumoniae*, *K. singaporensis*, *K. oxytoca*, *K. granulomatic*, and *K. variicola*) for the Klebsiella genus. Each species has multiple variants of the name which are directly attached to the species ID by :ID relationship. However, *K. granulomatis* has a sister ID 16961901 as synonym, which in turn has other variants. Given the graph structure, we can retrieve these identifiers which are not directly linked to 20774109. This allows to maintain the original IDs in the dictionary i.e. 16961901. We also note that 11473088 (*K. pneumoniae*) also has further sub-children which are sub-species. These should also be included when querying “Klebsiella” as a genus therefore we add infinite depth to the IS_A relationship. 153

Figure 36. Network of human metabolites constructed from the human metabolite database (HMDB) classified into chemical ontologies and linked to their corresponding synonyms. Triglycerides, diglycerides, cardiolipins, phosphatidylcholines and phosphatidylethanolamines are encircled, and the metabolites in the ontology 'Organic Derivatives' are highlighted. Synonyms are shown for 'phosphatidylethanolamine PE(14:0/18:3(6Z, 9Z, 12Z)). Figure published in (Galea, Laponogov, & Veselkov, 2018a). 154

Figure 37. Raw learning curve plots when genes, proteins and variants were considered as a single "GeneProteinVariants" superclass. Different corpora may differ in the annotation standards for the same entities, resulting in poor/no predictive performance on other corpora. However, overall performance generally does not decrease substantially with the introduction of new data from other sources. Figure published in (Galea et al., 2018b). 162

Figure 38. 'GeneProtein' class learning curves obtained by each corpus. Learning curves for models trained on each corpus and applied to test data from all corpora to test the generalizability of each corpus. (A) AIMed; (B) OSIRIS; (C) CellFinder; (D) IEPA; (E) miRTeX; (F) SETH; (G) VariomeCorpus; (H) mTOR. Figure published in (Galea et al., 2018b). 163

Figure 39. Average accuracy measured by F-score for the "GeneProtein" class when entities are predicted with the model trained on all merged data. Corpora annotating genes and/or proteins were merged and split for training and testing. Mean, median and weighted mean F-score obtained when applying the trained model to the test data is shown, where performance appears dependent on the training size up to 1200 documents. Figure published in (Galea et al., 2018b). 164

Figure 40. Learning curves obtained when: (i) multiple sources are used as training data to predict test data from each other corpus, individually; and (ii) each corpus is excluded from training and its test data is predicted by the other corpora (leave-corpus-out cross-validation). Each subplot represents training and testing of the different entity classes: (A) Genes and proteins (dashed lines represent leave-corpus-out validation approach); (B) variants; (C) chemicals; (D) metabolites; (E) RNA; and (F) drugs. Figure published in (Galea et al., 2018b). 167

Figure 41. "Variants" class learning curves obtained by each corpus. Learning curves for models trained on each corpus and applied to test data from all corpora to test the generalizability of each corpus. (A) OSIRIS; (B) SETH; (C) tmVar; and (D) SNPcorpus. Figure published in (Galea et al., 2018b). 168

Figure 42. Orthographic features identified as significant to the entity classes: Gene-Protein, RNA, variants, chemicals, drugs and metabolites. Highlighted features were identified to be univariately significant in the orthographic feature analysis for a given entity class in a given corpus. Rows represent such orthographic features while sectors/columns represent corpora; grouped by the entity classes. Figure published in (Galea et al., 2018b). 170

Figure 43. Graph model iterations. Different graph structures considered to model the association data. A) Basic unit of the graph-based model for the database management system. Each association is represented by a single node at the center of the graph that is linked to nodes representing the entities and the document(s) supporting this association claim. B) Alternative model structure that introduces additional edges between the entities

themselves. While this information introduces redundancy to the model, it may improve traversing performance for obtaining directly and indirectly related entities. C) Graph model that represents associations with a node. This allows direct look-up of an association and allows for storing additional attributes such as scores, and type (e.g. in-silico predicted association). D) A graph model equivalent to B) with the structure of C). This introduces redundancies but may improve traversing and look-up speed. 178

Figure 44. Final graph model. Detailed property graph model used in production, with metadata properties assigned to documents and raw sentence strings and negation assigned as properties to the sentences. The document type attribute is derived from parsed publication/document type (e.g. clinical study or in silico prediction), negation is detected by the negation cue detection module, and section represents the predicted paper section for each sentence. 185

Figure 45. Scalability of neo4j. Preliminary benchmarks for the effect of node size on simple query execution time A) before indexing; and B) after indexing of the queried entity 'id' property. Identical queries were run and averaged. Due to caching of queries and results, the timings for the first queries (Ai and Bi) and subsequent repetitions (Aii and Bii) were kept separate. The difference is evident in the scale of the execution time. A linear O(n) reference line is shown with respect to the average observed execution time. 188

Figure 46. HASKEE documentation on each of the pipeline's modules, available resources and utilities. In addition to technical usage example, documentation includes background information, practical suggestions and warnings, and citations to original resources or publications utilizing the mentioned resource or algorithm. 191

Figure 47. Slack pipeline progress monitoring. HASKEE integration of slack progress and status logging using the slack-progress library for desktop and mobile monitoring of each pipeline module. 192

Figure 48. Screenshot of the proof of concept for the results page when querying for a link between "cancer" and "cetrorelix". The recognized entities are listed and articles claiming an association are listed below. Recognized entities can be added and/or modified by the user in case of false positive and false negative entities recognized from the inputted statement. Articles are represented by their PubMed identifier (PMID), their article type, the sentences claiming such association, and the year of publication. Each entry could have a button (represented by a red circular button in the screenshot here) that enables to capture user feedback in instances where the article is falsely recalled. Additionally, a dropdown button can provide additional metadata and article information. 201

Figure 49. Screenshot of the proof of concept for the results page when querying for a link between "cancer" and "cetrorelix" showing the retrieved synonyms from the dictionary graph for each of the recognized entities. 202

List of Supplementary Figures

Supplementary Figure 1. PRISMA flow diagram of the study filtering and selection process used to generate a corpus of studies that was in turn used to generate a systematic review of molecular biomarkers influencing radiation response in rectal cancer. Source: (Poynter et al., 2019) 244

Supplementary Figure 2. Output for the hierarchical leave-class-out prediction of bacterial species. Predicting hierarchical taxonomic classes (Gram positive, Bacilli, and Lactobacillales) for Streptococcus agalactiae using the algorithm developed in Galea et al. (2017). Source: (Galea et al., 2017). 245

Supplementary Figure 3. Word2vec and fastText training time benchmarks as a function of different values for the various hyperparameters: (i) window size; (ii) negative sub-sampling rate; (iii) sampling rate; (iv) word count; (v) alpha/learning rate; (vi) dimensions; and (vii) n-gram range. Y-axis units expressed in terms of fold-change to the default hyper-parameters. Source: (Galea et al., 2018c) 246

List of Supplementary Tables

Supplementary Table 1. List of compiled biomedical corpora, the available formats and sources. Number of documents for each corpus may vary based on the source, and a document "unit" may be defined differently in different corpora (e.g. abstract, title, whole manuscript text). Source may be the original manuscript published, or if not available (or available in a different formation), other secondary sources hosting the resource. When a corpus is available in various formats and multiple sources, these are indicated. As links may become offline with time, a more dynamic and community-updated table is also hosted on https://github.com/dterg/biomedical_corpora. Source: Galea et al. (2018b). 247

List of Tables

Table 1. List of python packages used in the HASKEE pipeline. Modification and/or extension of packages to tailor them for our objectives or improve them, are indicated by a *. Algorithms/implementations which are not packaged but their code is integrated in HASKEE is indicated by < >. 31

Table 2. Categories of features used in biomedical named entity recognition literature. Adapted from: (Alshaikhdeeb & Ahmad, 2016). 62

Table 3. Top 10 non-anti-cancer drugs identified with the potential of having anti-cancer properties, their respective target and a brief description of their mechanism. Adapted from (Gonzalez Pigorini, 2018). 82

Table 4. Number of metabolic features identified to be altered between bacterial classes by univariate feature selection. Source: (Galea, 2015) 87

Table 5. Pathways identified to be commonly altered between different cancer types. 88

Table 6. Tokens and unique tokens in the processed training data derived from PubMed articles at different word frequency thresholds. Source: (Galea et al., 2018c). 95

Table 7. Top 5 most similar words to the out-of-vocabulary chemicals: 1,2-dichloromethane and 1-(dimethylamino)-2-mthyl-3,4-diphenylbutane-1,3-diol, and gene ZNF560. Source: (Galea et al., 2018c). 102

Table 8. Top 5 most similar words to phosphatidylinositol-4,5-bisphosphate. Syntactically similar terms are recalled by fastText whereas word2vec recalls less syntactically similar terms but relevant entities such as abbreviated forms and synonyms, where PtdIns(4,5)P2 and PIP2 are synonymous to the query term. Source: (Galea et al., 2018c). 103

Table 9. Top 10 most similar words to rs2243250; 590C/T polymorphism of Interleukin 4. "RS"- prefixed terms refer to Reference SNP identifiers. Source: (Galea et al., 2018c). 103

Table 10. Top 10 most similar words to acrodysostosis - a skeletal malformations disorder. Most of the recalled terms refer to genetic disorders of the bone, skin or endocrine system. Source: (Galea et al., 2018c). 103

Table 11. Top 5 most similar terms to the out-of-vocabulary genetic variant LRG_1:g.8463G>C. "RS-" prefixed terms represent database accession identifiers for reference variants. Source: (Galea et al., 2018c). 104

Table 12. Top 10 most similar words to the term: "ZNF580 – Zinc Finger Protein 580" in the word2vec and fastText embeddings. Source: (Galea et al., 2018c). 104

Table 13. Top 10 most similar words to the term: "1,2-dichloroethane" in the word2vec and fastText embeddings. Source: (Galea et al., 2018c). 104

Table 14. Top 10 most similar words to the term: "zinc_finger_protein" in the word2vec and fastText embeddings. Source: (Galea et al., 2018c). 105

Table 15. The role of character n-gram ranges on intrinsic (UMNSRS, HDO and XADO; upper row = similarity, lower row = relatedness) and extrinsic performance (JNLPBA, CHEMDNER and BC2GM). Highest performance is highlighted in bold and accuracies within standard error of highest performance is indicated in italics. Source: (Galea et al., 2018c). 106

Table 16. Compilation of intrinsic and extrinsic performance for our trained embeddings and others reported in literature. 107

Table 17. Optimized hyper-parameters for word2vec (w2v) and fastText (FastT) across intrinsic and extrinsic datasets. Source: (Galea et al., 2018c). 107

Table 18. word2vec (w2v) and fastText (FastT) hyper-parameters optimized across intrinsic standards and extrinsic corpora. Source: (Galea et al., 2018c). 108

Table 19. Analogy resolution performance for the optimal word2vec and fastText models on the BMASS dataset. 109

Table 20. Examples of analogies from the 2 fastText's best performing relationship types (M2-noun-form-of and M1-adjectival-form-of) and 2 fastText's worst performing relationship types (L3-has-tradename and L2-has-lab-number). 110

Table 21. Evaluation of negative association detection by negation cue detection filters on the PolySearch datasets. 122

Table 22. Evaluation of the negation module based on the datasets from EU-ADR corpus. SA = speculative associations; PA = positive associations; NA = negative associations. 124

Table 23. Negation evaluation with the BioScope corpus. 125

Table 24. Negation scope detection accuracies achieved by the BiLSTM-CRF architecture under 4 evaluation methods: exact scope, token match, left margin match and right margin match. 125

Table 25. Abstract sentence classification into PIBOSO classes. 126

Table 26. Per-class performance metrics (precision, recall, F-score and support) for sentence classification model trained on 200k PubMed RCTs. 128

Table 27. Per-class performance metrics (precision, recall, F-score and support) for sentence classification model trained on 3 million PubMed structured abstracts and applied to its equivalent test set. 128

Table 28. Per-class performance metrics (precision, recall, F-score and support) for sentence classification model trained on 3 million PubMed structured abstracts and applied to the PubMed RCT test set. 128

Table 29. Dictionaries compiled from different sources as graphs. Ontological relationships and synonymy information is retained through respective edges. Terms are represented by name nodes and their respective source identifier. 141

Table 30. Data formats supported by a number of network visualization packages, the respective programming languages developed in, and usage license. 155

Table 31. List of compiled biomedically-related corpora, corresponding year of publication, different formats of availability and a brief description of the data, if available. Where corpora are available from multiple sources, size may differ and each document may be defined differently in different corpora (e.g. title, whole manuscript document, abstract). Originally published in (Galea et al., 2018b). For compactness, sources have been excluded from this version. A more complete and updated version is also available on GitHub: https://github.com/dterg/biomedical_corpora, and in Supplementary Table 1. 157

Table 32. Basic statistics on the number of entities and unique entities in each corpus, the original entity classes and the new entity class to which they were remapped in this study. As published in (Galea et al., 2018b). 159

Table 33. Comparison of query execution times in a relational database and neo4j with variable relation depth. Source: (Robinson, Webber, & Eifrem, 2013). 187

Table 34. Results returned by PubMed for a query "flunisolide cancer". 198

Table 35. Results returned by PubMed for a query “fluticasone fuorate cancer”. 199

List of Abbreviations

AUC	Area under the curve
BC2GM	BioCreative II Gene Mention
BiLSTM	Bi-directional long-short term memory
BLCA	Urothelial bladder carcinoma
BRCA	Breast adenocarcinoma
CBOW	Continuous Bag-Of-Words
CNN	Convolutional neural network
CRF	Conditional random fields
DOM	Document object model
FDR	False discovery rate
GBM	Glioblastoma multiforme
HDO	Human disease ontology
HMM	Hidden markov model
IOB	Inside-outside-beginning
IUPAC	International Union of Pure and Applied Chemistry
KIRC	Kidney renal clear cell carcinoma
KIRP	Kidney renal papillary cell carcinoma
LAML	Acute myeloid leukemia
LDA	Linear discriminant analysis
LGG	Lower-grade glioma
LSTM	Long-short term memory
LUAD	Lung adenocarcinoma
MAP	Maximum aposteriori
ML	Machine learning
MLE	Maximum likelihood estimation
MMC	Maximum margin criterion
NCBI	National center for biotechnology information
NER	Named entity recognition
NLP	Natural language processing
OOV	Out-of-vocabulary
PCA	Principal component analysis
PIBOSO	Population, intervention, background, outcome, study design, other
PICO	Population, intervention, comparison, outcome
PMID	PubMed Identifier
POS	Part-of-speech
PRAD	Prostate adenocarcinoma
RCT	Randomized clinical trial
Regex	Regular expression(s)
RMSE	Root mean square error
RNN	Recurrent neural network
ROC	Receiver operating characteristic
SG	Skip-gram
SGD	Stochastic gradient descent
SVM	Support vector machines
TCGA	The Cancer Genome Atlas
TF	Term frequency
TFIDF	Term frequency – inverse document frequency

UMNSRS	University of Minnesota Minneapolis semantic relatedness/similarity
XADO	Xenopus anatomy and development ontology
XML	extensible markup language

Chapter 1 - Introduction

1.1 General Introduction

The development of high-throughput 'omics' technologies has resulted in the generation of large quantity of data. About 2 billion human genomes are predicted to be sequenced by 2025, generating 1 Exabyte (1 trillion Terabytes) of data (Stephens et al., 2015). This has led to a rapid growth in research identifying putative biomarkers, with thousands of publications issued each year, increasing quasi-exponentially, reporting biomarkers at various stages of clinical management (Figure 1). However, these findings are commonly limited by their statistical power that results in poor reproducibility. This issue has been identified in a general survey by Nature (Baker, 2016) and discussed in relation to biomarker discovery in (McShane, 2017; Scherer, 2017). As a consequence, few biomarkers have progressed to the clinical validation stage. 'Validating' findings early in the process by searching public knowledge can speed up the clinical validation stage and save on downstream costs.

Another issue with such quantity of research is that information is lost in the mass. As a result, in clinical research, performing a systematic review and meta-analysis is critical to collate research findings. However, for a study to be included in a systematic review, this can take up to 6.5 years, and 23% of systematic reviews become out of date within 2 years of publishing due to new evidence published (Elliott et al., 2014). This highlights the need for a process, a natural language processing workflow, that facilitates or automates this. Such information extraction workflow involves a number of text mining tasks: from pre-processing, to the recognition of bioentities (named entity recognition), extraction of associations between

bioentities (relation/association extraction), and classification/categorization of articles (Figure 2).

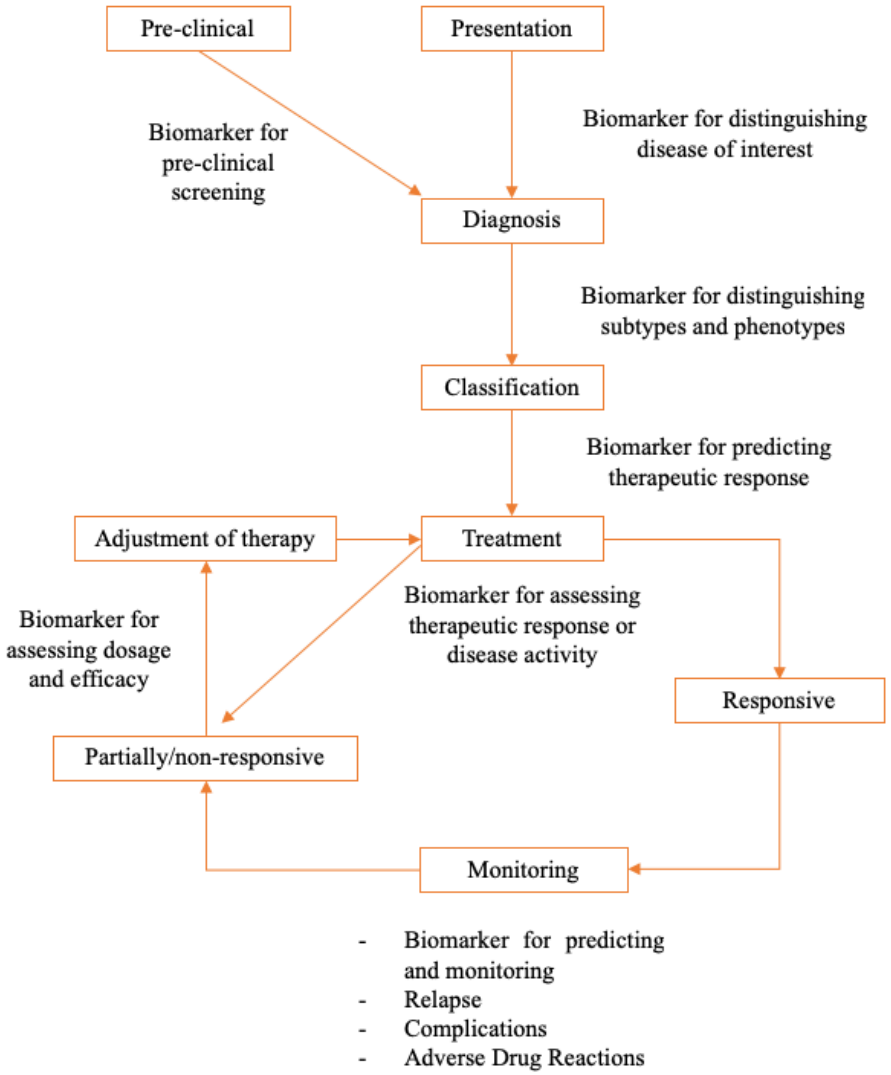


Figure 1. Diverse potential application of biomarkers by various levels of specificity: from broad screening to subtyping, treatment, and response monitoring. Adapted from: (Chan, Wasinger, & Leong, 2016).

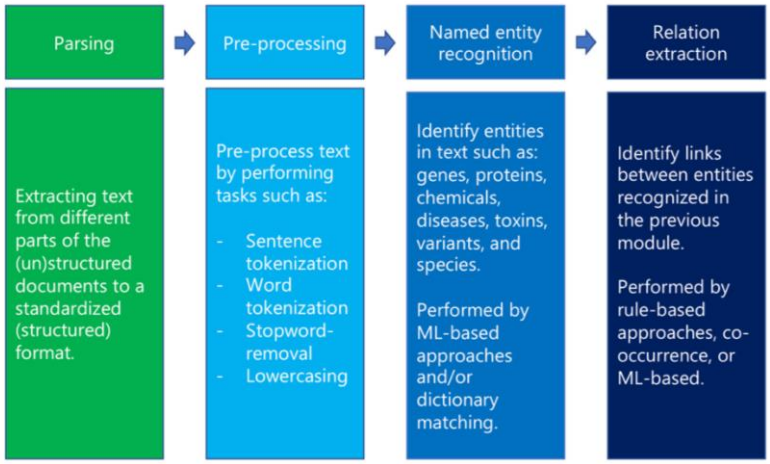


Figure 2. Basic natural language processing workflow for information extraction; specifically, extraction of associations/relations from unstructured text. Firstly, text is parsed from the unstructured source files to a standardized format. Pre-processing fragments the text into sentences and subsequently individual words/tokens. Additional pre-processing involves removal of stopwords and normalization through lowercasing, for example. Entities such as genes, chemicals, and diseases are recognized through either dictionary-matching or using machine-learning-based approaches and ultimately relationships between such entities are identified.

Each of these tasks has been researched mostly in isolation, achieving state-of-the-art performances, however little research has investigated their utility and realistic performance in a scalable system for more translational use. For the information extraction pipelines/tools that are currently available, these exhibit a number of limitations:

(i) Are restricted to a small number of entities, and often binary relations between them, such as: genes and diseases, or chemicals and diseases;

(ii) Ignore negative associations: when publication authors claim there is no association between X and Y, these findings are discarded rather than treated as negative findings;

(iii) Information is either extracted from specific parts of the publication (such as results), or if all sections are used, new claim statements are not differentiated from background sentences.

1.2 Scope of this thesis

With the need for an information extraction pipeline tailored for biomedical association extraction defined and limitations of existing platforms briefly outlined, the overall scope of this thesis is to research, develop and/or implement translational natural language processing methods required to extract diverse biomedical associations from scientific literature, as part of a highly configurable end-to-end pipeline. This pipeline is designed and in turn utilized to generate a queryable network-based repository of associations.

Overall aimed at facilitating the extraction of biomedical knowledge and findings, the specific aims of this work are:

(i) To research and develop an open-source modular pipeline with advancements in natural language processing for extraction of biomedical associations explicitly mentioned in scientific literature;

(ii) To build a repository and platform that enables the querying of literature-extracted biomedical associations;

(iii) To compute a graph that enables inference of medical associations that while they are public knowledge, are not stated explicitly in scientific literature;

With the above aims, the pipeline/search engine is named HASKEE: Hypotheses and ASsociations from KnowlEdgE; as the pipeline can be used to derive novel hypothetical associations by inference, or retrieve associations explicitly mentioned in literature.

To achieve this, a number of objectives are defined:

(i) To develop, demonstrate and envision how different applications and bioinformatics solutions can make use of an association extraction natural language processing pipeline to 'validate' their output;

(ii) To investigate the impact of different word representations on biomedical natural language processing performance and optimize such representations for downstream tasks;

(iii) To implement a module as part of a pipeline that parses and processes millions of biomedical publications efficiently for subsequent tasks;

(iv) To investigate the generalizability of machine learning named entity recognition methods to provide realistic performance metrics for highly diverse named entity recognition desired for large scale annotation;

(v) To implement different named entity recognition approaches as part of a biomedical natural language processing pipeline;

(vi) To develop a generalizable association extraction module as part of a pipeline that determines a positive, negative and/or false links between bioentities recognized in the named entity recognition module;

(vii) To develop a graph model that is suitable for storing biomedical associations in a format that is efficient and maximizes traceability of textual source;

(viii) To demonstrate the presentation of the pipeline and association graph results by developing a frontend proof of concept.

The development of the HASKEE pipeline is done in python and utilizes a number of third-party packages/libraries. Packages used for general data handling and processing, parallelization, progress logging, and machine learning, are used without modification. For

objectives where existing packages could be used as base code, these were utilized and extended. Specifically, the library *pubmed_parser*, used to import and parse PubMed articles, was modified and extended in this work as described in Section 5.4.1. Additionally, although not available as a package by itself, the Schwartz algorithm used for abbreviation resolution (Section 5.4.2), was extended and integrated as part of HASKEE. The full list of packages required to run and develop HASKEE is shown in Table 1. The overall workflow is shown in Figure 3.

Table 1. List of python packages used in the HASKEE pipeline. Modification and/or extension of packages to tailor them for our objectives or improve them, are indicated by a *. Algorithms/implementations which are not packaged but their code is integrated in HASKEE is indicated by < >.

Python Package	Use case
bokeh	Plotting and visualization
flashtext	Flat dictionary matching
gensim	Training and loading word embeddings (Section 4.4.2)
h5py	General machine learning model saving/loading
keras	Neural network framework
keras_contrib	Conditional random field layer implementation for keras
lxml	General training data parsing (Section 6.4.7)
matplotlib	Plotting and visualization
nltk	Generic NLP pre-processing
numpy	General data handling
pandas	General data handling
pathos	Parallelising processing
pubmed_parser*	PubMed article parser (Section 5.4.1)
py2neo	Neo4j database client (Section 7.4.2.3)
pygtrie	Character-based trie implementation for dictionary matching (Section 6.4.1)
pymongo	MongoDB client (Section 7.4.2.1)
<schwartz>*	Abbreviation resolution
sklearn	Machine learning framework
slackclient	Slack client to monitor pipeline progress (Section 7.5.5)
slacker	Slack client to monitor pipeline progress (Section 7.5.5)
tensorflow-gpu	Neural network backend framework
tqdm	Progress bar
unicodecsv	General data handling

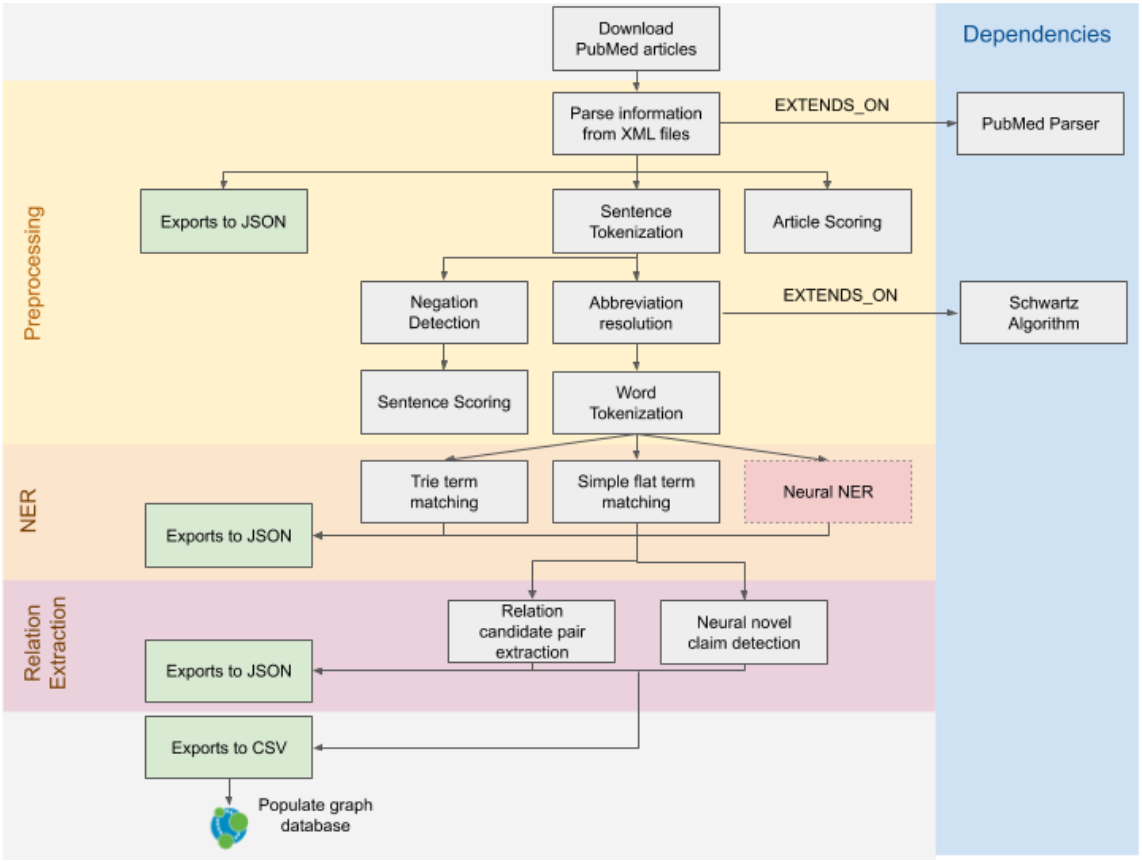


Figure 3. Overall workflow of the HASKEE pipeline: from parsing, to trivial pre-processing, extension of existing dependencies for improved parsing, abbreviation resolution, article/sentence scoring, named entity recognition and ultimately relation extraction. The output of the pipeline is a set of files that are compatible for importing into a graph database. The location where additional investigations executed as part of this study would fit as part of the pipeline, specifically word embeddings and their use for neural named entity recognition (neural NER), is also indicated for future work. The main existing packages which are used and extended on include: PubMed parser (see Section 5.4.1) (Achakulvisut et al., 2016) and the Schwartz algorithm implementation (see Section 5.4.2) (Schwartz & Hearst, 2003; Gooch, 2017/2018).

1.3 Structure of this thesis

To cover the aims and objectives defined in Section 1.2, this work is divided into 7 main parts (Figure 4). Following this brief introduction (Chapter 1), general background information and methodological details that have been or are currently used in the natural language processing field are provided, with specific reference to the biomedical subfield (Chapter 2). In the last part of Chapter 2, a review of the existing NLP tools in the biomedical field is included,

describing their advantages and disadvantages and highlighting the current gaps. Subsequently, 5 parts follow which report the main work performed as part of this thesis (Chapter 3-7). In the final chapter (Chapter 8), this work is concluded and directions for future work are discussed.

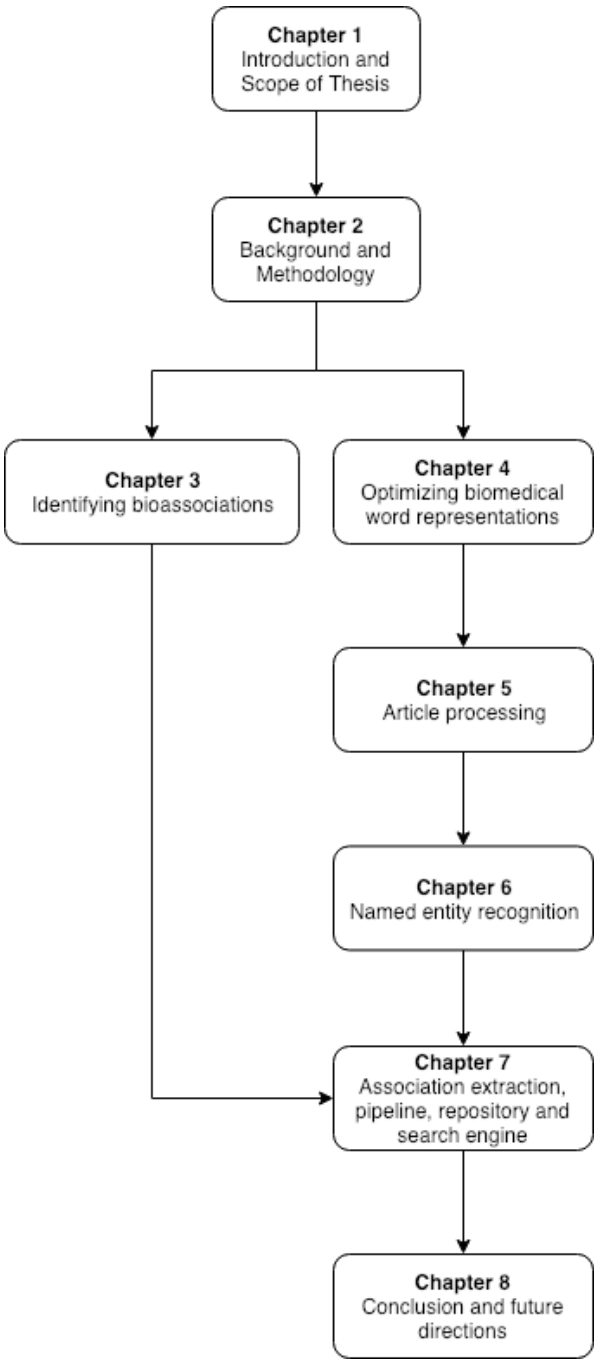


Figure 4. Structure of the presented thesis. Following an introduction and definition of the scope (Chapter 1), background information and specific methods are introduced (Chapter 2). Approaches for identifying bioassociations are developed and discussed in Chapter 3, and Chapters 4-7 report different parts of the developed pipeline for extraction of biomedical associations and demonstrate a proof of concept for a complimentary frontend. In the final chapter (Chapter 8), the work is concluded, and future directions are discussed.

In the first main part of this work (Chapter 3), non-textual methods that are commonly used in the field to identify links between bioentities, such as machine learning methods and/or univariate statistical methods, are introduced and developed. Additionally, laborious manual systematic reviewing of literature is executed and discussed. This chapter gives examples of different methods and different fields of research that can make use of a literature-searching platform to validate or discover novel findings. Some of the findings from this chapter are used to quantitatively 'validate' the developed pipeline (Chapter 7).

In the second part of this work, different NLP modules involved to develop an association extraction pipeline are researched and developed. As shown in Figure 3, the pipeline is divided into 3 main parts: (i) preprocessing (Chapter 4); (ii) named entity recognition (Chapter 5); and (iii) relation extraction (Chapter 6). Fundamental to all modules/chapters however, is identifying the optimal way to represent biomedical text for machine learning models used throughout the pipeline/thesis, described in Chapter 4. This is therefore subsequently followed by: the extraction and preprocessing of the textual information into a structured format (Chapter 5); (iii) identifying bioentities through performing named entity recognition (Chapter 6); and ultimately (iv) devising a strategy for identifying associations, compiling results into a graph-based model, compiling the work modules into a modular and highly customizable pipeline and demonstrating a proof of concept for a complimentary frontend (Chapter 7).

Each of these parts is structured as follows:

(i) **Abstract:** concise overview of the work, conclusions and implications;

(ii) **Aims and objectives:** overall purposes of the current work and specific objectives how this was tackled;

(iii) **Introduction**: a brief introduction to related research in the field, descriptively highlighting potential limitations aimed to be tackled by the current work;

(iv) **Methods**: description of the methodological work carried out in full detail to enable reproducibility of the work;

(v) **Results and Discussion**: compilation of the results obtained by the methods carried out and a discussion of their implications. Discussion is focused on the work in the current chapter as well as in light of developing the translational pipeline – the main goal of the project;

(vi) **Conclusion and Future Direction**: an overview of what was achieved in this work and what is envisioned for the future as further developments/improvements.

In the final chapter (Chapter 8), an overview of the conclusions and overall directions for future work are given.

Chapter 2 – Background and Methods in Natural Language Processing

2.1 General machine learning methods

2.1.1 Linear and Logistic Regression

In typical machine learning tasks, the aim is to predict interdependent variables. Given a set of observed features $x = \{x_0, x_1, \dots, x_n\}$, the goal is to predict the output variable y. Machine learning can be broadly categorized into 2 main tasks: regression and classification. In the former, with methods such as linear regression, a model of the general form (Equation 1) is built to predict a continuous variable ($\hat{y}$), where the coefficients b_i (and bias b_0) are learnt during training by ordinary least squares for each feature x_i. Starting with random values for such parameters, a loss function such as the RMSE (root mean squared error) is defined (Equation 2). With optimization algorithms such as stochastic gradient descent (SGD), the parameters are optimized by minimizing the error. In natural language processing, regression defines tasks such as propensity-based sentiment analysis, where a continuous sentiment score is predicted for a sentence/document.

$$\hat{y} = b_0 + b_1 x_1 + \cdots + b_n x_n \qquad \text{Equation 1}$$

$$L_{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(w^T x(i) - y(i))^2}{n}} \qquad \text{Equation 2}$$

When the target variable (dependent variable) y is a categorical variable, this is considered a classification problem. Given a set of observed features, the aim is to predict the class(es) the sample belongs to. A number of natural language processing tasks are based on classification,

such as: document topic classification, sentiment classification, claim detection, fraud detection, and spam detection.

Logistic regression is a simple linear classification method equivalent to linear regression (Figure 5), taking the same form as Equation 1. To predict classes, however, the output is converted to a probability, ranging from 0-1. This is achieved by "squashing" the weighted sum of the input features (and bias term) (Equation 1) with the logistic function (Equation 3).

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \qquad \text{Equation 3}$$

In logistic regression, coefficients are learnt while maximizing the posterior probability by minimizing the log loss (Equation 4).

$$L_{log}(\hat{y}, y) = \log\left(1 + \exp\left(-\left(\hat{y}_{[t]} - \hat{y}_{[k]}\right)\right)\right) \qquad \text{Equation 4}$$

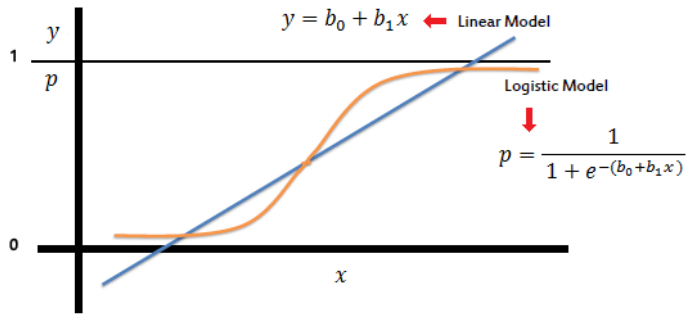


Figure 5. Graph visualization of the comparison between a linear regression model and a logistic model. The linear regression model predicts a continuous target variable whereas the logistic model returns a range between 0-1 representing probability of a sample being in the positive class (in a binary classification problem). Source: (Sayad, 2019).

2.1.2 Support Vector Machines

Support Vector Machines (SVM) is an alternative classification approach, where the decision boundary is defined by the outlining/supporting samples (vectors) maximizing the margin

(Figure 6). Classically, SVM does not output a probability estimation over the output classes but performs a hard decision by assigning classes to samples through computing the decision function (Equation 5). The parameters are optimized by minimizing the hinge loss (or SVM loss) function defined in Equation 6.

$$\hat{y} = \begin{cases} -1 \; if \; w^T.x + b < 0, \\ +1 \; if \; w^T.x + b \geq 1 \end{cases} \qquad \text{Equation 5}$$

$$L_{hinge(binary)}(\hat{y}, y) = \max(0, 1 - y \cdot \hat{y}) \qquad \text{Equation 6}$$

SVM has been used extensively as a classification model in the NLP field, achieving previous state-of-the-art performance in tasks such as named entity recognition and relation extraction (see Section 2.7.2 and Section 2.8).

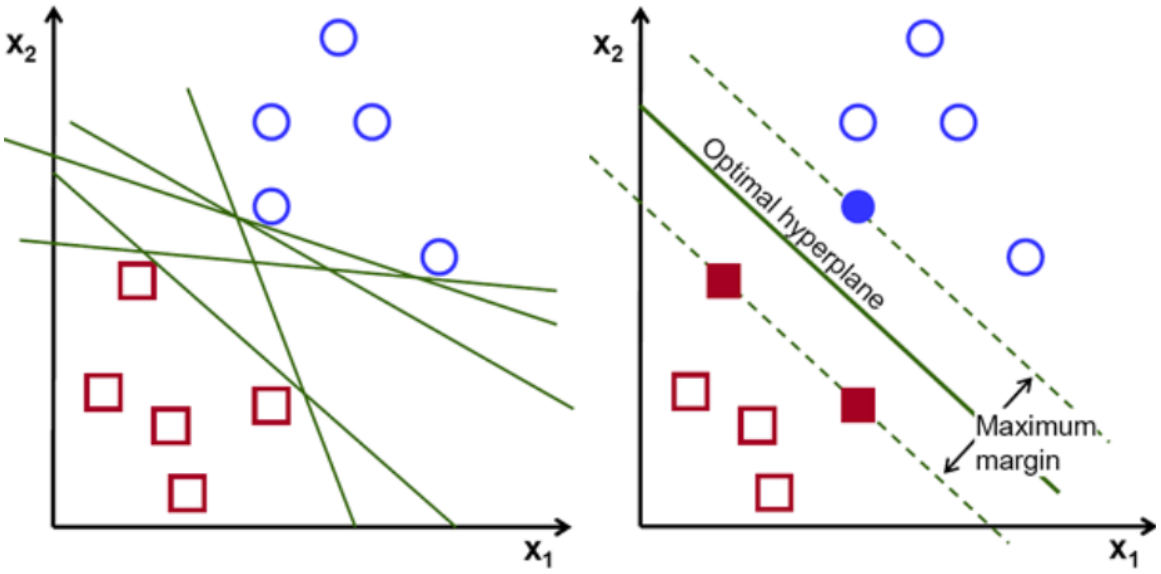


Figure 6. Two-dimensional plots indicating: A) all the possible boundaries that discriminate between 2 classes; and B) a hyperplane which maximizes the margin between the support vectors as in SVM. Source: (Drakos, 2018)

2.1.3 Naïve Bayes

Naïve Bayes is a linear classification method based on the Bayes' theorem (Equation 7). Being a generative model (the generative analogue to logistic regression), it models the probability of obtaining the observed label set from a given feature set (posterior; Equation 8) by considering the joint probability P(Y,X). Considering Equation 7, as the denominator does not depend on the label Y, this is constant and the numerator is equivalent to the joint probability P(Y,X).

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \qquad \text{Equation 7}$$

$$posterior = \frac{likelihood\ .prior}{evidence} \qquad \text{Equation 8}$$

Applying the chain rule, the joint probability distribution can be written as:

$$\begin{aligned} P(Y, x_1, \dots, x_n) &= P(x_1, \dots, x_n, Y) \\ &= P(x_1|x_2, \dots, x_n, Y)P(x_2, \dots, x_n, Y) \\ &= P(x_1|x_2, \dots, x_n, Y)P(x_2|x_3, \dots, x_n, Y)P(x_3, \dots, x_n, Y) \\ &= \cdots \\ &= P(x_1|x_2, \dots, x_n, Y) \dots P(x_{n-1}|x_n, Y)P(x_n|Y)P(Y) \end{aligned} \qquad \text{Equation 9}$$

As the name implies, Naïve Bayes has the naïve assumption of conditional independence for every feature, implying that each feature is independent. This implies that:

$$P(x_i \mid x_{i+1}, Y) = P(x_i \mid Y) \qquad \text{Equation 10}$$

Based on Equation 10, the joint distribution model from Equation 9 can be expressed as Equation 11.

$$P(Y|x_1, \dots, x_n) = P(Y, x_1, \dots, x_n) \qquad \text{Equation 11}$$

$$= P(Y) \prod_{i=1}^{n} P(x_i \mid Y)$$

This naïve Bayes probability model is combined with the maximum a posteriori (MAP) decision rule to pick the most probable hypothesis, constructing the naïve Bayes classifier. For a set of k classes, a Bayes classifier predicts a target label y by Equation 12.

$$\hat{y} = \underset{k \in \{1,\ldots,K\}}{\operatorname{argmax}} \; P(Y_k) \prod_{i=1}^{n} P(x_i \mid C_k) \qquad \text{Equation 12}$$

The simplicity of the Naïve Bayes classifier made it a popular choice in NLP tasks, often used as a performance baseline to more complex architectures such as neural networks (Section 2.3).

2.2 Sequential machine learning methods: traditional approaches

Logistic regression and Naïve Bayes are examples of models that formulate the problem in a discriminative and generative approach, respectively (Figure 7). In the latter, the joint probability distribution $p(y, x)$ is modeled, while in the former the conditional probability distribution $p(y|x)$ is computed (see Section 2.1).

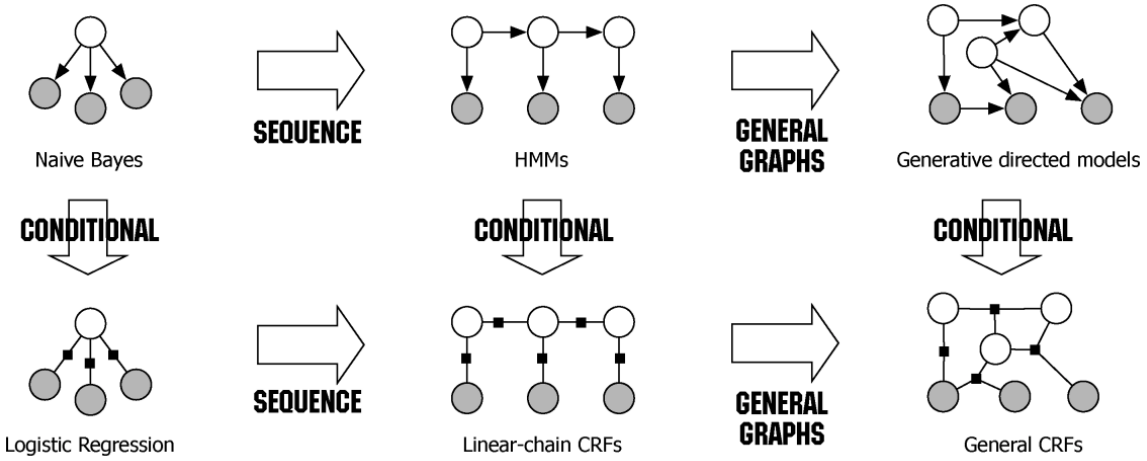


Figure 7. Graphic representation of the different models and the relationship between them, where logistic regression is the conditional equivalent of a Naïve Bayes model, linear-chain CRFs are the conditional equivalent to Hidden Markov Models, and the sequential equivalent of logistic regression. Source: (Sutton & McCallum, 2012).

Given the sequential nature of textual data and tasks such as part-of-speech tagging and named entity recognition, complex dependencies exist between positions. While performing independent classification at each position is one approach, dependency information is lost in the modeling. Graphical models such as Conditional Random Fields (CRF), and the generative equivalent, Hidden Markov Models (HMM) (Figure 7), mitigate this.

HMMs model sequences of observations x = {x_1,x_2, …, x_n} and their corresponding labels jointly by assuming an underlying sequence of states s = {s_1, s_2, …, s_n} (Qiu et al., 2010). Additionally, they assume: (i) each state (s_t) is dependent on its immediate preceding state (s_{t-1}) and independent of previous ancestors ($s_1, \ldots, s_{t-2}$); (ii) each observed variable (x_t) is dependent only on the current state s_t (Qiu et al., 2010). Being a generative model which factorizes as Equation 13, the joint probability for a HMM factorizes as Equation 14.

$$p(y,x) = p(y)p(x|y) \qquad \text{Equation 13}$$

$$p(y,x) = \prod_{t=1}^{T} p(y_t|y_{t-1})p(x_t|y_t) \qquad \text{Equation 14}$$

Linear CRFs are the discriminative equivalent of HMM, modeling the conditional distribution $p(y|x)$ and hence exclude modelling conditional $p(x)$. The variation in the dependencies between y and observations x leads to a number of CRF variants (Figure 8).

Generalizing the linear-chain CRFs model to a more general factor graph obtains the general CRF models. Formal definitions and further details are provided in (Sutton & McCallum, 2012).

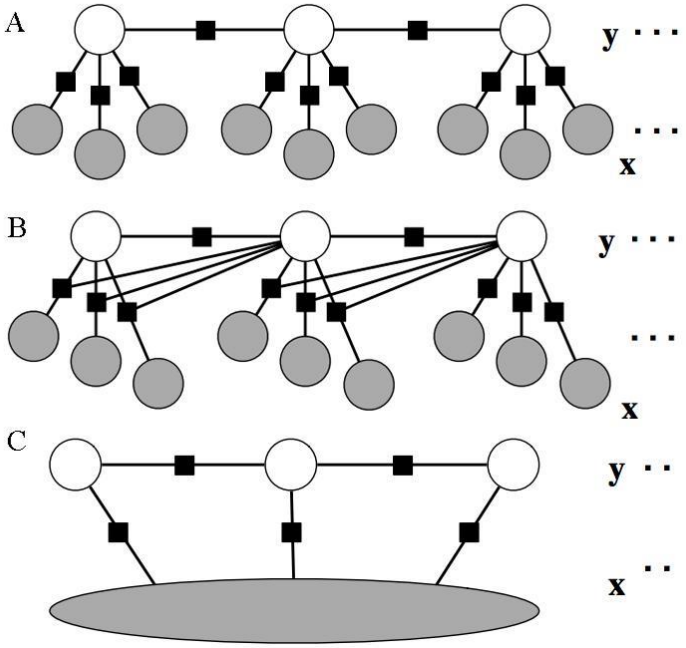


Figure 8. Graphical models for different linear-chain CRF variants where transition states Y depend only on the previous state and current observations (A), on the previous state, current observations and previous observations (B), and all observations (C). Adapted from (Sutton & McCallum, 2012).

2.3 Sequential machine learning methods: Neural network approaches

2.3.1 Recurrent Neural Networks

In Section 2.1 and 2.2 we introduced traditional linear models. Such models are unable to capture non-linear data trends, natively. Transforming the data prior to learning linear models by applying non-linear functions is one approach to mitigate this, however this requires knowing which transformation function to apply as a pre-processing step. This is where neural networks are advantageous; not only do they enable learning the model parameters, but also the representation of the data by applying non-linear transformations.

Recurrent neural networks are analogous to sequence-based models such as CRFs, with the chain-like structure enabling passing information from one unit to the next (Figure 9). This makes it a useful architecture for sequential data such as text, where making a prediction at a

specific time point might require ‘knowledge’ from previous time points (Figure 10). Unlike traditional neural networks, RNNs are able to persist information along the network.

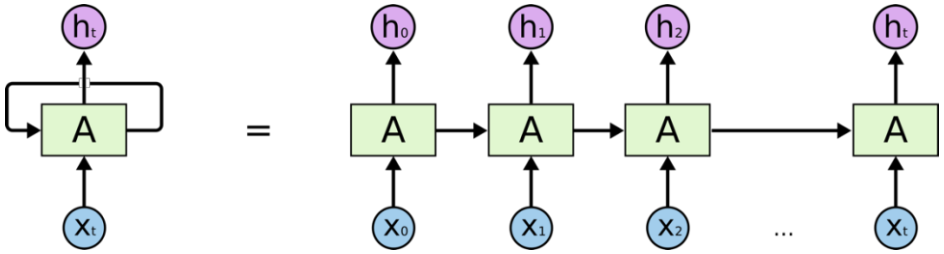


Figure 9. Basic structure of a recurrent neural network, where x represents the input and h is the neural predicted output. Source: (“Understanding LSTM Networks -- colah’s blog,” n.d.)

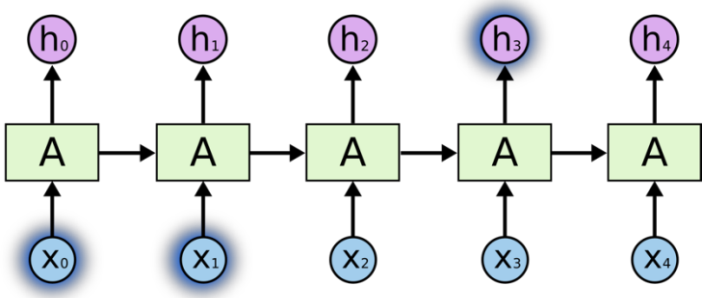


Figure 10. Visual example where predicting output h3 requires information from previous timepoints Xo and X1. Source: (“Understanding LSTM Networks -- colah’s blog,” n.d.)

RNNs however fall short when the gap between the source of information and the current prediction increases, and are therefore unable to handle long-term dependencies. This limitation is mitigated by long short-term memory networks (LSTMs). Unlike RNNs, LSTM modules have 4 layers, 3 of which are “gates” which control which information passes through to the next cell state (Figure 11). Each gate is a sigmoid layer that outputs a {0,1} value that in turn is pointwise multiplied to the current cell state. More detailed and technical explanation is provided by the excellent blog post: (“Understanding LSTM Networks -- colah’s blog,” n.d.).

LSTM is the basic layer used in the BiLSTM-CRF model architectures used extensively throughout this project.

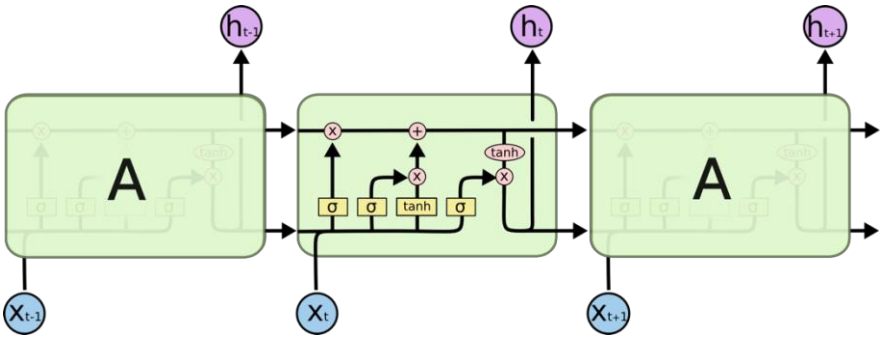


Figure 11. LSTM unit is composed of 4 neural layers: a tanh layer that is also found in a typical recurrent neural network unit, and an additional 3 sigmoid layers that act as gates, controlling which information flows through to the cell state. Source: ("Understanding LSTM Networks -- colah's blog," n.d.)

2.3.2 BiLSTM-CRF architecture

In a unidirectional chain-like architecture such as a forward LSTM, predictions are based on information retained only from predecessors, i.e. $p(y_t|\{x_1, \dots, x_t\})$. In a sequence-labeling task, having knowledge of the successors may help in improving predictions. This bidirectionality is achieved by stacking 2 LSTM layers, one in the forward direction and one in reverse time order, and the outputs are concatenated at each time point. This is referred to as a BiLSTM architecture (Figure 12).

In the context of named entity recognition (Section 2.7), the input is the sequence of words and the output is a label assigned to each word. This is therefore a sequence-to-sequence classification problem. In multi-word entities, additional classes (IB 'prefixes') are defined to establish the boundaries. Therefore, each class (e.g. "chemical") is now defined by two classes: B-chemical and I-chemical.

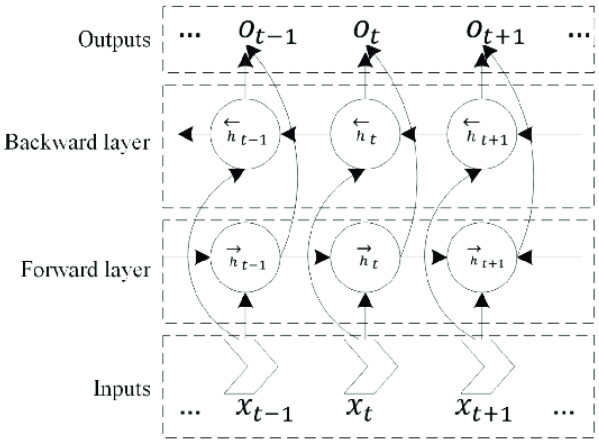


Figure 12. The BiLSTM architecture. Forward and backward LSTM layers are stacked to capture bidirectional information in a sequence labeling task. Source: (Hu, Li, Hu, & Yang, 2018).

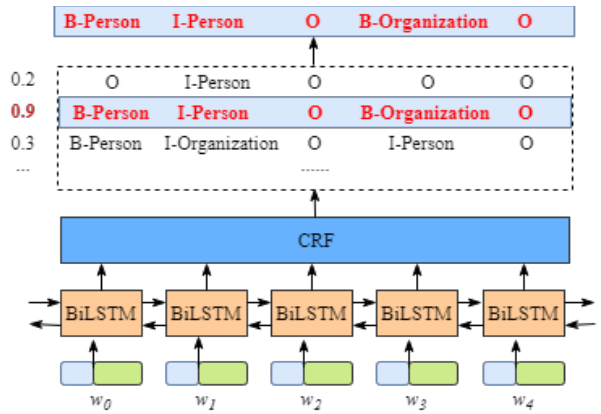


Figure 13. The BiLSTM-CRF architecture. CRF layer is stacked on top of the BiLSTM layer to learn the label sequence constraints. Source: ("CRF Layer on the Top of BiLSTM - 1," n.d.)

In the BiLSTM architecture, the BiLSTM layer outputs scores for each label. However, given that the BiLSTM layers does not capture label information of predecessors and successors, the ordering of the IOB 'pre-labels' are not learnt, where the first label for a multi-token entity must always be 'B' followed by an 'I', and a single-token entity must always be followed by another 'B' or an 'O'. This results in inaccurate predictions. In a two-token chemical entity example, the correct ordering would be "B-chemical, I-chemical", however the BiLSTM layer may predict "I-chemical, B-chemical".

To learn these constraints, a CRF layer is stacked on the output of the BiLSTM layer (Figure 13). The CRF layer has a loss function that is based on the emission scores from the BiLSTM output (the scores predicted for each label) and the transition score, which is derived from a learnt transition matrix that defines the IOB constraints. The loss function is the proportion of the score of the real sequence, defined as a path ($P_{true\ path}$), to the sum of the scores of all possible paths (Equation 15)

$$\frac{P_{true\ path}}{P_{total\ paths}} = \frac{P_{true\ path}}{\sum_{i=1}^{N} P_i} \qquad \text{Equation 15}$$

During learning, this proportion increases gradually, minimizing the log loss function (Equation 16), where the score of a path (P_i) is defined by the emission score and transition score (Equation 17).

$$-log\left(\frac{P_{true\ path}}{\sum_{i=1}^{N} P_i}\right) = -\left(log\left(P_{true\ path}\right) - log\left(\sum_{i=1}^{N} P_i\right)\right) \qquad \text{Equation 16}$$

$$-\left(\sum_{i=1}^{N} x_{iy_i} + \sum_{i=1}^{N-1} t_{y_i y_{i+1}} - log\left(\sum_{i=1}^{N} P_i\right)\right) \qquad \text{Equation 17}$$

The BiLSTM-CRF architecture (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016) is the current state-of-the-art for sequence-tagging tasks, particularly named entity recognition, and is benchmarked in the current work for bioentity named entity recognition.

2.4 Word Representations

Quantitative data to be used in machine learning tasks is compiled in a [nsamples x nfeatures] matrix to be fed into a model for training and testing; whether it's a classification (e.g. SVM

or logistic regression) or regression problem. This also applies to neural network models. Descriptive/free-text data is not different in this regard, and therefore requires a way to compile and represent the data into a numerical matrix.

There are three main approaches to do this: i) vocabulary-based; ii) syntax feature engineering; iii) distributional representations.

Approach One: In the first naïve approach, each token is considered a feature in itself, and therefore the dimensions of the matrix represents the number of unique words in the data (the vocabulary of the text). This is a very simple approach that may be sufficient for simple tasks where distinct keywords are commonly used and have good predictive capacity. However, the major limitations with this approach include:

- **Tokens are independent and equally distant from each other**. The tokens "dog" and "dogs" are treated equally different as the token "car" is to "dog".
- **Generates a large and sparse matrix**.

Approach Two: The second approach is to represent tokens by a set of features. This requires extensive feature extraction and engineering and has been researched for decades. Traditional features used include: orthographic, lemmas, stems and morphological features such as capitalization, a count of the uppercase characters and digits, suffixes and prefixes, character n-grams, and word shape patterns (Campos, Matos, & Oliveira, 2013b) (Figure 14). In terms of matrix compilation, each word is represented by a set of features that may or may not be shared by other tokens. A feature may be binary i.e. present/absent e.g. has upper case chars or not, or integer e.g. how many characters a token has.

Base	Token
Orthographic	Capitalization
	Counting
	Symbols
Linguistic	Lemma
	POS
	Chunk
	Dependency Parsing
Morphological	Char n-grams
	Suffix
	Prefix
	Word Shape
Lexicons	Gene/Protein
	Trigger Words
Local Context	Conjunctions

CAPITALIZATION	
Feature	Example
InitCap	Kappa
EndCap	KappaB
AllCaps	DAV
Lowercase	kappa
LowAndCap	RalGDS

COUNTING	
Feature	Example
SingleCap	U
TwoCap	US
ThreeCap	USS
MoreCap	USSR
SingleDigit	1
TwoDigit	12
ThreeDigit	123
MoreDigit	1234

SYMBOLS	
Feature	Example
Dash	-
BackSlash	/
OpenSquare	[
CloseSquare	]
OpenParen	(
CloseParen	)
Colon	:
SemiColon	;
Comma	,
Dot	.
Apostrophe	'
QuotationMa	"
Percent	%
Star	*
Equal	=
Plus	+

Figure 14. Feature sets typically engineered and extracted to represent biomedical words. Adapted from (Campos, Matos, & Oliveira, 2013a).

Feature extraction is often still important for achieving state-of-the-art performances (and is discussed in detail in (Section 2.7.1) in the context of named entity recognition), however is often not the main part of the research and is only done to provide auxiliary information. In addition to the major highly time-consuming limitation, semantics may not be represented well by these engineered features. For example, tokens “Good” and “Poor” both have the adjective (ADJ) POS tag, are four characters long, have one upper-case character and three lower case characters. Therefore, without more extensive features (and excluding context), these tokens share a very similar feature space and are therefore treated very similarly despite meaning the opposite.

In biomedical NLP, where entities are rich in features and entity nomenclature such as chemicals and genes follow nomenclature standards, feature extraction has enabled good results for tasks such as named entity recognition.

Approach Three: Away from manually extracting independent features from tokens, distributional representations provide a more continuous vector representation, where a token

is represented by a float vector of n dimensions. How this vector is computed depends on the algorithm used, but one of the basic approaches behind these representations is that words can be represented by the context they appear in. Referring to the original "dog" and "dogs" example, whereas previously these were completely different tokens, as these tokens are commonly used with a similar context (e.g. co-occurring with tokens such as barking, bark, walk, leash, collar etc), their vectors will be close to each other in the space. These representations encode many linguistic regularities and patterns; information that is not captured by traditional approaches.

Since their introduction, word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) and fastText (Bojanowski, Grave, Joulin, & Mikolov, 2017a; Mikolov, Grave, Bojanowski, Puhrsch, & Joulin, 2017) have been a popular choice of word embeddings and are benchmarked and optimized in this work for biomedical representations.

2.4.1 Word2vec models

The Skip-Gram (SG) and Continuous Bag of Words (CBOW) models are two popular architectures for the word2vec model introduced by (Mikolov et al., 2013). Starting with a sequence of words $\{w_1, w_2, w_3, \ldots, w_T\}$ in a vocabulary of size W, the objective is to learn vectorial representations for each word w (Bojanowski et al., 2017a; Mikolov et al., 2013). The objective of the SG model is to learn a vector representation for a given word (w_t) that predicts surrounding/context words (w_c) by maximizing the log likelihood (Equation 18). Inversely, in the CBOW model, the log likelihood of predicting the word w_t, given the context words (w_c), is maximized (Bojanowski et al., 2017a; Mikolov et al., 2013). This is shown figuratively in Figure 15.

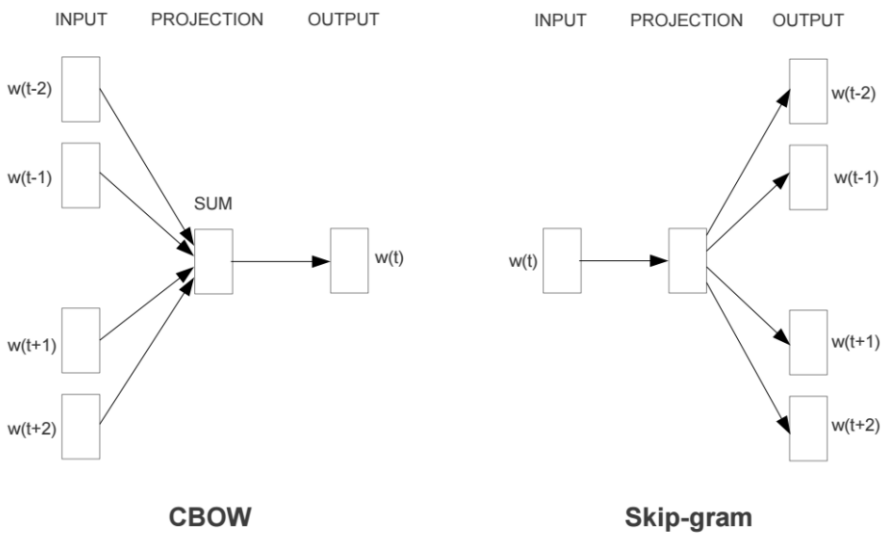


Figure 15. Graphic representation of the continuous bag-of-words (CBOW) and Skip-gram (SG) distributed word vector representation models. In the CBOW architecture, the probability of predicting a word w_t given the context words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ is maximized. In the Skip-gram architecture, given a word w_t, the probability of predicting the surrounding words $w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}$ is maximized. Source: (Mikolov et al., 2013).

$$\sum_{t=1}^{T} \sum_{c \in C_t} log\ p(w_c \mid w_t) \qquad \text{Equation 18}$$

C_t is the set of indices of context words surrounding word w_t, c is the index for a context word, and w_c is the word vector for the word at index c. Therefore, the word vectors parametrize the probability of observing a context word w_c given w_t (SG) (Bojanowski et al., 2017a), and vice-versa for the CBOW.

Considering a scoring function $s(w_t, w_c)$ which maps (word, context) pairs to scores $\mathbb{R}$, the softmax function can be used to define such probability (Equation 19).

$$p(w_c \mid w_t) = \frac{e^{s(w_t, w_c)}}{\sum_{j=1}^{W} e^{s(w_t, j)}} \qquad \text{Equation 19}$$

However, softmax is unable to predict all the context words; only one context word w_c can be predicted from a given word w_t. Framing this problem as a set of independent binary classification tasks changes the goal to independently predict the presence/absence of context

words (Bojanowski et al., 2017a). The negative log-likelihood with binary logistic loss for a given word is shown in Equation 20, where $N_{t,c}$ is a set of negative examples from the vocabulary (i.e. randomly chosen words which are not in context c).

$$log\left(1 + e^{-s(w_t, w_c)}\right) + \sum_{n \in N_{t,c}} log\left(1 + e^{s(w_t, n)}\right) \quad \text{Equation 20}$$

The obtained full objective function is shown in Equation 21.

$$\sum_{t=1}^{T} \left[log\left(1 + e^{-s(w_t, w_c)}\right) + \sum_{n \in N_{t,c}} log\left(1 + e^{s(w_t, n)}\right) \right] \quad \text{Equation 21}$$

Denoting the logistic loss function by Equation 22, the objective function can be rewritten as Equation 23.

$$l : x \mapsto log(1 + e^{-x}) \quad \text{Equation 22}$$

$$\sum_{t=1}^{T} \left[\sum_{c \in C_t} l\left(s(w_t, w_c)\right) + \sum_{n \in N_{t,c}} l\left(-s(w_t, n)\right) \right] \quad \text{Equation 23}$$

Parameterizing the scoring function with a vector u_{w_t} for w_t and v_{w_c} for the context word w_c, the score is computed as the scalar product between a given word and its corresponding context vectors (Equation 24) (Bojanowski et al., 2017a).

$$s(w_t, w_c) = {u_{w_t}}^T v_{w_c} \quad \text{Equation 24}$$

2.4.2 fastText models

fastText takes a similar approach to the word2vec SG model. The scoring function is modified such that the vector of the current word u_{w_t} is computed as a sum of the scalar product between the character n-grams' vectors z_g, for each of the words n-grams (g), and the context words (Equation 25) (Bojanowski et al., 2017a). This captures internal word structure.

$$s(w,c) = \sum_{g \in G_w} {z_g}^T v_c$$ Equation 25

In this part of our work, we train, optimize and benchmark word2vec and fastText representations for biomedical word representations.

2.4.3 Evaluating distributional word embeddings

2.4.3.1 Intrinsic evaluation

Intrinsic evaluation of word embeddings is based on assessing the similarity and relatedness of a set of concept pairs. This is performed by comparing such metrics to a pre-defined score. Starting with a curated list of concept pairs with an assigned score of similarity and/or relatedness (e.g. ranging from 0-1), the cosine similarity is calculated from the word embeddings between the same concept pairs and then correlated with the score.

Spearman correlation is a more appropriate choice for correlation (as opposed to Pearson correlation) when the ranges of the scores are not in the 0-1 range. In Spearman correlation, scores are first converted to ranks and therefore it's only the order of the scores that matter rather than the absolute values.

Distinguishing between similarity and relatedness informally is best done with an example: "car" and "truck" are similar concepts as they are both automotive vehicles, whereas "car" and "road" are related. Therefore, similar concepts are a specific subset of related concepts.

In the context of graphs, similarity and relatedness can be defined as (Pedersen, Pakhomov, McInnes, & Liu, n.d.):

- similar concepts share ancestors through “is_a” relationships
- related concepts share ancestors through any other (non is_a) relationship.

Having defined these metrics in graph terms, multiple measures can be used to define and calculate similarity/relatedness. Path-based methods are the simplest and commonly used measures.

Similarity measures

The general intuition for path-based methods is that given a pair of concepts, the closer the concepts are in a graph, the shorter the path and therefore the most similar they are. Similarity can therefore be defined by the inverse of the shortest path between concepts (Equation 26).

$$similarity(a,b) = \frac{1}{short_path(a,b)} \qquad \text{Equation 26}$$

However, a potential limitation with such naïve measure is that paths are assigned equal weights. In a hierarchical graph, depth defines specificity: the deeper we go in the graph, the more specific the nodes are. This can be improved by considering depth in addition to path (Wu & Palmer, 1994) (Equation 27).

$$similarity(a,b) = \frac{2\,.depth(common_ancestor(a,b))}{depth(a) + depth(b)} \qquad \text{Equation 27}$$

In Equation 27, the common ancestor is the closest ancestor shared by both concepts and depth is the length of the path starting from the root node.

A further modification to this makes use of additional information for each node, namely the term frequencies. Concepts which occur frequently are likely to be less specific than terms occurring less frequently. The same equation as above is used, however, depth is replaced by the information content calculated as the negative log form of the proportion of term frequencies added to the inherited frequency, to the total frequency (Equation 28) (Lin, 1998).

$$Information\ Content = -\log\left(\frac{tf + if}{N}\right) \qquad \text{Equation 28}$$

Relatedness measures

Based on the graph definition, relatedness can also be calculated with path-based measures, given the is-a relationships are excluded when calculating path length. Alternatively, relatedness can be derived from concept definitions, with approaches including: Lesk (Lesk, 1986), adapted Lesk (Banerjee & Pedersen, 2002), and gloss vector (Patwardhan, 2006).

The general intuition behind definition-based approaches is that concepts which share similar definitions are more related and therefore overlap is computed between definitions. Being definition-based, a graph connecting concepts is not required. This is not only beneficial when a graph is not available but also when nodes are disjoint – path-based metrics cannot be computed for disjoint nodes. However, there are a number of limitations with using definitions:

- Definitions tend to be concise/short
- Definitions may use different terminologies and synonyms, therefore exact overlap may be difficult (unless a lexicon is available and concepts are normalized to it)

2.4.3.2 Extrinsic evaluation

Extrinsic evaluation involves performing a NLP task and assessing how different word representations effect the performance. NLP tasks which have commonly been used to assess embeddings' performance are:

- Named entity recognition
- Event extraction
- Text Classification
- Text Summarization
- Analogy detection/resolution

In our work investigating different word representations for biomedical text, we perform both intrinsic and extrinsic evaluation.

2.5 Biomedical Abbreviation Resolution

Biomedical text includes a large number of abbreviated word forms. The lack of standardization of such abbreviations and high rate of new abbreviations requires mapping shortened forms to their definition (or long form).

The Schwartz algorithm (Schwartz & Hearst, 2003) is a popular, fast, simple and readily-accessible abbreviation resolution algorithm that improves on previous algorithms utilizing pattern-based approaches and linear regression, achieving 82% recall and 96% precision (J. T. Chang, Schutze, & Altman, 2002; J. Pustejovsky, Castaño, Cochran, Kotecki, & Morrell, 2001). The algorithm is subdivided into 2 tasks: extracting the <short-form, long-form> candidate pairs, and subsequently identifying the correct long form from the list of candidates (Schwartz & Hearst, 2003). Candidate pairs considered by the algorithm are of the forms: (i)

<long form> (<short form>); (ii) <short form> (<long form>). If the string enclosed within parentheses is longer than 2 words, form 2 is assumed, therefore the short form is looked for to the left of the left parenthesis.

Candidate short forms are considered if they meet the criteria:

i. consist of maximum 2 words;
ii. length is between 2-10 characters;
iii. at least one character is a letter;
iv. first character is alphanumeric;

Candidate long forms are terms which meet the criteria:

(i) appear in the same sentence as the short form;
(ii) must be adjacent to the short form;
(iii) consist of less words than min $(n_{chars} + 5,\ 2\ .n_{chars})$ where n_{chars} is the number of characters of the short form;

With a list of short and long term candidate pairs, the algorithm utilizes two indices starting at the end of each form: short_form_idx and long_form_idx. The long form index is decremented until the short form character at the short_form_idx matches. If long_form_idx reaches 0 before the short_form_idx, this is considered as "no match". When a character match is found, the short_form_idx is decremented and the long_form_idx is set to the matching index – 1. An additional constraint is applied to the first character of the short form, where this can only match the first character of a word in the long form (Schwartz & Hearst, 2003).

This algorithm for abbreviation resolution is integrated and extended as part of the pre-processing module in the proposed natural language processing pipeline.

2.6 String Matching Algorithms and Data Structures

String matching is a simple yet effective approach for identifying strings of text from a collection of terms. In its simplest form, the collection of terms can be compiled in a dictionary and regular expressions (Regex) is used to search for string matches. For each entry in the dictionary, this approach requires O(MxN) time complexity for a document of size N (characters) and a dictionary of M keywords – which can be considered slow at scale (Singh, 2017).

Matching strings to a collection of terms can be achieved in a number of approaches. The "longest string match" is a common approach where tokens are sequentially considered for matching and only the term with the longest number of tokens is considered. For example, given the phrase "human colon cancer", and a dictionary {"human", "colon", "cancer", "human colon", "human colon cancer"}, starting from the first token, this would match "human", which is also the first token of "human colon" and "human colon cancer". However, in the "longest string match" approach, only "human colon cancer" is considered a match. This approach is similar to the strategy used by ElasticSearch to match dictionary strings; a technology that is used by NLP pipelines such as PolySearch2 (Y. Liu, Liang, & Wishart, 2015).

In other approaches, each token can also be considered as the initial token even if part of a match already. Therefore, for the phrase "human colon cancer", "human", "human colon", "colon", "cancer", "human colon cancer" and "colon cancer" can all be considered as matches.

This nested nature of matching is best represented by the trie data structure; a tree-based data structure where nodes represent word characters and traversal down a branch retrieves a full term (Figure 16). All nodes are connected to an empty root node and each node is an array of pointers to child nodes.

A node in a trie can have zero (in which case ending the string), to multiple child nodes. In case where a node has a single child node for a number of tree layers, this can be considered inefficient as 1-edge length traversal only provides a single result. By merging subsequent nodes with single child nodes into one node, this saves on space and traversal time. Such modified tree is referred to as a radix tree (Figure 16).

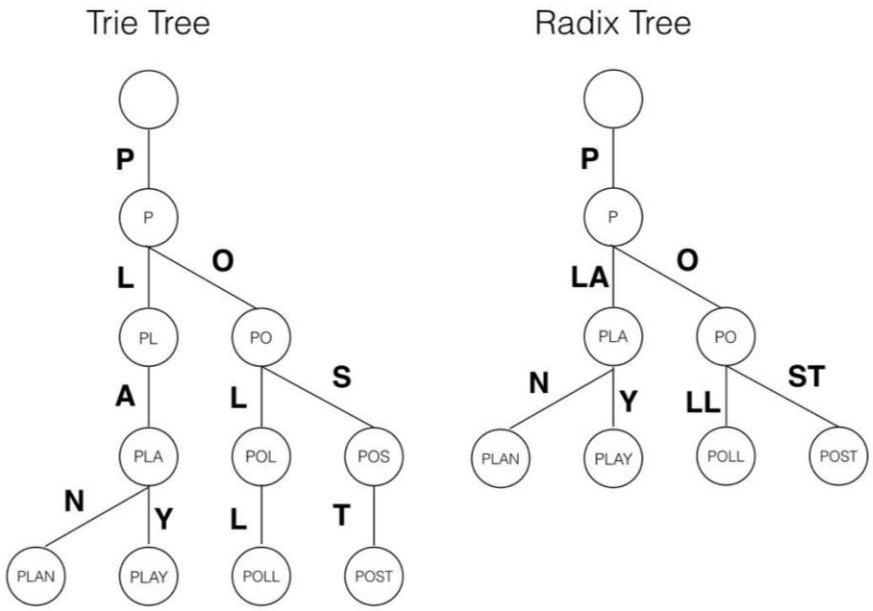


Figure 16. Trie and radix tree data structures. Trie tree (left) and radix tree (right) for the same set of keywords: {plan, play, poll, post}, where radix tree groups nodes with a single descendant; saving on memory and space. Source: ("Radix tree - Swift Data Structure and Algorithms [Book]," n.d.)

In this work, we utilize and implement both a trie-based dictionary matching algorithm and longest-string match algorithm to perform dictionary-based named entity recognition.

2.7 Named Entity Recognition

Named entity recognition (NER) is a subtask of information extraction that identifies named entities in a string and classifies such mentions into pre-defined classes. This process can therefore be subdivided into: (i) detection of entities as a segmentation problem where boundaries are identified; and (ii) classification problem where an ontology is assigned to the identified name. In generic models, typical classes include: people names, organizations, and time (Figure 17), whereas in the biomedical field this can include classes such as: genes, diseases, chemicals, and metabolites.

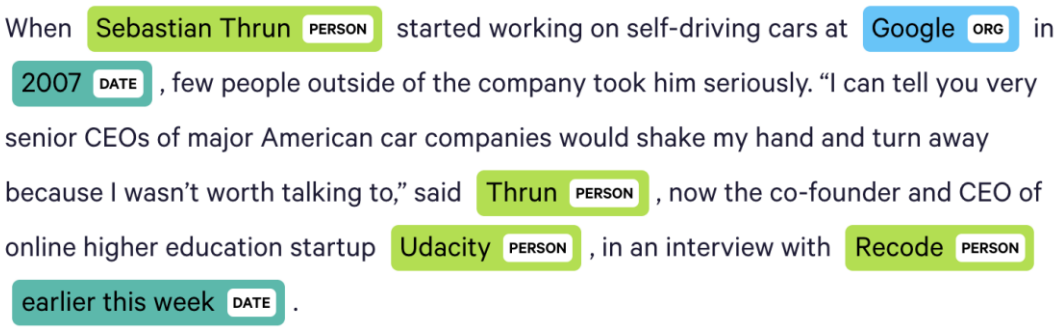


Figure 17. Text annotated for "person" (PERSON), "organization" (ORG), and "date" (DATE) by a named entity recognition model. Source: spaCy as visualized by displaCy Named Entity Visualizer.

In its simplest form, NER can be performed by string matching to a curated list/dictionary (Section 2.6). In machine learning approaches, text is treated as sequential information and a number of sequential methods have been utilized, such as: HMMs (G. Zhou & Su, 2002), CRFs (Finkel, Grenager, & Manning, 2005) and RNNs (Section 2.2-2.3), with state-of-the-art architectures such as BiLSTM-CRF (Lample et al., 2016) (Section 2.3.2).

In this work, we investigate the CRFs and BiLSTM-CRFs architectures for biomedical named entity recognition.

2.7.1 Feature Engineering

Specifically in the biomedical domain, with training corpora (textual datasets used for training machine learning models) and challenges such as: GENIA (J.-D. Kim, Ohta, Tateisi, & Tsujii, 2003), CHEMDNER (Krallinger, Rabal, et al., 2015), and BioCreative II GeneMention (Smith et al., 2008), supervised machine learning methods have been mostly used and researched for NER. This has led to extensive feature engineering research, categorized by (Alshaikhdeeb & Ahmad, 2016) as: (i) morphological; (ii) lexical; (iii) dictionary-based; and (iv) distance-based (Figure 18). The use of these different features in biomedical NER research is summarized in Table 2.

Morphological features provide word-level morphological representations and can be subdivided into: (i) numeric features; (ii) Boolean features; and (iii) nominal features. Numeric features such as word length and frequency are represented by integer values, whereas Boolean features such as "contains uppercase characters" or "contains punctuation" are binary features representing presence or absence. Finally, nominal features are categorical features such as Part-Of-Speech (POS) tags, indicating whether a token is a verb, noun, adjective or otherwise.

Lexical features provide syntactic word-level information by capturing grammar. POS tagging is one popular example which provides such information, with biomedical entities such as genes, diseases, and chemicals commonly having the noun syntactic tag.

So far, none of the described features capture word semantics. Tokens such as "alpha", "beta", and "kappa" may be words neighboring entities, however their semantics (i.e. that they are all Greek words) is not captured. Contrastingly, tokens such as "Rab", "Alu", and "Gag" representing proteins can be treated similarly to "Phe", "Arg", and "Cys" as they share similar

feature sets, despite being abbreviations for amino acids (Settles, 2004). By indicating whether tokens are present in a curated dictionary (dictionary-based features), this improves the identification step as a matched word is likely to be an entity, as well as the classification step as a specific ontology is derived from the dictionary used to match the tokens. Additional dictionary-based features include: trivial names for chemicals, modifiers, family of entity (e.g. “alcohol” for “ethanol” and “methanol”), and abbreviations (e.g. “Ca” for “Calcium”) (Alshaikhdeeb & Ahmad, 2016).

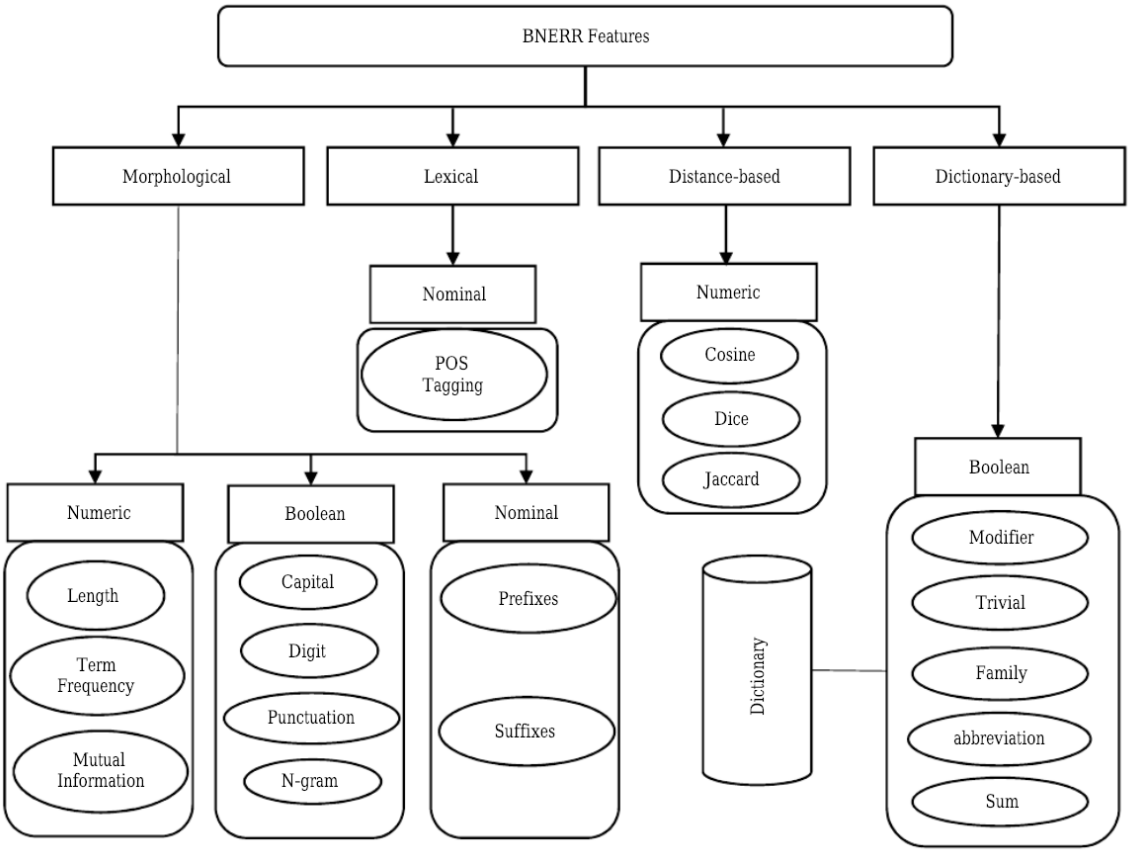


Figure 18. Categorization of various features used in biomedical named entity recognition. Source: (Alshaikhdeeb & Ahmad, 2016)

As biomedical entities can be highly variant but follow a similar structure, similarity/distance-based features have also been used. To generalize this further, a “word class” feature can also be derived where alphanumeric characters are replaced with “A” if uppercase and “a” if lowercase and digits replaced with “0” while all other characters are replaced with “_”. For

example, for the tokens “IL5” and “SH3”, these are both converted to AA0, whereas “F-actin” and “T-cells” are converted to “A_aaaaa” (Settles, 2004).

Table 2. Categories of features used in biomedical named entity recognition literature. Adapted from: (Alshaikhdeeb & Ahmad, 2016).

Authors	**Morphological**			**Lexical**	**Distance-based**	**Dictionary-based**
	Boolean	**Nominal**	**Numeric**			
(Friedrich, Revillion, Hofmann, & Fluck, n.d.)	x			x		x
(Degtyarenko et al., 2008)						x
(Corbett & Copestake, 2008)				x		x
(Klinger, Kolářik, Fluck, Hofmann-Apitius, & Friedrich, 2008)	x					
(de Matos et al., 2010)		x				x
(Rocktäschel, Weidlich, & Leser, 2012)			x	x		x
(Lamurias, Grego, & Couto, 2013)						x
(Alharbi & Tiun, 2015)		x		x	x	
(Batista-Navarro, Rak, & Ananiadou, 2015)	x			x		
(Usié, Cruz, Comas, Solsona, & Alves, 2015)			x			x
(Yaoyun Zhang et al., 2016)						x
(Hashim Mohammed & Omar, 2016)		x				
(Bhasuran, Murugesan, Abdulkadhar, & Natarajan, 2016)	x					

Moving towards distributional word representations (Section 2.4), feature engineering has been given an auxiliary role, with some engineered features used to boost benchmark performance.

2.7.2 Architectures and Performance

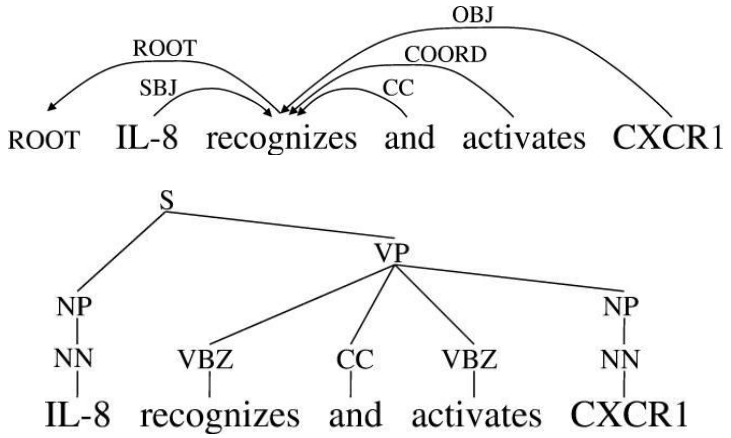
In terms of biomedical NER architectures and absolute performances, previous work has utilized HMMs (Ponomareva, Rosso, Pla, & Molina, n.d.; G. Zhou & Su, 2002), SVM (K.-J. Lee, Hwang, Kim, & Rim, 2004; G. Zhou, Zhang, Su, Shen, & Tan, 2004), CRF (Campos et al., 2013b; Settles, 2005; H.-J. Song, Jo, Park, Kim, & Kim, 2018), and CNN (Yao, Liu, Liu, Li, & Waqas Anwar, 2015), with the current state-of-the-art being achieved by BiLSTM-CRF. BiLSTM-CRF was reported to achieve 83.14% for chemicals and 84.68% for disease on the BC5 CDR corpus, 84.41% for diseases on the NCBI corpus, chemical, and gene/protein (Dang, Le, Nguyen, & Vu, 2018), 75.87% on JNLPBA (Gridach, 2017), 89.46% for BioCreative II GM (Gridach, 2017). This is in comparison with traditional methods such as CRF achieving 69.9% on GENIA (Settles, 2005), 72.82% on JNLPBA (H.-J. Song et al., 2018) or 65.7% with HMM on JNLPBA (Ponomareva et al., n.d.).

These NER systems have provided state-of-the-art performances on benchmark datasets, however their translational utility in a scalable and diverse pipeline may be limited. In this work we investigate this by determining model generalizability across different sources, and test the role of training data quantity on performance.

2.8 Association Extraction

Association or relationship extraction is an information extraction subtask aiming at identifying the occurrence of a relationship between entities (Shahab, 2017). Systems such as (Chen, Hripcsak, Xu, Markatou, & Friedman, 2008) relied on co-occurrence to identify this, however rule-based and machine learning systems have been heavily explored since.

Machine-learning approaches are popularly based on classification methods, where given a set of entities, a classifier identifies if an association exists or not (in a binary scenario) or classifies the association into more fine-grained relationship types, such as: “is a”, “phosphorylates”, “causes”, “effects”, and “treats”. In addition to the tokens themselves, syntactic and semantic structures have been researched and used to capture grammatical relations between different words (Miyao, Sagae, Saetre, Matsuzaki, & Tsujii, 2009). Some examples include: dependency parsing (e.g. CoNLL dependency tree), phrase structure parsing (e.g. PennTreebank phrase structure tree and Stanford dependencies) and deep parsing for syntactic and semantic structures (Miyao et al., 2009) (Figure 19). Several approaches have been proposed to utilize these structures in relation extraction methods, including: all-path graph kernel (Airola et al., 2008), and shortest dependency path kernels (S. Kim, Yoon, & Yang, 2008).



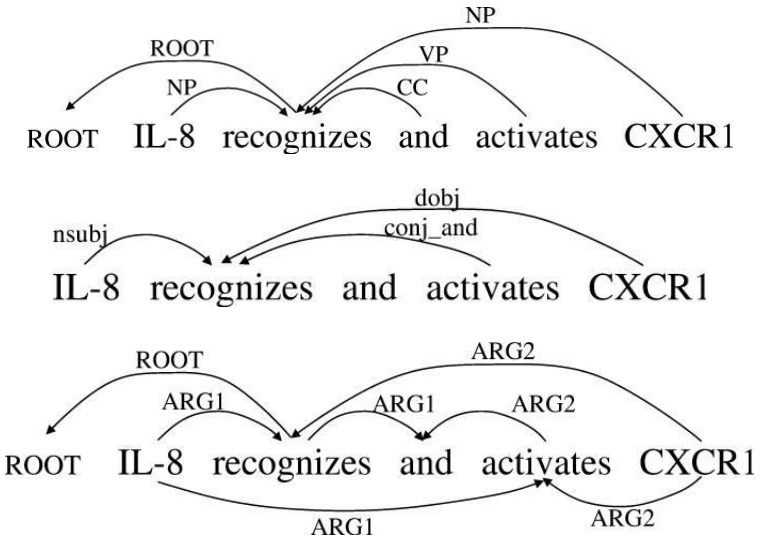


Figure 19. Different syntactic structures utilized in relation extraction approaches. A) CoNLL dependency tree; B) PennTreeBank phrase structure tree; C) head dependencies; D) Stanford dependencies; E) predicate-argument structure. Source: (Miyao et al., 2009)

Beyond kernel methods, maximum entropy and conditional random fields have also been used in gene-disease and disease-treatment relation extraction (Bundschus, Dejori, Stetter, Tresp, & Kriegel, 2008; Chun et al., 2006), while a number of rule-based approaches have been proposed for gene-protein and protein-protein relation extraction obtaining up to 66.05% accuracy for AIMed (Raja, Subramani, & Natarajan, 2013; Saric, Jensen, Ouzounova, Rojas, & Bork, 2006), and (Fundel, Küffner, & Zimmer, 2007) reporting up to 80% recall and precision on a custom corpus.

More recently, neural architectures have been applied to tackle this task by automatically learning features. Recurrent neural networks and convolutional neural networks were applied to protein-protein and drug-drug interaction datasets (separately) by (Yijia Zhang et al., 2018), achieving 61.7% for AIMed, 64.8% for BioInfer, 78.2% for IEPA, 75.6% for HPRD50, and 85.2% for LLL. This has outperformed a number of approaches (Airola et al., 2008; Y.-C. Chang, Chu, Su, Chen, & Hsu, 2016; S. Kim, Yoon, Yang, & Park, 2010; Miwa, Saetre, Miyao,

& Tsujii, 2009; Y. Peng, Gupta, Wu, & Shanker, 2015; Yijia Zhang, Lin, Yang, & Li, 2011; Zhao, Yang, Lin, Wang, & Gao, 2016).

These association extraction approaches, particularly advanced machine learning-based methods, provide state-of-the-art performances on benchmark datasets, however with the limited training data, such approaches are not generalizable to be applied to a scalable and diverse pipeline. Similarly, rule-based approaches require extensive crafting of domain-specific rules, rendering them time-consuming and non-generalizable. This is even more amplified in works such as (Saric et al., 2006), where a domain-specific POS tagger is trained, which is in turn used to define the rules. In this work, we attempt to find a balance between simple rule-based methods and general machine learning models to achieve an association extraction approach that can be integrated as part of the proposed HASKEE pipeline.

2.9 Biomedical NLP tools

In Sections 2.1-2.8, the fundamental, technical developments and latest state of the art developments of the individual tasks required in a NLP pipeline such as named entity recognition, abbreviation resolution, and association extraction, have been introduced. While this information is critical in understanding how methods utilized in the field of NLP, such tasks are often studied in isolation on benchmark datasets and not translated or applied. A more in-depth review of the literature for each of these tasks is given in the respective chapter.

For end-to-end information extraction however, such as the aim of this thesis, these individual tasks need to be orchestrated in form of a pipeline. This has been briefly outlined in Section 1.1, Figure 2. Developments of usable end-to-end platforms are much less commonly

researched, with a handful of platforms such as: PolySearch (Cheng et al., 2008; Liu, Liang, & Wishart, 2015), ArrowSmith (Smalheiser, Torvik, & Zhou, 2009), and BeFree (Àlex Bravo et al., 2015), and older ones such as: iHOP (Hoffmann & Valencia, 2004), BioRAT (Corney, Buxton, Langdon, & Jones, 2004), BITOLA ("Bitola (Biomedical Discovery Support System)," 2008), and LitLinker (Yetisgen-Yildiz & Pratt, 2006). Collectively, these platforms have great use-cases but also have a number of shortcomings, discussed below. The gap in the current tools is aimed to be filled with the development of the proposed HASKEE pipeline.

2.9.1 PolySearch and PolySearch2

Originally developed in 2008, PolySearch was a platform for querying direct associations between human diseases, genes, mutations, drugs and metabolites (Cheng et al., 2008), and therefore addressed the open-discovery query type: "Given entity X, find all associated entities Y". For PubMed articles, the PolySearch workflow involved computing synonyms for the queried entity from several sources of thesauruses (including Entrez Gene (Maglott, Ostell, Pruitt, & Tatusova, 2005), HGNC (Yates et al., 2017), HPRD (Fundel et al., 2007), UMLS (Humphreys, Lindberg, Schoolman, & Barnett, 1998), OMIM (Hamosh et al., 2005), KEGG (Kanehisa et al., 2016), BioCarta (http://www.biocarta.com), and LitMiner (Yetisgen-Yildiz & Pratt, 2006)).

Following the named entity recognition step, Boolean operators for all pairwise combinations are computed and passed to the NCBI search E-utilities API which retrieves articles containing any of the pairs queried. PolySearch then scores the results by the frequency of occurrence, where the higher the times an entity pair is found in different text sources, the higher the significance. Subsequently, sentences are ranked by relevancy, where the highest relevancy (R1/2) is assigned to sentences containing both entities mentioned in the sentence, as well as

association words which are predefined by PolySearch (specifically for gene/protein pairs: “complex”, “complexes”, “inhibitor”, “inhibitors”, “interaction”, “interactions”). On the other hand, a lower rank (R3) is assigned to sentences containing the entities but no association words, and the lowest score is assigned to sentences with only one term (R4). Rank is assigned by giving a score in the score set {1, 5, 25, 50}, where a score of 1 is the best score for R4 sentence types (Cheng et al., 2008).

In the named entity recognition task based on dictionary matching, PolySearch is reported to achieve 87.6% F-score for recognizing genes and proteins. For association extraction based on matching with a lexicon of association terms, an accuracy of 69.2-80.8% is claimed for extracting protein-protein interaction for 5 different proteins, and up to 78.5% for gene-disease extraction for 10 different diseases (Cheng et al., 2008).

The limited coverage of entities in PolySearch was recognized and improved on by PolySearch2 (Liu, Liang, & Wishart, 2015). The dictionary sources have been extended to cover a wider range of entity types (totalling to 20 types), including: toxins, food metabolites, gene ontologies, MeSH terms/compounds and ICD-10 medical codes (Liu, Liang, & Wishart, 2015). In addition to this, the technical stack and algorithms used for association extraction/scoring was improved by utilizing an ElasticSearch engine which natively enabled refining the scoring by introducing a “tightness measure” that scores sentences which mention the entities of interest higher the closer they are to each other, and penalizes more distant cooccurrences (Liu, Liang, & Wishart, 2015a). Additionally, the list of association terms was extended to almost 30,000 terms by adding terms like “catalyzes”, “phosphorylates”, etc. (Liu, Liang, & Wishart, 2015).

This extension has led to improved named entity recognition as well as association extraction, with disease/gene, drug/gene, protein/protein and metabolite/gene associations predicted with 85-90% accuracy and drug/adverse effects, toxin/disease and toxin/adverse effects predicted with 77-79% accuracy (Liu, Liang, & Wishart, 2015).

2.9.2 Arrowsmith

In 1986, Swanson coined the phrase undiscovered public knowledge (Swanson, 1986), where he recognized the fragmentation of knowledge holds hidden links. Swanson argues that "novel"/undiscovered findings are indirectly incorporated in the current literature, and literature-based discovery can be tapped into with an A-B-C model. The A-B-C model describes how an article can contain the link A-B, while another describes the link B-C. In no publication is the A-C link described, however by inference we can hypothesize this. This model of association discovery was proven by Raynaud's disease linked with eicosapentaenoic acid in literature, and eicosapentaenoic acid linked with dietary fish oil. The indirect linkage was that dietary eicosapentaenoic acid can decrease blood viscosity in Reynaud's patients with high blood viscosity. Clinical studies supported the hypothesis that fish oil can be used to counteract the symptoms of Raynaud's disease (Swanson, 1986).

Arrowsmith was the first tool to bridge the gap in literature and find "meaningful links between two disparate sets of articles" (Smalheiser, Torvik, & Zhou, 2009). Unlike PolySearch which tackles the query type: "Given X find all Y associated with it", Arrowsmith can be used to retrieve indirect support from scientific literature for novel hypotheses (Smalheiser, Torvik, & Zhou, 2009).

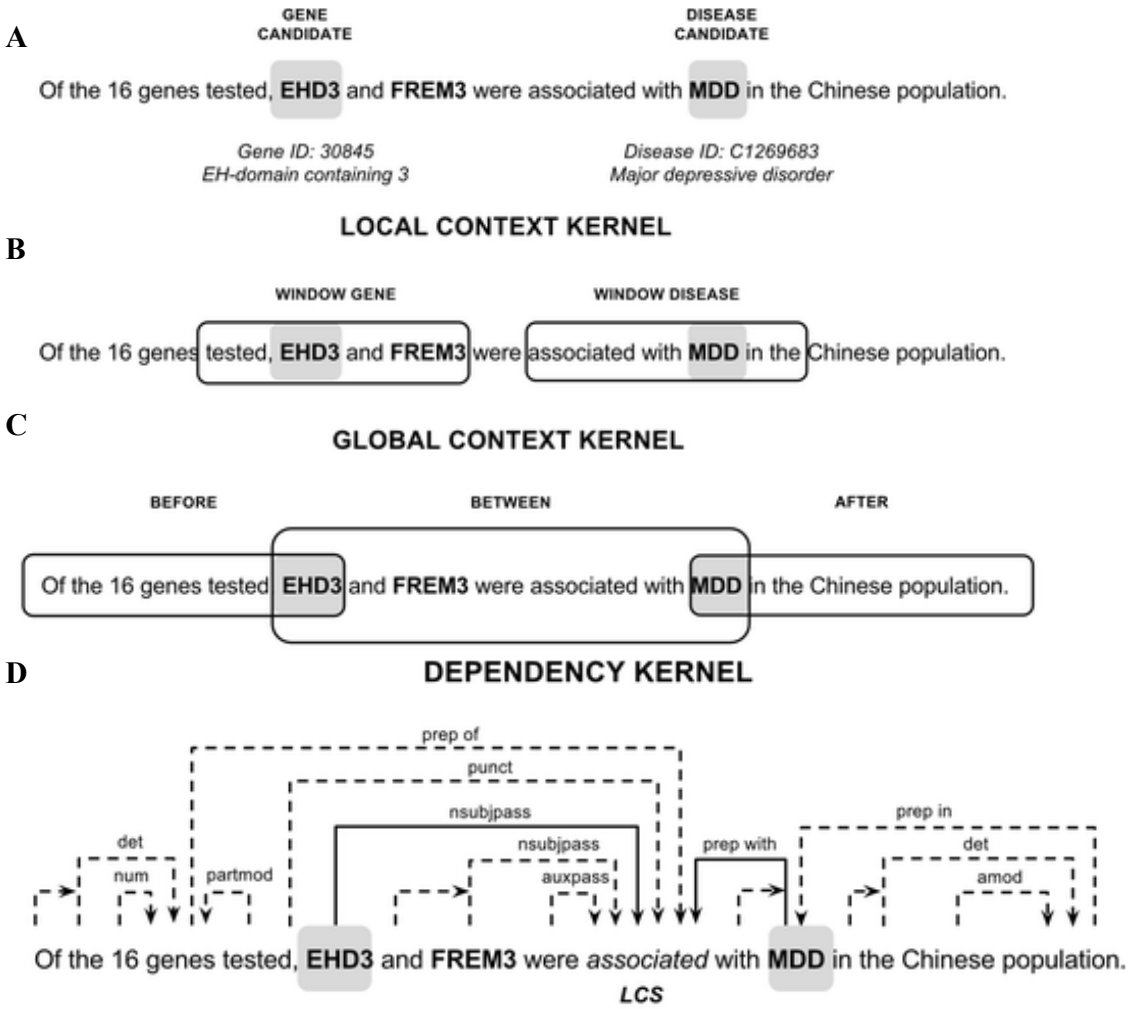
The methodology by which this is done is rather simple. Given 2 entities (A and C), the article sets describing each individually are retrieved, and words and phrases in common in the article titles (referred to as phrases B), are extracted. These B-phrases can be suggestive of an indirect link between the 2 disparate queried entities. This approach has been reviewed extensively in Smalheiser (2017).

2.9.3 BeFree

Similar to PolySearch, BeFree is a text mining system that focuses on the extraction of gene-disease, drug-disease and drug-targets associations (Àlex Bravo et al., 2015). This is executed in a 2 step process: (i) named entity recognition of the longest string match by soft/fuzzy-matching, followed by filtering of ambiguous cases, and (ii) relation extraction. While the first version performed relation extraction similar to PolySearch (where an association is defined by cooccurrence of entities and scored by the frequency of the co-occurring pair) (Àlex Bravo et al., 2015), later version makes use of supervised machine learning to exploit more dynamic syntactic and semantic information. A shallow linguistic kernel and a dependency kernel were trained on annotated data and applied to PubMed for the extraction of relationships (Figure 20).

These advanced approaches are reported to achieve up to 79.3% F-score for drug-disease associations, up to 84.6% for gene-disease associations and up to 83.3% F-score for target-drug associations. Comparing these metrics to PolySearch performance at face-value is unfair, as benchmark data used is different. Authors of BeFree however assessed their algorithms on several benchmarks, claiming more balanced metrics between precision and recall (Àlex Bravo et al., 2015).

While BeFree provides a more advanced, comprehensive and complete approach that mitigates the limitations of PolySearch and PolySearch2 (which depend a lexicon containing "association terms"), such an approach does not scale very well to several entity classes. In fact, one of the limitations of BeFree is its lack of coverage and versatility in the entity classes which is hard to extend without considerable amount of work creating training data that covers all the entity classes of interest for the machine learning models.



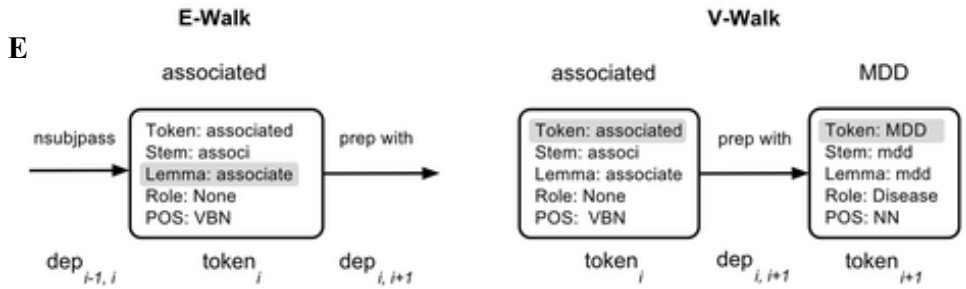


Figure 20. Local and global shallow linguistic kernels (B-C) and dependency kernel (D-E) devised by BeFree for association extraction, exemplified by the sentence (A) mentioning the gene EHD3 and disease MDD (Major Depressive Disorder). The local context kernel (B) exploits orthographic and shallow linguistic features such as POS tags, lemmas and stems for tokens within a window of each entity mention. The global context kernel (D) captures positional and sentence order information, represented by bi-trigrams. "associated" is identified as the least common subsume (LCS). Features considered in the dependency kernel include the token, stem, lemma, POS tag, and the role (disease or gene) (E). Adapted from Àlex Bravo et al. (2015).

2.9.4 Precompiled association databases

Beyond tools and platforms, the biomedical domain is very rich in compiled repositories of associations; from experimentally derived associations, to literature extracted associations. DisGeNET, for example, is a repository of over 600,000 gene-disease associations and over 200,000 variant-disease associations (Piñero et al., 2015; 2016).

Often, development of such repositories involves compiling associations from different sources into a single repository. In the case of DisGeNET, the repository integrates curated repositories, GWAS catalogues, and animal models. Additionally, it also includes literature-extracted associations from the BeFree repository (see Section 2.9.3).

Other databases of associations include: eDGAR (disease-gene associations) (Rindflesch et al., 2009) compiling associations from OMIM (Hamosh et al., 2005), HuVarBase (Ganesan et al., 2019) and ClinVar (Landrum et al., 2014), the Genetic Association Database (GAD) compiling gene-disease associations (Àlex Bravo, Piñero, Queralt-Rosinach, Rautschka, & Furlong, 2015), and STRING for protein-protein interactions (Szklarczyk et al., 2017) compiling curated

databases such as HPRD (Fundel et al., 2007), BioGRID (Stark et al., 2006), KEGG (Kanehisa & Goto, 2000), EcoCyc (Keseler et al., 2017).

2.9.5 Recognizing the gaps

As outlined above, current biomedical association tools are either online platforms that provide access to a predefined pipeline that computes biomedical associations (e.g. PolySearch, Arrowsmith, BeFree), or repositories of pre-compiled databases of associations (e.g. DisGeNET, and GAD). There are several limitations to such platforms/resources:

(i) these platforms are not open-sourced but rather compiled and hosted online, therefore users rely on the original authors to keep the system up to date;

(ii) different users/researchers have different use cases and having a single platform that fits all use cases is not feasible;

(iii) platforms/algorithms/repositories such as BeFree are limited to a small number of entity classes. Others, like Polysearch/PolySearch2, while they are versatile in terms of entity types and therefore can be used by a wide variety of researchers, are restricted to querying direct associations. On the other hand, Arrowsmith allows for inference of novel associations, however algorithmically is very primitive (not to mention outdated), and is based on perfect keyword matching.

In this work, we propose an open source generalizable pipeline rather than a compiled platform. The availability of a pipeline allows users to tailor it according to their needs, and add extra modules or improve on top of it.

Beyond usability, from an NLP task perspective, existing translational platforms generally perform an oversimplified 2-step process: (i) named entity recognition which accounts for synonyms for a variety of entity types; and (ii) association extraction and ranking. Collectively, such platforms lack one or all of the following:

(i) extraction of negative associations;

(ii) abbreviation resolution: e.g. normalizing BRCA to Breast Adenocarcinoma;

(iii) differentiation between novel claims and citation of a previous claim, and therefore heavily cited findings are currently highly ranked based purely on popularity.

With HASKEE, we propose and provide a pipeline implementation that in addition to the minimal essentials of named entity recognition and association extraction modules, it also incorporates new sub-modules that specifically tackle these limitations. Additionally, the pipeline not only provides the capability of executing open querying in the style of PolySearch2, but also enables the discovery of novel associations by inference as originally proposed by Arrowsmith. This provides a single yet versatile and customizable solution.

Chapter 3 – Developing and utilizing methods for the identification of biomedical associations from structured and unstructured data

3.1 Abstract

Diverse wet lab and bioinformatics methods can identify potential biomarkers and subsequently biomedical associations. With quantitative and structured data, univariate tests or machine learning-based classifiers can be used to identify condition/disease-specific biomarkers. For unstructured data, a manual systematic review is still a common practice in medicine. In this work we devise and/or apply methods that can be used to identify such associations by: (i) performing a systematic review for colorectal biomarkers for resistance to radiotherapy; (ii) developing a hierarchical classification system for bacterial species and cancer types/subtypes based on their mass spectral and genomic profiles, respectively; and (iii) identifying potential drugs that can be repurposed for cancer therapy using network propagation approaches and supervised machine learning approaches. Using the findings from the latter, we demonstrate the validity of the results obtained using the developed literature search pipeline.

3.2 Aims and Objectives

Aim(s):

- To devise and apply a variety of methods for the identification of candidate biomarkers (and therefore biomedical associations) that can subsequently make use of a literature search pipeline for validation.

Objective(s):

- To perform a manual systematic review of colorectal biomarkers for resistance to radiotherapy;
- To devise machine learning approaches for the hierarchical classification of bacterial species and cancer subtypes;
- To identify potential drugs that can be re-purposed for cancer therapy using network propagation approaches and supervised machine learning.

3.3 Introduction

Identification of biomarkers and biomedical linkages is a critical aspect of biomedical research leading to fundamental biological understanding and development of prognostic, diagnostic and therapeutic solutions.

In clinical microbiology, identification of bacteria is critical to assign the best treatment, as for every hour of treatment delay to sepsis, mortality rate is estimated to increase by 7.6% (Kumar et al., 2006). Current solutions require lengthy tests based on phenotypic characteristics or sequencing. This highlights the importance of developing methods that efficiently discriminate between different bacterial species and subsequently identifying bacterial species-specific biomarkers. Similarly, in cancer, biomarkers can provide early detection or identify the best treatment for a specific cancer type and subtype.

Computationally, identification of biomarkers can be achieved by feature selection. The use of feature selection methods has been an active field of research, however the significance of such features identified has been questioned (Haury, Gestraud, & Vert, 2011). As biomedical entities such as genes or metabolites are not in isolation but part of a complex network, a single

biomarker is unlikely to provide wide biological insights, whereas investigating pathway perturbations are more likely to give an idea of the whole picture. By performing pathway analyses, we can obtain a deeper biological understanding of biological systems.

Developing a treatment such as drugs is the ultimate goal, however this comes at a big cost of 2.5 billion dollars (Mullin, n.d.) and can take up to 16 years to reach the markets (Matthews, Hanison, & Nirmalan, 2016). By identifying novel purposes for existing drugs, this process can be sped up and made more efficient in terms of cost. Repositioning drugs can be performed by utilizing methods that take into account of the current knowledge of existing drugs and the complex network of bioentities such as genes. In this work, drugs are compared by the genomic network profiles effected.

From bacterial and cancer-related biomarker identification, to drug repositioning, these described approaches can make use of a natural language processing platform to 'validate' the findings across independent research or identify potential novel findings. Here some of these methods are developed and/or executed to demonstrate the output generated that can make use of such a platform. Ultimately, some of such findings are aimed to be used to assess the developed platform qualitatively by comparing the results with a manual search.

3.4 Methods

3.4.1 Network propagation and supervised classification

Gene-gene interaction networks were built from Bioplex (Huttlin et al., 2015) and STRING (Szklarczyk et al., 2017), while drug-gene interactions were collected from the STITCH (Kuhn, von Mering, Campillos, Jensen, & Bork, 2008) repository. Drugs were separated into 2 classes:

anti-cancer and non-anticancer drugs based on their classification in DrugCentral (Ursu et al., 2017), DrugBank (Wishart et al., 2008), and RepoDB (Brown & Patel, 2017), where FDA-approved drugs were considered anti-cancerous.

Drug-gene interactions were propagated through the gene-gene human interactome using a Google PageRank algorithm to obtain smoothed/diffused profiles (Figure 21). These profiles and the respective anti-cancer/non-anticancer labels were used to train a supervised classifier. Misclassified non-anticancer drug profiles were considered as drugs with a potential for repurposing. Sorting by probability, the top candidates were later used to assess the developed pipeline through a literature search.

3.4.2 Hierarchical classification

In (Galea et al., 2017), a hierarchical classification approach was developed and applied to biomedical data. This is based on performing dimensionality reduction at each node of a tree using 4 methods: (i) alternative partial least squares regression (SIMPLS) ("SIMPLS: An alternative approach to partial least squares regression - ScienceDirect," n.d.), (ii) recursive linear discriminant analysis using maximum margin criterion (MMC-LDA) (Haifeng Li, Jiang, & Zhang, 2006), (iii) support vector machine (SVM), and (iv) linear discriminant analysis using Fisherfaces approach (PCA-LDA) (Belhumeur, Hespanha, & Kriegman, 1997). This is followed by training a logistic regression classifier in a cross-validated one-vs-all fashion and choosing the best classifier by a voting system (Galea et al., 2017).

In a second approach, we developed a class prediction algorithm whereby a class from the bottom-most part of the hierarchical tree is left out of the cross-validation completely and upper

tree classes are predicted recursively starting from the root node at the upper-most hierarchical level (Galea et al., 2017).

These algorithms were applied to mass spectral data for 15 different bacterial species with their taxonomic tree as labels. To generalize this further, this was extended and applied to 9 different cancer types: breast adenocarcinoma, glioblastoma multiforme, kidney renal clear cell carcinoma, kidney renal papillary carcinoma, acute myeloid leukemia, lower grade glioma, lung adenocarcinoma, prostate adenocarcinoma, and bladder urothelial carcinoma. The genomic data for these cancer types was retrieved from The Cancer Genome Atlas (TCGA), processed, and their subtypes retrieved from literature (Galea et al., 2017) to devise the hierarchical tree.

Finally, a new visualization approach was devised and developed in Python to indicate misclassifications between classes in a hierarchical fashion, that is also appropriate for visualizing a large number of classes. Full methodological details are provided in the Supplementary Methods as derived from Galea et al. (2017).

To identify features which are potentially biomarkers to a bacterial species, genus or other taxonomic level, we have previously performed univariate Kruskal-Wallis tests at each node (Galea, 2015). Here this was extended to cancer types and sub-types by also performing pathway analysis to identify perturbed pathways.

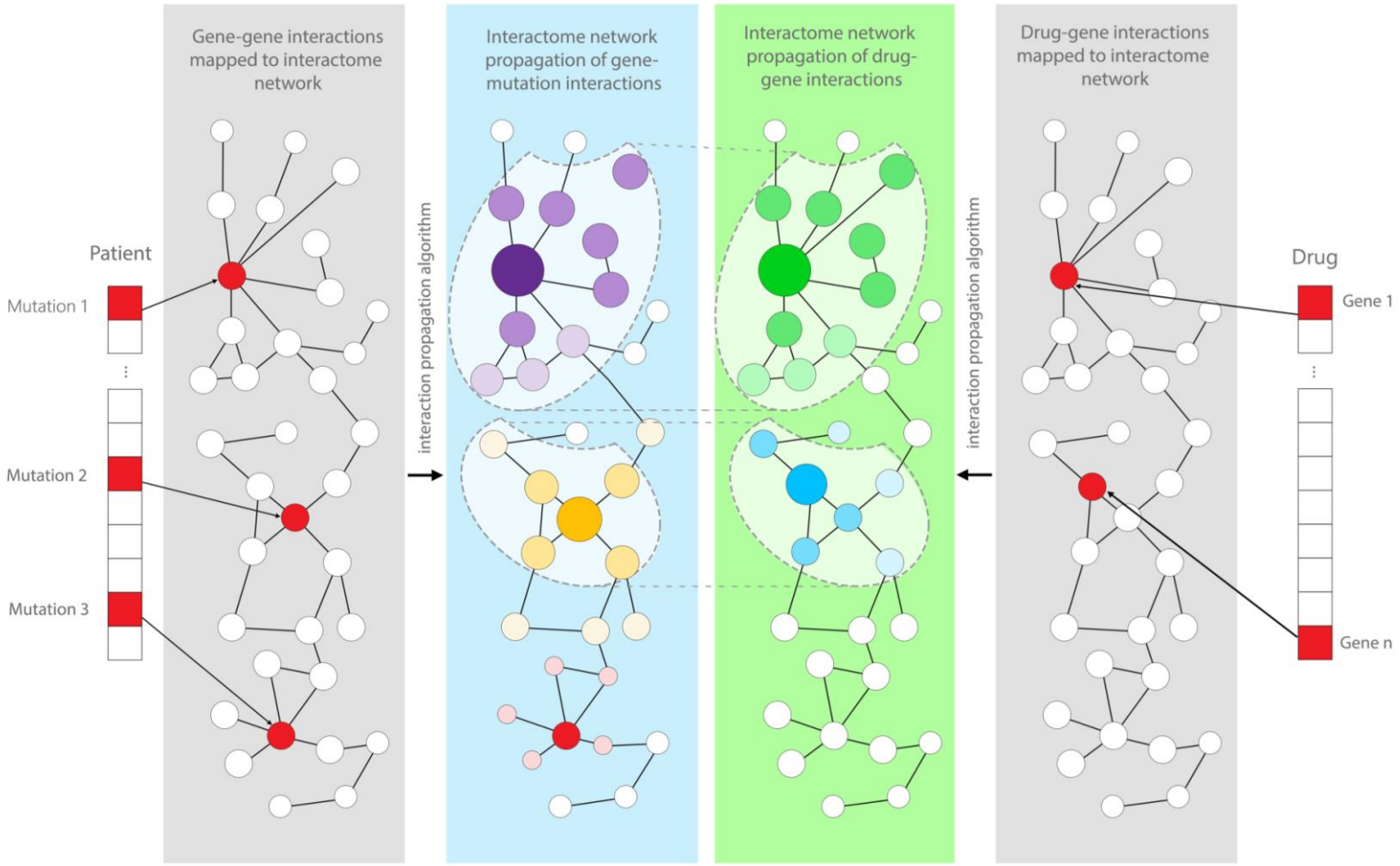


Figure 21. Graphic overview of the proposed network propagation approach developed and applied for the identification of drugs that may potentially be re-purposed for anti-cancer treatment. Gene-drug interactions are propagated to identify drug-influenced genes (RIGHT). Drugs with a similar mechanism are expected to overlap in the propagated gene profile. In a more patient-based genome approach, patient mutation data can also be propagated through the human interactome network to identify influencer genes (LEFT). The overlap between the propagated networks may indicate drugs that potentially can be more suitable based on a patient's genomic profile.

3.4.3 Systematic reviews

Literature on colorectal biomarkers for resistance to radiotherapy was manually reviewed. Briefly, MEDLINE articles were queried for the keywords: “rectal”, “cancer”, “neoplasm”, “response”, “pathological complete response”, “tumor response”, “radiotherapy”, “chemoradio*”, “neoadjuvant”, “predict*”, “biomarker”, “*omic”, “RNA”, and “DNA” (Poynter et al., 2019).

Candidate articles were manually identified, screened and included for information extraction according to the workflow in Supplementary Figure 1. Biomarkers studied, their statistical significance, and their respective gene ontologies were compiled. Subsequently, network-based meta-analysis was performed, where findings were grouped based on the gene ontology network, their statistical significance was aggregated based on p-values, q-values, odds ratios and sensitivity/specificity, and results represented as a node-based figure where the number of biomarkers in a given ontology is represented by the node size and statistical significance by the node color.

3.5 Results and Discussion

3.5.1 Network propagation

Classifying the genomic profiles for non-anti-cancer and anti-cancer drugs following network propagation achieved an accuracy of up to 91% (Gonzalez Pigorini, 2018; Veselkov et al., 2018). Influencer genes in non-anticancer drugs played a role in metabolism pathways, whereas the anti-cancer group of drugs were identified to play a role in cancer pathways such as: prostate cancer, endometrial cancer, melanoma, glioma, and general pathways in cancer (Gonzalez Pigorini, 2018).

Looking at the misclassifications, the top 10 non-anticancer drug profiles that were assigned to the anti-cancer class were: cetrorelix, celecoxib, indocyanine green, fluticasone furoate, clobetasol propionate, ethynodiol diacetate, gentian violet, cefradine, and phenmetrazine. The validity and mechanism of action of these drugs was discussed in detail by (Gonzalez Pigorini, 2018) and is summarized in Table 3.

Determining the robustness and independent reproducibility of these findings through comparison with previous literature would indicate validity of such findings, increasing confidence. Performing this manually would be very time consuming and therefore this highlights the need for an automated system. Some of these findings are used in this work for quantitative evaluation of the developed pipeline.

Table 3. Top 10 non-anti-cancer drugs identified with the potential of having anti-cancer properties, their respective target and a brief description of their mechanism. Adapted from (Gonzalez Pigorini, 2018).

Drug	Target	Description
Cetrorelix	Gonadotropin-releasing hormone receptor	Binds to gonadotropin-releasing hormone receptor to inhibit gonadotropin release
Celecoxib	Prostaglandin G/H synthase 2	Inhibits prostaglandin synthesis
Indocyanine green	*NA*	*NA*
Fluticasone furoate	Glucocorticoid receptor	Corticosteroid with anti-inflammatory activity
Flunisolide	Glucocorticoid receptor	Glucocorticoid receptor agonist
Clobetasol propionate	Glucocorticoid receptor	*uncertain*
Ethynodiol diacetate	Progesterone receptor	Binds to progesterone/estrogen receptors, slowing gonadotropin releasing hormone
Gentian violet	DNA	Bacterial cell wall stain. Mutagen and mitotic poison.
Cefradine	Penicillin-binding protein 1A	Binds to penicillin-binding proteins in bacteria, inhibiting bacterial cell wall synthesis
Phenmetrazine	Sodium-dependent noradrenaline/dopamine transporter	Blocks norepinephrine/dopamine reuptake

3.5.2 Systematic Reviews

From the compiled node graph (Figure 22), we observe that literature reports a large number of biomarkers in the cell communication and cell death ontologies that are linked with radiotherapy resistance. Specifically, apoptosis and cellular metabolism are the most studied ontologies, with findings being overall marginally significant. Of even higher significance are biomarkers belonging to the signal transduction ontology.

We also observe that ontologies with few biomarkers are either of low significance or high significance. This may indicate that not much research has investigated this aspect and therefore not many diverse biomarkers have been identified. Alternatively, this may also suggest that these findings may not be too robust and may require further validation to ensure these are not due to chance, given that cancer is known to affect a number of well-connected pathways. On the contrary, the large apoptosis node indicates the high diversity of biomarkers reported in literature to be related to radiotherapy response in cancer patients. Apoptosis is a well-studied mechanism that causes and controls cancer activity and therefore this justifies the observed compiled results.

While the aim of the current work is not to provide in-depth biomedical or biochemical insights, we hint on the importance of performing literature review for gaining insight into the progress and status of a field, and how compiling results in a compact way may provide quick and useful insights – all based on existing literature findings. Given this work was performed manually by searching through literature, we aim and hope to speed up this process by achieving similar insights when the information extraction process is automated.

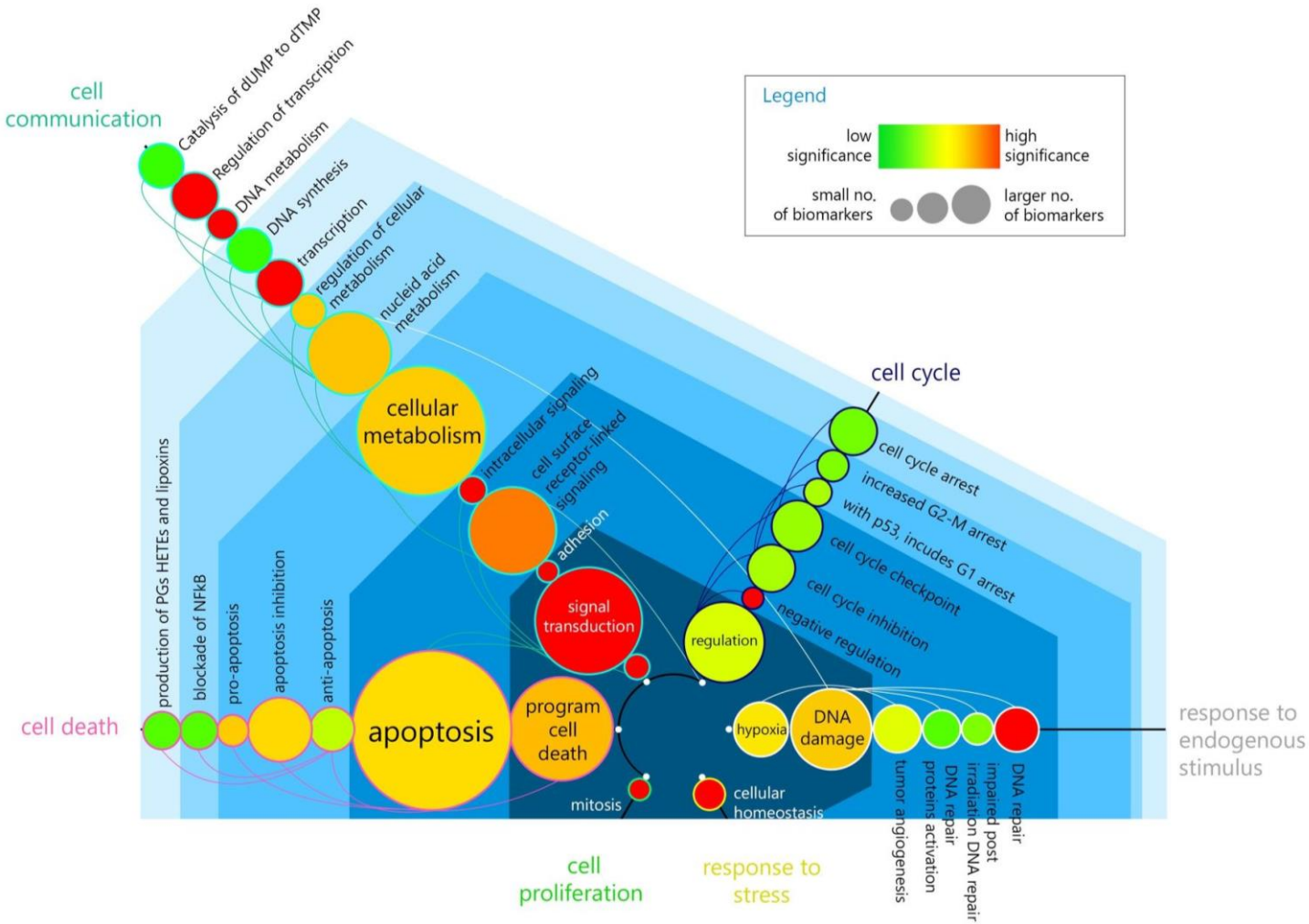


Figure 22. Network diagram summarizing published biomarkers studied in colorectal cancers. Size of nodes represents number of unique biomarkers for an ontology. Color of a node represents overall statistical significance reported in the original studies. Source: (Poynter et al., 2019)

3.5.3 Hierarchical Classification

Applying the hierarchical classification algorithm to the mass spectral profiles of 15 bacterial species achieved an average classification accuracy of 100% up to the family level and 91-94% at the genus and species levels (Galea et al., 2017) (Figure 23). For the cancer dataset, 32 cancer subtypes achieved an average of 76% classification accuracy. 19 out of 35 subtypes obtained over 80% accuracy, 10 achieved 60-80% accuracy while 6 subtypes were recorded with less than 60% accuracy (Figure 24) (Galea et al., 2017).

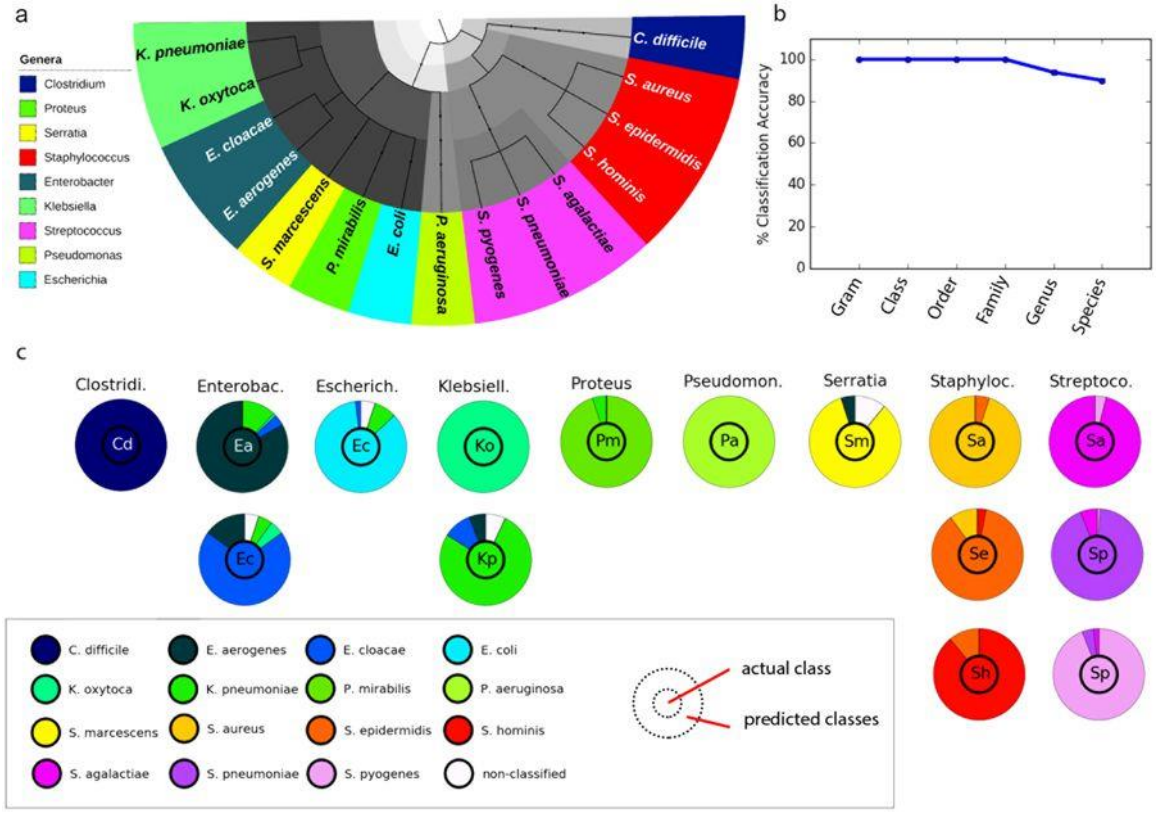


Figure 23. Classification of bacterial spectra into their respective taxonomic classes. a) taxonomic tree of the bacterial species considered, color-coded by their respective genera; b) classification performance for each level of the taxonomic tree (class, order, family, genus and species) and Gram properties; c) novel semi-quantitative visualization of the species classification performance indicating misclassification across the species and genera. Source: (Galea et al., 2017)

When performing leave-one-class-out validation, where a whole class from the bottom-most part of the hierarchical tree is omitted during training and predicted top-down, 13 out of 15 bacterial species achieved 100% accuracy (exemplary output for *Streptococcus agalactiae* is

shown in Supplementary Figure 2) and all cancer types except for one LGG and KIRP subtype were predicted with 95-100% accuracy (Figure 25). These results are discussed in detail in (Galea et al., 2017).

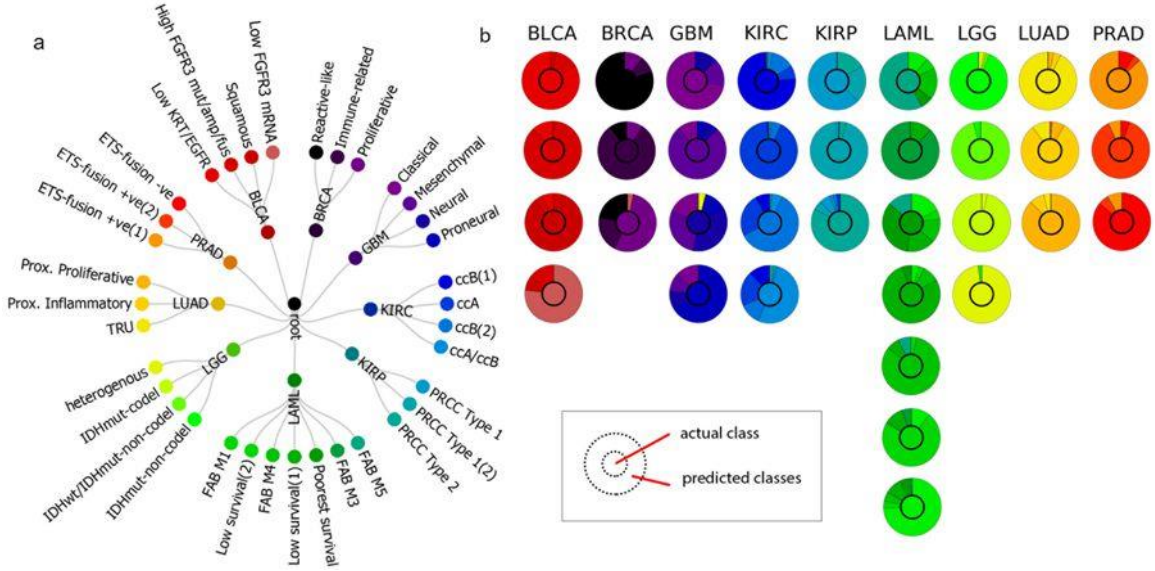


Figure 24. Classification of cancer genomic data into classes defined in literature. a) Hierarchical classification tree derived from literature and used for supervised training and prediction of different cancer types; b) semi-quantitative hierarchical classification performance of the different cancer subtypes and indication of where misclassification occurs across the same or different cancer types. Source: (Galea et al., 2017)

Taking this forward to identifying biomarkers by feature selection, in (Galea, 2015), a number of metabolic peaks were identified to be specific to bacterial classes (Table 4). Performing similar analyses to identify cancer-specific biomarkers or pathways effected, 78 genes were identified to be commonly differentially expressed across cancer types, including genes such as: NOP2, SASH1, and TGFBR3. Performing pathway analysis on this gene set, 6 pathways related to cell cycle, mitosis and DNA replication (Table 5) were identified. Validation of such findings based purely on experimental and quantitative data through an automated literature search would benefit greatly to determine the robustness and consistency with other independent research.

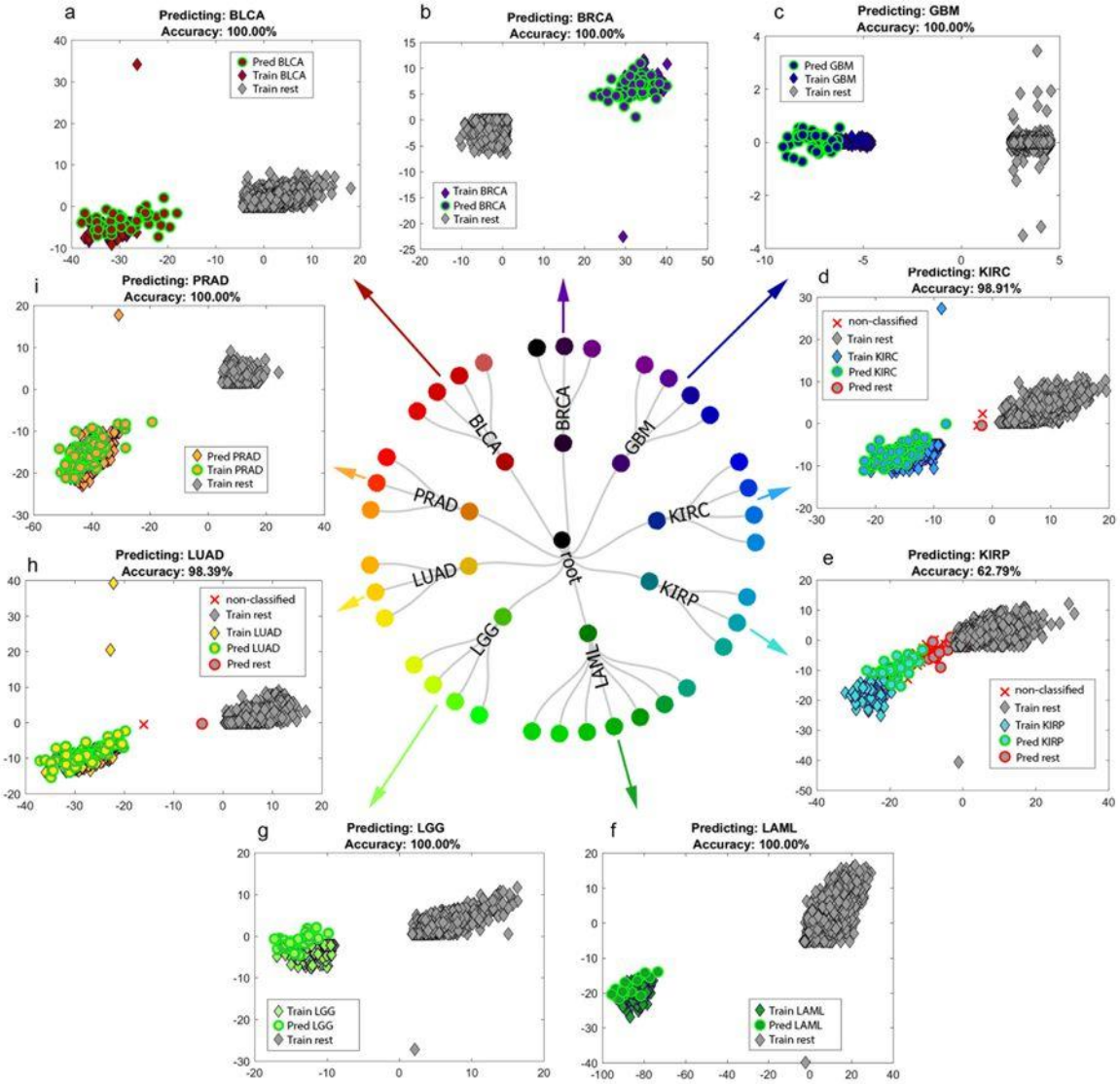


Figure 25. Prediction of the respective cancer type (BLCA, BRCA, GBM, KIRC, KIRP, LAML, LGG, LUAD and PRAD) for a selection of cancer subtypes. Source: (Galea et al., 2017)

Table 4. Number of metabolic features identified to be altered between bacterial classes by univariate feature selection. Source: (Galea, 2015)

Classes compared	Number of features extracted
Gram positive vs. Gram negative	1277
Bacilli vs. Clostridia	383
Bacillales vs. Lactobacillales	1141
Enterobacteriales vs. Pseudomonadales	589
Escherichia vs. other	80

Proteus vs. other	28
Serratia vs. other	41
Enterobacter aerogenes* vs. *Enterobacter cloacae	11
Klebsiella oxytoca* vs. *Klebsiella pneumoniae	16
Staphylococcus aureus* vs. *Staphylococcus epidermidis	21
***Streptococcus agalactiae* vs. other *Streptococcus* spp.**	28
***Streptococcus pneumonia* vs. other *Streptococcus* spp.**	29
***Streptococcus pyogenes* vs. other *Streptococcus* spp.**	19

Table 5. Pathways identified to be commonly altered between different cancer types.

Pathway	Overlapping genes	All genes	p-values	q-values
S Phase	6	71	5.48e-07	0.00147
Cell cycle, mitotic	11	400	7.53e-07	0.00147
Synthesis of DNA	5	47	1.64e-06	0.0021
Cell cycle	11	445	2.15e-06	0.0021
DNA replication	5	52	2.74e-06	0.00214
DNA strand elongation	4	32	1.02e-05	0.00664

3.6 Conclusion(s) and Future Direction(s)

Through developing new approaches and utilizing existing methods, we demonstrate how different approaches can lead to the identification of features associated with biomedical entities (drugs-mutations, bacteria-metabolites, cancer-genes, cancer-pathways) and therefore approaches that can make use of a literature search pipeline that can potentially validate or measure robustness of such findings.

Drugs identified as potentially anti-cancerous in this work are used in the final module to quantitatively assess the developed HASKEE pipeline. In future work, ‘validation’ or comparison of the manual systematic review findings and cancer-related genes/pathways can be executed.

Chapter 4 – Comparing and optimizing biomedical word representations

4.1 Abstract

Popular embeddings such as word2vec are word-level representations able to compute vectors only for words seen during training. Such models also do not capture sub-word information. These limitations are mitigated by character-based representations such as fastText, however such models have been minimally explored and/or used in the biomedical domain. In this work, fastText and word2vec models are optimized, evaluated and compared. In the named entity recognition task, fastText consistently outperformed word2vec. This is likely a result of capturing entity word compositionality information, as well as due to the representation of terms which are out-of-vocabulary. On the other hand, intrinsic evaluation performance varied, with optimal hyper-parameters identified to be dataset-dependent. This may be due to term type distribution differences between the different datasets. As different optimal hyper-parameters may be required for different tasks at hand, we provide pre-trained word2vec and fastText models with a range of different optimal hyper-parameters. The optimized models achieve state-of-the-art performance. Models and hyper-parameter sets are available on: https://github.com/dterg/bionlp-embed

4.2 Aims and Objectives

Aim(s):

- To compare and optimize word-based and character-based word representations for biomedical text;

Objective(s):

- To train and compare word2vec and fastText biomedical representations;
- To optimize model hyper-parameters based on intrinsic and extrinsic evaluation;

4.3 Introduction

Words have classically been represented by unique identifiers (as dictionary entries where labels correspond to the actual word) or by a set of extracted features. These approaches generate sparse vectors which do not capture word semantics. The development of distributional representations such as word2vec (Mikolov et al., 2013) and GloVe (Pennington, Socher, & Manning, 2014) has enabled the computation of dense word vectors based on word semantic similarity (see Section 2.4).

Word2vec embeddings are used extensively in the biomedical domain and have been optimized and investigated thoroughly in studies such as (Chiu, Crichton, Korhonen, & Pyysalo, 2016). However, word-level models such as word2vec suffer from 2 main limitations: (i) are unable to compute vectors for terms which are not seen during training (out-of-vocabulary; OOV); and (ii) the computed vectors do not capture sub-word information. In the biomedical domain, where new species are discovered, new genetic mutations reported, and new drugs/chemicals are synthesized, not being able to represent such entities may lose information. Additionally, throughout the years, improving orthographic and morphological word feature-engineering has been shown to generally improve NLP task performance. This also applies to the biomedical domain, where more recently, character-based neural networks have achieved state-of-the-art performance (Gridach, 2017). This therefore indicates the important role of capturing sub-word information.

Recently, character-based embeddings have been proposed, including: fastText (Bojanowski, Grave, Joulin, & Mikolov, 2017b) and MIMICK (Pinter, Guthrie, & Eisenstein, 2017). In such models, representations are computed based on word compositionality and co-occurrence with surrounding words (see Section 2.4.2). Such models therefore enable to not only capture word

semantics but also sub-word information. As vectors are computed based on the sequence of characters, these approaches also enable to compute vectors for OOV words.

Despite these advantages of character-based representations, they have been minimally explored and utilized in the biomedical domain. Therefore, in this work biomedical fastText representations are trained, optimized and evaluated on intrinsic datasets as well as extrinsic tasks, specifically biomedical named entity recognition. Performance is compared to word2vec embeddings by also training and optimizing word2vec models – an extension to the work by (Chiu, Crichton, et al., 2016). The optimal model is used for downstream tasks in the HASKEE pipeline.

4.4 Methods

4.4.1 Training data pre-processing

Word representations were trained based on PubMed abstracts and titles. PubMed parser (Achakulvisut et al., 2016) was used to parse the PubMed 2018 baseline corpus, representing articles by a one-line string. New line terminators were replaced by a whitespace and pre-processing was carried out by the NLPre package (He & Chen, 2018). This involved acronym resolution (where acronyms were replaced with full term phrases e.g. “Acute Chest Syndrome (ACS)” was changed to “Acute Chest Syndrome (Acute Chest Syndromee)”), and single character tokens and URLs removed. Tokenization was performed on whitespace and punctuation was retained (Galea, Laponogov, & Veselkov, 2018c).

4.4.2 Embeddings and hyper-parameters

The skip-gram architecture (see Section 2.4) was used to train word2vec and fastText representation models with the respective gensim implementations (Řehůřek & Sojka, 2010). The role of hyper-parameter selection on the quality of the embeddings and hence downstream

performance was tested as in (Chiu, Crichton, et al., 2016). The hyper-parameters: negative sample size, sub-sampling rate, learning rate (alpha), dimensionality, window size and minimum word count were tested, with larger value ranges for window size. As fastText models introduce the additional n-gram hyper-parameter, this was also tested for value ranges tested by the original authors (Bojanowski et al., 2017a). As fastText models may be up to 7.2x slower to train compared to word2vec (Supplementary Figure 3), only one hyper-parameter is modified at a time, while other hyper-parameters are kept constant. Performance was determined by intrinsic and extrinsic measures on a number of datasets (Galea et al., 2018c).

4.4.3 Intrinsic Evaluation

Intrinsic evaluation was performed on a number of datasets. The UMNSRS dataset consists of term pairs for disorders, symptoms, and drugs (Pakhomov, Finley, McEwan, Wang, & Melton, 2016). For each of the term pairs, the cosine similarity is retrieved from the train embedding and this is correlated to the manually assigned score in the UMNSRS dataset. 2 additional datasets were synthesized by computing graph-based similarity and relatedness scores: human disease ontology graph (HDO) (Schriml et al., 2012), and Xenopus anatomy and development ontology graph (XADO) (Segerdell, Bowes, Pollet, & Vize, 2008). Each graph was used to randomly generate 1 million pairs of entities. For each term pair, Wu and Palmer similarity metric (Wu & Palmer, 1994) was computed. A simplified Lesk algorithm (Lesk, 1986) was devised to establish a relatedness score by calculating the definition token overlap (after stopword removal) and normalizing by the maximum definition length. If a definition was not available for at least one of the terms in a given pair, the pair was excluded (Galea et al., 2018c).

Word2vec models are unable to compute vectors for out-of-vocabulary words; that is, terms which are not available in the training corpus. Therefore, pairs of terms which are not in-

vocabulary are often excluded from evaluation in literature. However, as fastText is able to compute vectors for OOV terms, to enable a fair comparison between word2vec and fastText, OOV terms were represented with null vectors in word2vec models - as performed by the original authors (Bojanowski et al., 2017a). To establish the gain (or loss) in performance resultant of computing vectors by fastText, the correlation was determined in 2 ways: by considering only in-vocabulary terms, and by null-imputing OOV term pairs for word2vec (Galea et al., 2018c).

4.4.4 Extrinsic Evaluation

Extrinsic evaluation involves determining the impact on performance on real-word natural language processing tasks. In this case, performance differences was quantified mainly in the named entity recognition task by evaluating on 3 different corpora: (i) JNLPBA (Jin-Dong Kim, Ohta, Tsuruoka, Tateisi, & Collier, 2004) which annotates cell lines, cell types, proteins, DNA, and RNA; (ii) BioCreative II Gene Mention task corpus (BC2GM) (Smith et al., 2008) annotating genes; and (iii) CHEMDNER (Krallinger, Leitner, et al., 2015) as processed by (Luo et al., 2018), which annotates drugs and chemicals. These corpora were split into training, development and test sets (Galea et al., 2018c). The BiLSTM-CRF neural network architecture (Lample et al., 2016) (Section 2.3.2) was used. Corpus-specific word2vec and fastText were trained with hyper-parameters recorded to achieve the best performance. To obtain a generalizable model, separate embeddings were also trained on the average optimal hyper-parameters across all datasets (intrinsic and extrinsic). Optimization was carried out on the development set while the final evaluation was performed on the test subset.

4.4.5 Performance Generalizability

To determine the generalizability of the optimized models, the optimized word2vec and fastText models were applied to additional datasets and NLP tasks. The NER performance for diseases and chemicals were tested on the CDR corpus (J. Li et al., 2016), the sequential sentence classification performance was compared for both models (as described in detail in Section 5.4.5), as well as analogy completion performance. The analogy completion task was tested on the BMASS corpus (Newman-Griffis, Lai, & Fosler-Lussier, 2017), where evaluation was carried out using the code by (Newman-Griffis et al., 2017).

4.5 Results and Discussion

4.5.1 Corpus

Pre-processing the PubMed 2018 baseline corpus produced a 3.4 billion token training dataset with a vocabulary of 19 million terms (Table 6).

Table 6. Tokens and unique tokens in the processed training data derived from PubMed articles at different word frequency thresholds. Source: (Galea et al., 2018c).

Minimum word frequency threshold	0	5	10
Tokens	3,435,773,079	3,412,644,449	3,402,300,795
Vocabulary	19,099,369	3,410,473	1,806,181

4.5.2 General trends: word2vec hyper-parameter selection

Generally, word2vec models (as visualized in the exemplary Figure 26) achieved intrinsic and extrinsic performance trends similar to those reported and discussed in detail by (Chiu, Crichton, et al., 2016) for the corpora: UMNSRS, BC2GM and JNLPBA. Here, as in (Galea et al., 2018c), we record minor trend differences for the hyper-parameters: minimum word count (Figure 27), and window size (Figure 28). (Chiu, Crichton, et al., 2016) report decreasing

relatedness when increasing minimum word count for the UMNSRS standards, whereas here we find that this trend occurs for both intrinsic metrics (Galea et al., 2018c).

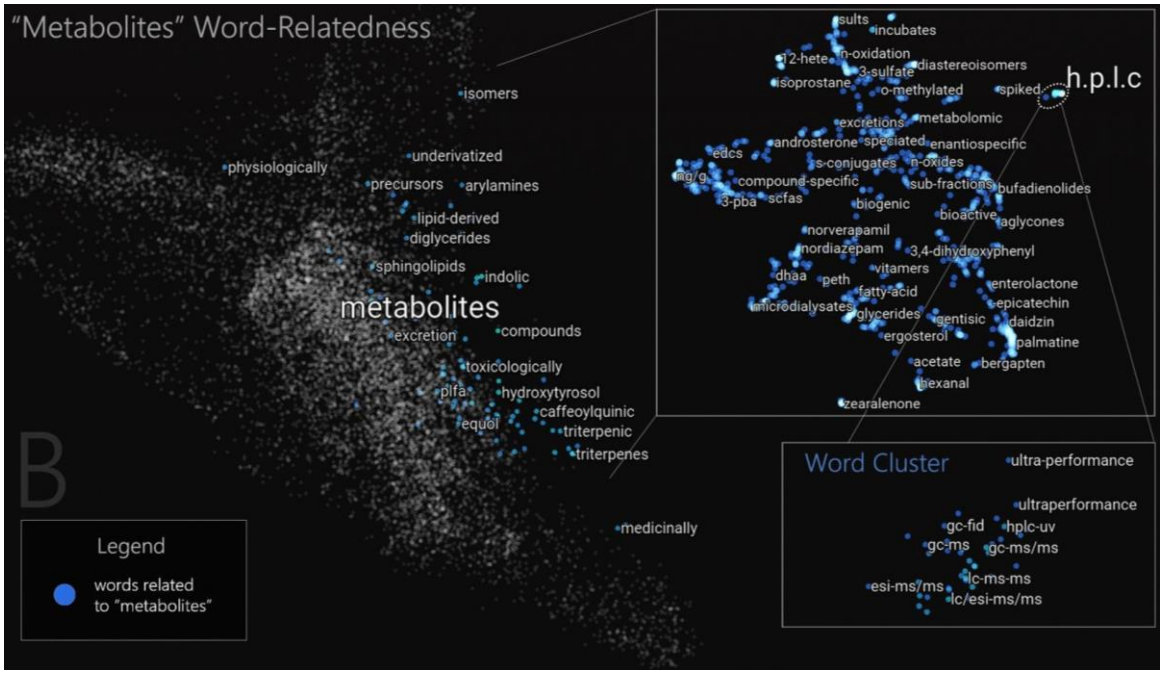


Figure 26. Word2vec word representations reduced to a three-dimensional space with t-SNE. A selection of words inferred to be related to the word "metabolites" by cosine similarity are highlighted and isolated. In turn, "h.p.l.c" appears to cluster with words relating to the technologies used in metabolic profiling. Figure published in (Galea et al., 2018c).

Contrastingly, increasing window size also increased performance, particularly in the UMNSRS standards (Figure 28). An identical trend was reported by (Chiu, Crichton, et al., 2016) up to a maximum window size of 30. By reasoning that abstracts likely refer to a single topic, we predicted that by extending the window size to the average abstract length, this might increase the performance by capturing more topical information. This was confirmed by an increase in similarity and relatedness for UMNSRS (0.675/0.639 obtained here compared to 0.627/0.584 by (Chiu, Crichton, et al., 2016). It is worth noting that as the absolute performance obtained here for window size values shared with (Chiu, Crichton, et al., 2016), there are other factors contributing to such performance improvements, including: different pre-processing approaches, and increase in PubMed training data (Galea et al., 2018c).

When considering extrinsic evaluation, similar to (Chiu, Crichton, et al., 2016), a lower window size obtained better performance.

4.5.3 General trends: fastText hyper-parameter selection

As fastText shares the same overall model architecture with word2vec, the hyper-parameter space is similar. This gave similar performance trends for fastText to those obtained by word2vec (Figure 27; Figure 28). Nonetheless, optimal hyper-parameter values were not always identical; as discussed in detail in the following sections (Galea et al., 2018c).

4.5.4 Model performance comparison – Intrinsic evaluation

Comparing word2vec and fastText performance based on intrinsic datasets achieved variable results. Whereas word2vec consistently achieved higher UMNSRS similarity and relatedness across all values for negative sampling size, dimensions and window size, performance on HDO and XADO varied. For hyper-parameters such as window size, dimensionality, and negative sampling, fastText tended to compare with or outperform word2vec (Galea et al., 2018c).

Dataset-dependent performance differences may be due to several factors, including: (i) differences in the number of OOV terms; (ii) differences in the frequency of terms in the training data (hence rarity of the terms); and (iii) differences in the term types (Galea et al., 2018c). The UMNSRS dataset was constructed on several corpora, including PubMed Central, therefore only up to 1.3% of the tokens (9 unique tokens) were OOV. Contrastingly, HDO contained up to 5% OOV terms. As null vectors do not capture any information and are used to represent OOV terms in word2vec models, these expect a decrease in performance as the number of OOV terms increases in a dataset (Galea et al., 2018c).

However, when evaluation was carried out by excluding OOV terms, and therefore these were omitted completely rather than imputed, similar performance trends were obtained. This suggests that differences in the rates of OOV between the different datasets is not the major factor contributing to intrinsic performance differences (Galea et al., 2018c). As reported by the original authors for the English WS353 dataset (Bojanowski et al., 2017a), one potential reason could be that fastText degrades performance for UMNSRS in-vocabulary terms (Galea et al., 2018c).

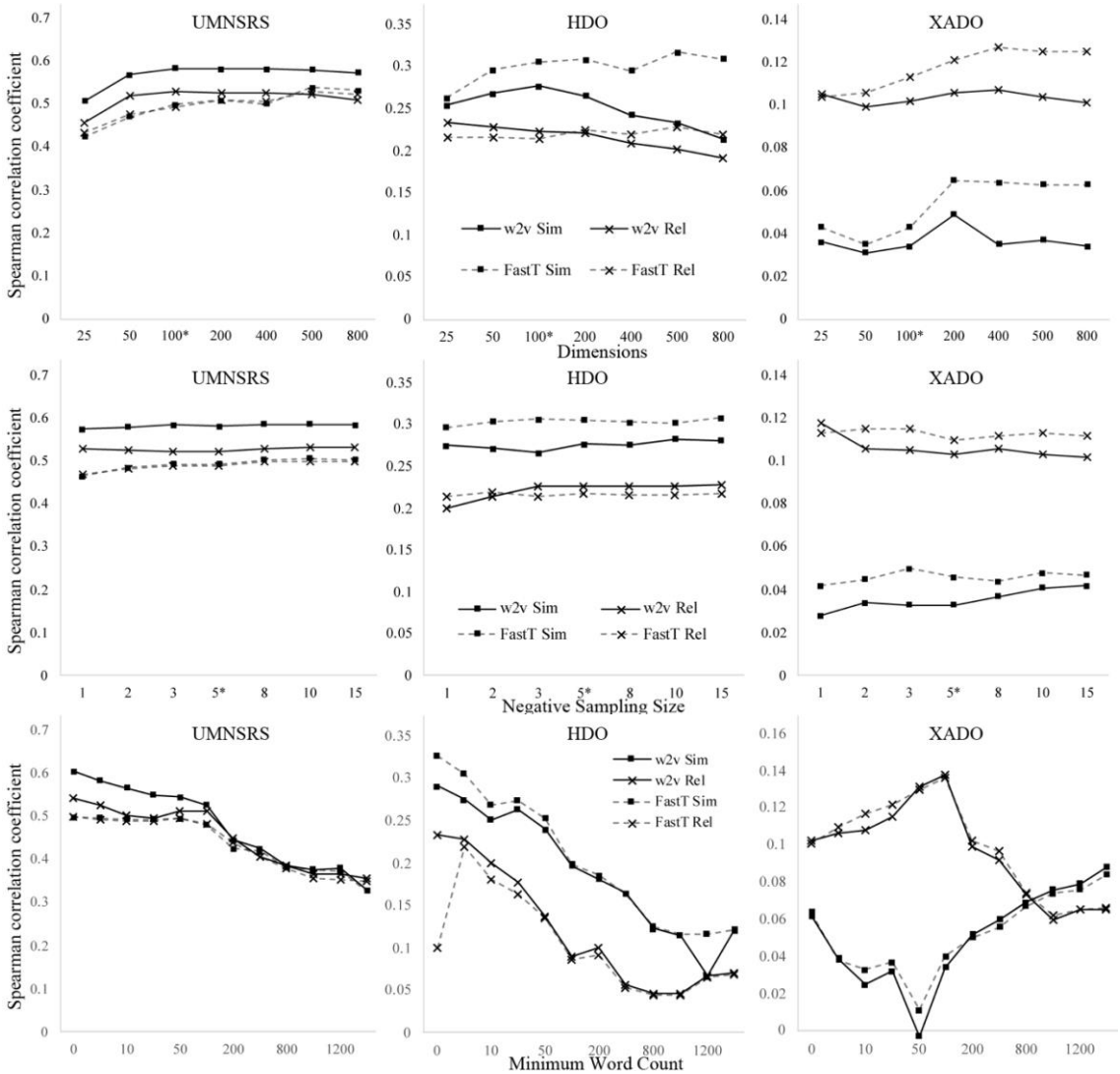


Figure 27. Intrinsic performance of word2vec (w2v) and fastText (FastT) word representations measured by similarity and relatedness for UMNSRS, HDO and XADO datasets, when varying the hyper-parameters: dimensions, negative sampling size and minimum word count.

When considering the frequency by which in-vocabulary terms for each dataset occurs in the training data, UMNSRS was measured to have a 4-fold median rank frequency higher than HDO. As fastText was identified to perform better on HDO, this may suggest that fastText representations are better for rarer terms. However, XADO is similar to UMNSRS in terms of in-vocabulary rate, yet overall fastText outperforms word2vec. This indicates performance differences are likely contributed to by a number of factors, including potentially the quality of the ontology graph (Galea et al., 2018c).

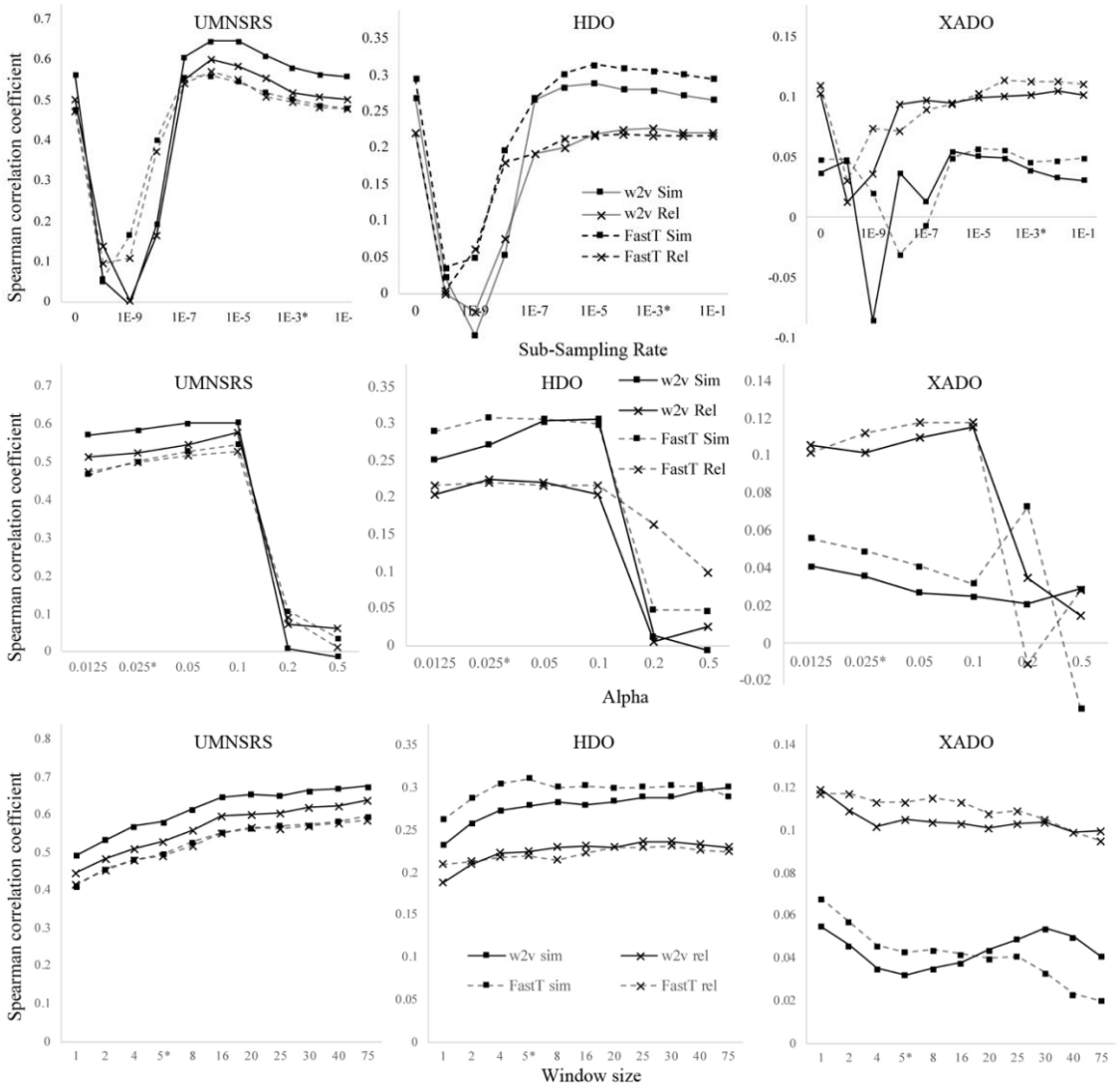


Figure 28. Intrinsic performance of word2vec (w2v) and fastText (FastT) word representations measured by similarity and relatedness for UMNSRS, HDO and XADO datasets, when varying the hyper-parameters: sub-sampling rate, learning rate (alpha) and window size.

fastText was reported to outperform word2vec in languages such as German, Russian, Arabic and English words (Bojanowski et al., 2017a). This was backed up by the word compositionality and character feature differences between different languages. As biomedical text contains various types of entities such as chemicals, proteins, and genes which are rich in orthographic features like punctuation, special characters, and digits, and nomenclature standards which may introduce compositionality, we expect different representation quality between the entity classes. Differences in the distribution of entity types in the intrinsic datasets may be an additional factor contributing to such performance differences.

4.5.5 Model performance comparison – Extrinsic evaluation

Unlike intrinsic performance, fastText consistently outperformed word2vec across all 3 corpora at any hyper-parameter value when evaluated on the named entity recognition task (Figure 29), with BC2GM and CHEMDNER recording the highest performance differences between the different representation models. As these corpora annotate one entity type (chemicals in CHEMDNER; genes in BC2GM), whereas JLNPBA annotates 5 entity classes, performance differences are more consistent and less variable in the former corpora (Galea et al., 2018c).

As introduced, fastText captures word compositionality and has been reported to outperform word2vec for more structured languages (Bojanowski et al., 2017a). In the biomedical domain, entities such as chemicals follow standardized nomenclature, introducing internal word structure. In the IUPAC nomenclature standards, for example, prefixes denote properties such as quantity of residues and substituents, whereas suffixes denote structure of the molecule. This structure enables fastText to train a model that captures chemical structure similarity. For example, when querying the OOV terms: 1,2-dichloromethane and 1-(dimethylamino)-2-

methyl-3,4-diphenylbutane-1,3-diol, fastText returns structurally-related chemicals (Table 7). word2vec is unable to perform this as it treats tokens which share prefixes/suffixes as completely different tokens. This accounts for performance differences between the two representations. A more detailed discussion is provided in (Galea et al., 2018c). Furthermore, with complex chemical names such as "1-(dimethylamino)-2-methyl-3,4-diphenylbutane-1,3-diol", it is highly likely for such names to be OOV; an advantage of sub-word-based representations.

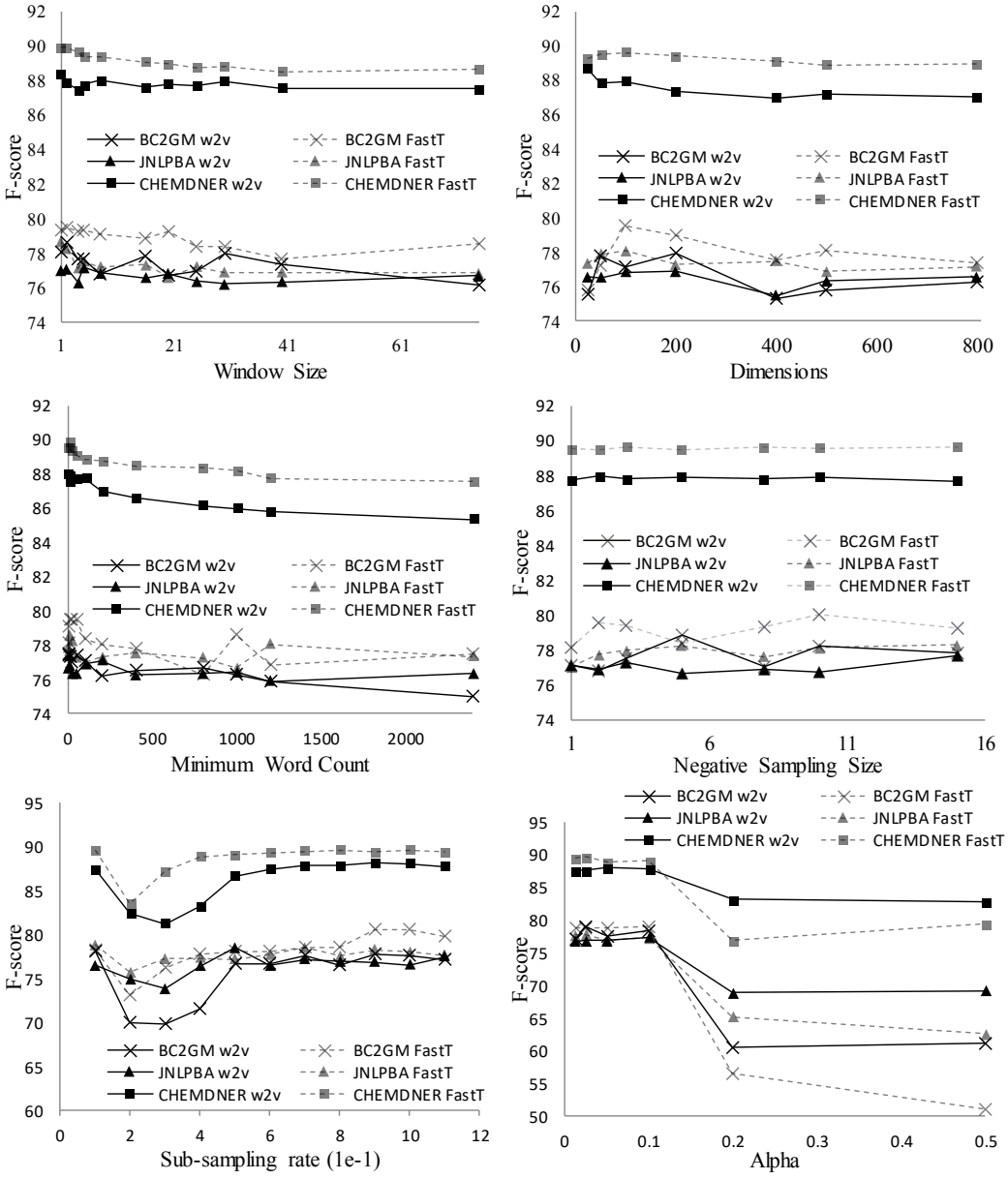


Figure 29. Extrinsic performance of word2vec (w2v) and fastText (FastT) word representations measured by named entity recognition accuracy (F-score) on the corpora BC2GM, JNLPBA and

CHEMDNER, when varying the hyper-parameters: window size, dimensions, minimum word count, negative sampling size, sub-sampling rate and learning rate (alpha).

Table 7. Top 5 most similar words to the out-of-vocabulary chemicals: 1,2-dichloromethane and 1-(dimethylamino)-2-mthyl-3,4-diphenylbutane-1,3-diol, and gene ZNF560. Source: (Galea et al., 2018c).

1,2-dichloromethane	1-(dimethylamino)-2-methyl-3,4-diphenylbutane-1,3-diol	ZNF560
1,2-dichloroethane	8-(N,N)-diethylamino)octyl-3,4,5-trimethoxybenzoate	ZNF580
1,2-dichlorobenzene	1,3-dimethylamylamine	ZNF545
Dibromochloromethane	8-(diethylamino)octyl	ZNF582
1,2-dichloropropane	2-chyclohexyl-2-hydroxy-2-phenylacetate	ZNF521
Water/1,2-dichloroethane	diethylamine	SOX1

While to a lesser extent than chemicals, gene and protein names also contain some internal structure in the full names and the symbolic acronymic abbreviations, where the root symbols represent the gene family (Galea et al., 2018c). Given the gene ZNF560 as an example, the prefix ZNF* stands for “Zinc Finger (proteins)” and fastText recalls names which share this prefix and therefore functionality (Table 7).

From the above examples, the advantages of character-based models over token-based models have been highlighted, particularly for OOV tokens. However, based on intrinsic evaluation, it is suggested that in some cases word2vec outperforms fastText. An example of this is when recalling similar vectors to the term “phosphatidylinositol-4,5-bisphosphate”. Whereas fastText is able to recall minor chemical variants (Table 8), word2vec is able to capture synonymous terms and abbreviations such as PIP2 and PtdIns(4,5)P2 (Galea et al., 2018c). This is a similar trade-off for genetic variants, where fastText recalled “rs-” prefixed terms while word2vec recalled the actual genetic variation term specified by the RS identifier, that

is: 590C>T and 590C/T (Table 9) (Galea et al., 2018c). Additional examples demonstrating this trade-off are given in Table 10 - Table 14.

Table 8. Top 5 most similar words to phosphatidylinositol-4,5-bisphosphate. Syntactically similar terms are recalled by fastText whereas word2vec recalls less syntactically similar terms but relevant entities such as abbreviated forms and synonyms, where PtdIns(4,5)P2 and PIP2 are synonymous to the query term. Source: (Galea et al., 2018c).

word2vec	fastText
4,5-bisphosphate	phosphatidylinositol-4,5-biphosphate
phosphatidylinositol	phosphatidylinositol-(4,5)-bisphosphate
4,5)-bisphosphate	phosphatidylinositol-4-phosphate
PIP2	4,5-bisphosphate
PtdIns(4,5)P2	phosphatidylinositol-4

Table 9. Top 10 most similar words to rs2243250; 590C/T polymorphism of Interleukin 4. "RS"- prefixed terms refer to Reference SNP identifiers. Source: (Galea et al., 2018c).

word2vec	fastText
Rs2070874	Rs1800896
Rs1800871	Rs2275913
Rs8193036	Rs2430561
Rs1800872	Rs2241880
Rs20541	Rs2070874
Rs2243248	Rs1800925
Rs2227284	Rs2228145
Rs4711998	Rs1143634
590C>T	Rs1800872
590C/T	Rs1143627

Table 10. Top 10 most similar words to acrodysostosis - a skeletal malformations disorder. Most of the recalled terms refer to genetic disorders of the bone, skin or endocrine system. Source: (Galea et al., 2018c).

word2vec	fastText
Albright_hereditary_osteodystrophy	dysostosis
Familial_glucocorticoid_deficiency	pycnodysostosis
McCune_Albright_syndrome	pyknodysostosis
McCune_-Albright	dysostoses
Melnick_Needles_syndrome	spondyloenchondrodysplasia
Hypochondroplasia	hereditary_multiple_exostosis
Hajdu-Cheney	alright_hereditary_osteodystrophy
PHP-la	pseudoachondroplasia
NFNS	chondrodysplasia
PHP1A	macrodystrophia

Table 11. Top 5 most similar terms to the out-of-vocabulary genetic variant LRG_1:g.8463G>C. "RS-" prefixed terms represent database accession identifiers for reference variants. Source: (Galea et al., 2018c).

rs2243250
rs2241880
rs3212227
rs3212986
rs3748067

Table 12. Top 10 most similar words to the term: "ZNF580 – Zinc Finger Protein 580" in the word2vec and fastText embeddings. Source: (Galea et al., 2018c).

word2vec	**fastText**
hCTGF	ZNF545
Focal_adhesional_kinase	ZNF582
Tmfn2	ZNF521
Deltanp63a	ZNF24
RTEF-1	ZNF202
p-CREB-1	ZNF32
ITGa5	ZNF217
BMP9-dependent	BTG1
Ox-LDL-injured	ZNF281
IGFBP-3-mediated	ZNF703

Table 13. Top 10 most similar words to the term: "1,2-dichloroethane" in the word2vec and fastText embeddings. Source: (Galea et al., 2018c).

word2vec	**fastText**
1,1,1,2,2-tetrachloroethane	dichloroethane
1,1,2-trichloroethane	Water/1,2-dichloroethane
chlorobenzene	1,1-dichloroethane
1,2-dichlorobenzene	1,1,2-trichloroethane
1,4-dioxane	1,2-dichlorobenzene
nitrobezene	chloroethane
dichloroethane	cis-1,2-dichloroethene
Toluene	1,1,2,2-tetrachloroethane
1,1-dichloroethene	trichloroethane
tetrachloroethane	Trans-1,2-dichloroethylene

Table 14. Top 10 most similar words to the term: "zinc_finger_protein" in the word2vec and fastText embeddings. Source: (Galea et al., 2018c).

word2vec	fastText
Zinc-finger	Zinc_finger_proteins
ZFP	RET_finger_protein
cGATA-1	Zinc_finger
Neural-restrictive	Ret_finger_protein
Six-zinc	Zinc_fingers
SZF1	Zinc-finger
SP/KLF	ZFP
Six-finger	KRAB
GAGA-like	KRAB-ZFPs
UtroUp	KRAB-ZFP

4.5.6 Effect of n-grams size

Similar to other hyper-parameters, the range of n-grams showed high variability between the different intrinsic datasets, with optimal values recorded at 6-7 and 3-4 for UMNSRS and XADO, respectively and 5-{6,7,8}, 4-6 and 6-8 for HDO (Table 15). As previously discussed, this heterogeneity is potentially due to entity type differences in the different datasets (Galea et al., 2018c).

High consistency was achieved for n-gram ranges based on extrinsic datasets, with best performance recorded at the ranges 3-7 and 3-8 (Table 15).This suggests short and long n-grams capture information, particularly for genes and proteins, complying with previous discussion on nomenclature standards and conventions (Galea et al., 2018c).

Table 15. The role of character n-gram ranges on intrinsic (UMNSRS, HDO and XADO; upper row = similarity, lower row = relatedness) and extrinsic performance (JNLPBA, CHEMDNER and BC2GM). Highest performance is highlighted in bold and accuracies within standard error of highest performance is indicated in italics. Source: (Galea et al., 2018c).

	UMNSRS						HDO						XADO					
	3	4	5	6	7	8	3	4	5	6	7	8	3	4	5	6	7	8
2	0.443	0.463	0.503	0.532	0.544	0.554	0.219	0.282	0.295	0.296	0.302	0.302	0.031	0.047	0.039	0.032	0.030	0.024
	0.427	0.458	0.497	0.507	0.512	0.516	0.193	0.218	0.219	0.222	0.223	0.224	0.105	0.106	0.106	0.113	0.114	0.114
3		0.487	0.517	0.548	0.560	0.561		0.298	0.307	0.307	0.312	0.312		**0.054**	0.048	0.038	0.032	0.030
		0.478	0.506	0.524	0.530	0.522		0.213	0.217	0.215	0.226	0.225		0.111	0.112	**0.117**	**0.117**	**0.117**
4			0.534	0.562	0.570	0.582			0.313	*0.318*	0.316	0.315			0.040	0.036	0.035	0.030
			0.523	0.539	0.533	0.540			0.218	0.227	0.228	0.227			0.110	0.111	0.113	0.109
5				0.584	0.603	0.596				**0.320**	*0.319*	**0.320**				0.034	0.031	0.029
				0.554	**0.565**	0.552				0.230	0.226	0.228				0.112	0.108	0.109
6					**0.612**	0.607					0.317	*0.319*					0.037	0.035
					0.556	0.549					0.228	**0.234**					0.110	0.108
7						0.601						0.314						0.033
						0.542						0.231						0.102

	BC2GM						JNLPBA						CHEMDNER					
	3	4	5	6	7	8	3	4	5	6	7	8	3	4	5	6	7	8
2	78.96	*79.72*	*79.78*	*79.91*	*79.71*	**80.26**	*78.20*	77.76	77.99	77.89	77.96	77.96	89.14	89.48	*89.66*	*89.72*	89.58	89.46
3		*79.69*	78.88	78.77	*79.90*	*79.91*		77.83	77.86	77.67	**78.46**	*78.30*		89.48	*89.67*	*89.67*	**89.75**	*89.62*
4			78.94	78.42	78.91	79.05			77.58	76.91	78.00	77.58			89.28	89.37	89.22	89.32
5				77.12	78.67	77.45				76.72	78.04	76.92				89.22	89.13	89.10
6					77.77	77.97					77.82	78.06					89.03	88.81
7						77.73						77.83						88.60

4.5.7 Optimized models' performance

As observed in the overall trends, different datasets provide different optimal hyper-parameter values. Training embeddings on UMNSRS' optimal hyper-parameters achieved 0.733 similarity and 0.686 relatedness with the word2vec model, exceeding 0.652/0.601 similarity/relatedness by (Chiu, Crichton, et al., 2016) and 0.681/0.635 by (Yu, Wallace, Johnson, & Cohen, 2017) (Galea et al., 2018c) (Table 16).

For named entity recognition, fastText optimized individually on extrinsic corpora outperformed word2vec consistently, achieving: 79.33%, 73.30% and 90.54% for BC2GM, JNLPBA and CHEMDNER, respectively (Galea et al., 2018c) (Table 16). These results outperform (Chiu, Crichton, et al., 2016), (Luo et al., 2018), (S. Pyysalo, Ginter, Moen, Salakoski, & Ananiadou, 2013) and (Kosmopoulos, Androutsopoulos, & Paliouras, n.d.). It is worth noting, however, that except for (Luo et al., 2018), such differences are also contributed to by advancements in architectures.

Table 16. Compilation of intrinsic and extrinsic performance for our trained embeddings and others reported in literature.

Author(s)	Description	UMNSRS	BC2GM	CHEMDNER	JNLPBA
Our embeds	Optimized hyper-params; BiLSTM-CRF	**0.733 / 0.686**	**79.33%**	**90.54%**	**73.30%**
(Chiu, Crichton, et al., 2016)	Optimized hyper-params	0.652 / 0.601	76.89%	64.13%	-
(Luo et al., 2018)	BiLSTM-CRF	-	-	89.28%	-
(S. Pyysalo et al., 2013)	Default params	0.549 / 0.506	77.01%	-	63.66%
(Kosmopoulos et al., n.d.) [BioASQ]	-	0.589 / 0.509	75.51%	-	62.85%

When optimizing embeddings across both intrinsic and extrinsic datasets (Table 17), performance difference decreased. This is accounted for by the different optimal hyper-parameters and overall trends between the datasets. As intrinsic performance has been reported to not be reflective of extrinsic performance (Chiu, Korhonen, & Pyysalo, 2016), we trained separate models based on optimal hyper-parameters from intrinsic and extrinsic datasets (Table 18).

Table 17. Optimized hyper-parameters for word2vec (w2v) and fastText (FastT) across intrinsic and extrinsic datasets. Source: (Galea et al., 2018c).

	w2v	FastT
Window	30	25
Negative	15	10
Sampling	1e-5	1e-5
Min-count	0	0
Alpha	0.025	0.025
Dim	200	200
N_grams	-	6-7

Table 18. word2vec (w2v) and fastText (FastT) hyper-parameters optimized across intrinsic standards and extrinsic corpora. Source: (Galea et al., 2018c).

	Optimized on intrinsic standards		Optimized on extrinsic corpora	
	w2v	FastT	w2v	FastT
Window	30	25	2	1
Negative	15	3	5	10
Sampling	1e-5	1e-5	1e-3	1e-3
Min-count	0	0	10	5
Alpha	0.05	0.05	0.1	0.025
Dim	200	200	50	100
N_grams	-	6-7	-	3-7

4.5.8 Generalizability of optimal model performance

Qualitatively, fastText models were observed to recall chemical structural analogs when performing an analogy task. For example, subtracting the vector for the term “sulfur” from the term “sulfuric_acid” and adding “phosphorus” returns “phosphoric acid” (Galea et al., 2018c). Given the discussed internal structure of chemicals, this suggests fastText models may provide good performance in analogy resolution. However, given the observed tradeoff between different entity types, results may vary. When determining the generalizability of the extrinsically optimized models, this analogy resolution capability of both word2vec and fastText was evaluated.

fastText outperformed word2vec in 18/25 of the analogy relationship types (Table 19), with the highest performance gain being for the relationship types: M2-noun-form-of and M1-adjectival-form-of, whereas performance loss is for the relationship types: L3-has-tradename and L2-has-lab number. Looking at examples from these relationship types (Table 20), it can be observed that these trends agree with the qualitative observations.

Table 19. Analogy resolution performance for the optimal word2vec and fastText models on the BMASS dataset.

Analogy relationship	word2vec	fastText	fastText % gain
A1-anatomic-structure-is-part-of	0.016	0.018	**10**
A2-anatomic-structure-has-part	0.011	0.009	-18
A3-is-located-in	0.025	0.027	**8**
B1-regulated-by	0.292	0.436	**49**
B2-regulates	0.027	0.046	**67**
B3-gene-encodes-product	0.010	0.009	-8
B4-gene-product-encoded-by	0.017	0.033	**95**
C1-associated-with-malfunction-of-gene-product	0.001	0.002	**67**
C2-gene-product-malfunction-associated-with-disease	0.004	0.009	**109**
C3-causative-agent-of	0.064	0.091	**42**
C4-has-causative-agent	0.013	0.018	**30**
C5-has-finding-site	0.007	0.016	**129**
C6-associated-with	0.029	0.049	**70**
H1-refers-to	0.043	0.090	**110**
H2-same-type	0.078	0.132	**71**
L1-form-of	0.116	0.113	-3
L2-has-lab-number	0.019	0.003	-85
L3-has-tradename	0.017	0.001	-95
L4-tradename-of	0.013	0.005	-58
L5-associated-substance	0.024	0.037	**54**
L6-has-free-acid-or-base-form	0.149	0.174	**17**
L7-has-salt-form	0.472	0.436	-8
L8-measured-component-of	0.016	0.020	**23**
M1-adjectival-form-of	0.036	0.107	**197**
M2-noun-form-of	0.019	0.088	**360**

Generalizing the NER performance further, using the optimized embeddings for training and prediction of CDR, an unseen corpus, fastText outperformed word2vec (82.04% vs. 78.56% for diseases and 90.55% vs. 88.34% for chemicals) (Ayoub, 2018).

Table 20. Examples of analogies from the 2 fastText's best performing relationship types (M2-noun-form-of and M1-adjectival-form-of) and 2 fastText's worst performing relationship types (L3-has-tradename and L2-has-lab-number).

Best-performing	
M2-noun-form-of	muscular : muscle rotated : rotation macroglial : macroglia
M1-adjectival-form-of	ataxia : ataxic radius : radial sweating : sweaty
Worst-performing	
L3-has-tradename	aluminium hydroxide : gaviscon basiliximab : simulect vorinostat : zolinza
L2-has-lab-number	cefixime : fk027 dimesna : bnp7787 bexarotene : lgd1069

4.6 Conclusion(s) and Future Direction(s)

In this work, pre-trained biomedical word representations were optimized, and the performance of word2vec and fastText models was compared. fastText was shown to consistently outperform word2vec in named entity recognition, but intrinsic evaluation varied. Quantitatively, a trade-off between sub-word information and semantics from context was observed between word2vec and fastText embeddings. This was observed for a selection of biomedical entities as well as in analogy resolution benchmark datasets. The models trained achieve state-of-the-art performance, meeting the objective, and provide fundamental biomedical word embeddings to be used by the HASKEE pipeline for downstream machine learning-based tasks. Future work may investigate the impact of integrating knowledge base information during embedding representation training.

Chapter 5 - Processing of PubMed articles:
Parsing, tokenization, abbreviation resolution, negation detection and sentence classification

5.1 Abstract

PubMed articles vary in structure based on journal and article type, requiring parsing and additional (pre)-processing for downstream analysis. In this module, we integrate and extend a PubMed parser to extract textual data, metadata and structural information. When an article is unstructured, we have trained a sequential neural model that is capable of predicting paper section by up to 90.9% accuracy, where method, results, and conclusion sentences are predicted with up to 90-96% accuracy. This model enables distinguishing sentences claiming novel association reporting from others citing previous findings. This work also resulted in the generation of a new 4 million annotated sentence corpus.

Following parsing, we perform sentence and word tokenization and resolve abbreviations by integrating the Schwartz and Hearst algorithm. The latter was also extended to resolve shortened species names. Finally, we implement a rule-based sentence-level negation detection approach to discriminate between negative and positive associations which achieved an F-score of up to 94%. Sentence prediction, abbreviation resolution, negation detection, and tokenization are pre-requisite steps for the downstream association extraction.

5.2 Aims and Objectives

Aims:

- To implement a pipeline module that parses, (pre-)processes and outputs articles in an adequate format for subsequent NLP tasks;

Objectives:

- To parse text and metadata for PubMed articles from the PubMed article files;
- To perform sentence and word tokenization, and resolve acronyms/abbreviations;
- To score articles based on parsed article metadata;
- To determine 'sentiment' of sentences by performing negation cue detection;
- To classify sentences from unstructured abstracts into paper sections (background, objective, results, methods, conclusion) for subsequent association ranking.

5.3 Introduction

PubMed® is a free citation repository launched in 1996 with over 28 million references, maintained by the National Center for Biotechnology Information (NCBI). It includes MEDLINE, PubMed Central and NCBI Bookshelf, with MEDLINE® being the largest subset of more than 25 million indexed biomedical and life science articles that constitute the National Library of Medicine® journal citation database. In addition to this, PubMed includes citations such as: in-process citations, "ahead of print" citations, pre-1966 citations, and out-of-scope citations, e.g. those concerning astrophysics and plate tectonics ("MEDLINE, PubMed, and PMC (PubMed Central)," n.d.).

These large resources of medical research and information are updated daily with new research, and published as an annual bulk baseline in the case of MEDLINE as downloadable XML files. This provides an accessible source of medical information that can, and has been, tapped into by text mining approaches. Extraction of such medical information is the initial step in any pipeline, with open source tools such as PubMed parser being available (Achakulvisut et al., 2016). PubMed parser allows to extract textual data for articles and basic metadata. However, additional information such as article structure and section/abstract headings are not parsed. In

this work this parser is extended to retrieve such information from the original XML files. In the case of abstract only articles (>90% of PubMed citations), unless structured, this information is also not available in the raw source files.

Knowing whether an abstract sentence is background information or is describing a novel result discovered/reported in the article would enable to assign different value to different parts of the publication. For example, consider the cases: (i) "we investigate the role of entity X in entity Y"; (ii) "we find that entity X is associated with entity Y"; and (iii) "it was previously reported that entity X is linked with entity Y". In case (i) this statement defines an objective and not a claim of a link between entity X and entity Y. Similarly, case (iii) does include a claim of an association between entity X and Y however this is not a novel claim but a reference to a previous claim. These two cases should be treated differently to case (ii) which reports an exclusive claim of a link between the entities. To our knowledge, this challenge has not been integrated in existing text mining tools and therefore we look at predicting the paper section for sentences from unstructured abstracts.

Related research work has focused on classifying sentences for evidence-based medicine in attempt to automate or facilitate tasks such as systematic reviews, where sentences are classified into PICO: population (P), Intervention (I), Comparison (C), and Outcome (O); or PIBOSO with the additional classes Background (B), Study Design (S) and Other (O). In the ALTA-NICTA challenge, highest-performing systems achieved 92-96% using SVM, stacked logistic regression, maximum entropy, random forests and CRF (Amini, Martinez, & Molla, 2012). More recent work devised a 200k abstract corpus and utilized sequential neural approaches to classify randomized clinical trial abstract sentences into: (i) Background; (ii) Objective; (iii) Methods; (iv) Result; and (v) Conclusions (J. Y. Lee & Dernoncourt,

2016)(Dernoncourt & Lee, n.d.). This achieved over 90% classification accuracy, however, leaves some outstanding questions:

(i) as PubMed/MEDLINE has over 28 million articles and randomized clinical trials are only a subset of this, how well is this model generalizable to non-randomized clinical trials?;

(ii) is performance limited by the quantity of the training data? Would adding more training data by extracting additional documents from PubMed itself improve performance?

Another aspect which has been much ignored by biomedical NLP pipelines is the differentiation of negative associations from false associations. In tools such as PolySearch2 (Y. Liu et al., 2015), a sentence not claiming an association (false association) is considered the same as one which claims a negative one. A negative association is an important claim that should not be considered as a "null" association. For example, in systematic reviews (as performed in Section 3.4.3) results are aggregated and weighted by how many citations claim such finding. Having a citation claiming the opposite questions the robustness and reproducibility of the results. In this work we recognize such importance and develop a negation-cue detection approaches by improving on the lexicon-based PolySearch2 approach (Y. Liu et al., 2015).

An additional challenge particularly with biomedical text is the extensive use of non-standardized abbreviations and acronyms, where a shortened sequence of characters is used to represent the full form. A number of abbreviation solving algorithms are available, with popular implementations such as the Schwartz and Hearst algorithm (Schwartz & Hearst, 2003) (see Section 2.5) and LINNAEUS (Gerner, Nenadic, & Bergman, 2009). The former was

reported to achieve 82% recall and 96% precision which overall outperforms previous work reporting 83% recall and 80% precision (J. T. Chang et al., 2002), and 72% recall and 98% precision (James Pustejovsky et al., 2004). However, unlike LINNAEUS implemented in Java, this algorithm does not resolve shortened species names. Therefore, in this work this implementation is extended and integrated with the HASKEE pipeline. Collectively, this module parses information and article structure from PubMed files, carries out trivial tokenization, resolves abbreviations, performs negation detection, and predicts which sections in the publication abstract sentences come from.

5.4 Methods

5.4.1 Parsing, Sentence and word tokenization

Extraction of text and metadata from PubMed articles was implemented by the integration of the pubmed parser library (Achakulvisut et al., 2016). Out of the box, this parses metadata information such as: abstract text, and paragraph text from full papers. As text location is used in the weighting of the extracted information/associations further down the pipeline, parsing was extended to extract additional information from the original XML files. For full paper articles and structured abstracts, section headings such as “Methods”, “Conclusion”, “Discussion”, and “Introduction” are extracted. MESH article types are also extracted and used later for weighting.

By default, Python’s NLTK 3.0 English punkt tokenizer model is used for sentence tokenization. Subsequently, word tokenization is also performed using NLTK’s implementation of TreebankWordTokenizer, by default. Prior to tokenization, however, contracted strings such as “can’t” and “it’s” are expanded with a curated list of contractions

that map to their expanded form. Finally, stopwords such as ("no", "it", "the") can be removed. A compiled list of such stopwords was obtained from (Brigadir, 2016/2018).

5.4.2 Acronym and abbreviation solver

As part of the pre-processing module, the Schwartz and Hearst algorithm (Schwartz & Hearst, 2003) was integrated for solving abbreviations and acronyms, as refactored for python 3 by (Gooch, 2017/2018). This implementation was modified to a class which takes an input text for identification of definitions and replaces identified acronyms by their full form. Furthermore, this algorithm was extended through regular expressions to solve shortening of species names. In such extension, in accordance with the standard binomial nomenclature, we consider a shortened binomial species name to be the first consecutive characters of the genus name in the binomial nomenclature system followed by a period and the species name. For example, for the species name "Pseudomonas aeruginosa", potential short forms considered a match would be: "P. aeruginosa" and "Ps. aeruginosa".

5.4.3 Negation cue detection

Negation cue detection is implemented at the sentence level in a number of (combinatorial) approaches as a filter function with a binary {True, False} output.

Approach 1: "un-" prefixed tokens return True unless token is in a curated list of exceptions compiled from http://morewords.com/starts-with/un/ (e.g. "unisex", "uniform");

Approach 2: tokens "no" and "not" return True;

Approach 3: Dependency parsed sentences with "neg" and "no"/"det" dependencies return True;

Approach 4: Adverb modifiers with a negative polarity as determined by NLTK and the VADER algorithm (Hutto & Gilbert, 2014), return True;

Approach 5: Any negation token from PolySearch curated list that was revised returns True;

Evaluation of negation cue detection was carried out on Polysearch (Cheng et al., 2008), GAD (Becker, Barnes, Bright, & Wang, 2004), DDIcorpus (Herrero-Zazo, Segura-Bedmar, Martínez, & Declerck, 2013), EU-ADR (Mulligen et al., 2012), and BioScope (Vincze, Szarvas, Farkas, Móra, & Csirik, 2008).

5.4.4 Negation scope detection

Bioscope was converted to IOB format, where tokens part of a scope were tagged with B-scope at the starting boundary and I-scope elsewhere. As the aim was to identify negation scopes only and this corpus contains also speculation cues and scopes, any sentences containing speculation cues and scopes only were ignored and skipped. Sentences which contained both speculation scopes as well as negation scopes were retained but only the negation scope was labeled.

Data was randomly split at the sentence level into train/dev/test with 60/20/20 splits. 10 different splits were generated for a 10-fold validation. The sequential tagging BiLSTM-CRF architecture was adapted for tagging scope tokens. Model training was performed on each training set and applied to the development sets. The number of epochs with the highest development accuracy was chosen and a final model was trained and applied to the test sets. Previously optimized word2vec and fastText embeddings from Section 4 were used and evaluated.

Evaluation was carried out by 4 metrics: (i) exact scope boundary matching: F-score based on matches were the lower and upper boundary of a scope matches perfectly the predicted one; (ii) token match: the F-score based on number of tokens assigned to belong to a scope; (iii) left

boundary match: F-score based on lower boundary matches; (iv) right boundary match: F-score based on right boundary matches.

5.4.5 Sentence classification

The NICTA-PIBOSO corpus presented at the ALTA 2012 was obtained by direct communication with the shared task organizer: Dr David Martinez. This was utilized for 2 classification problems: classifying abstract sentences into PIBOSO classes, and paper sections: objective, methods, results, conclusion. CRF models were trained with Mallet (McCallum, 2002) as in Section 6.4.8 using the features: lemmas, headings, POS tags, tokens, tokens_POS and the sentence position in the abstract.

Abstract sentence classification was also trained on 200,000 PubMed randomized clinical trial abstracts obtained from (J. Y. Lee & Dernoncourt, 2016), where abstract sentences are labeled into the classes: background, objective, method, result or conclusion. A BiLSTM-CRF neural network (see Section 2.3.2) was trained to perform sequential classification in a number of ways.

In the first trial, abstracts were padded to have equal number of sentences, sentences padded to have equal number of tokens, and the respective vectors were concatenated. In an alternative approach, sentences were represented by the mean word vector and null vectors were appended to short abstracts to maintain equal number of sentences. This resulted in a 3-dimensional matrix: (samples, maximum number of sentences, vector dimensions) = (190654, 51, 50). To mitigate appending with null vectors to maintain equal number of sentences, training and prediction was also run with a batch size of 1. In this case, uneven number of sentences are allowed as each abstract is trained and predicted independently.

Based on (Reimers & Gurevych, 2017), the architecture and hyper-parameters were optimized. The findings were used to choose the optimal: optimizer, gradient normalization, dropout, number of LSTM layers, number of recurrent units and epochs from the development dataset. Recurrent dropout was also implemented and tested, as suggested by (Gal & Ghahramani, 2015) and performance differences when using optimized word embeddings from Section 4 were determined.

Finally, to determine how generalizable the achieved accuracy is and whether this is limited by the training size, a new corpus was derived from all of PubMed's structured abstracts. Some sentences in structured abstracts are not assigned a heading and therefore these were ignored.

Training was performed on the corpus train split, optimizations carried out on the development set and final accuracy evaluated on the test set.

5.5 Results and Discussion

5.5.1 Acronym and abbreviation solver

Non-standard acronyms are commonly used in publications. Examples include:

Example 1: “Development of temozolomide (TMZ) resistance contributes to the poor prognosis for glioblastoma multiforme (GBM) patients. [...] could sensitize highly TMZ-resistant GBM tumors to TMZ." (S. Y. Lee, 2016)

Example 2: “[...] clinical risk factors associated with emergency room (ER) visits and diabetes-related hospitalizations." (Doubova et al., 2018)

In the first example, TMZ is defined as temozolamide, and glioblastoma multiforme is assigned the acronym GBM. These acronyms are used in the rest of the publication however are likely to vary between different publications as they are not standard acronyms. In the second example, whereas “ER” is used to represent emergency room, “ER” is also a commonly used acronym for “Endoplasmic Reticulum”, therefore introducing ambiguity. In order to recall an entity from its acronymic form and decrease ambiguity, this requires for the acronym to be converted to the identified definition. Therefore, by integrating the Schwartz and Hearst algorithm, the above examples are converted to:

Example 1 resolved: “Development of temozolomide resistance contributes to the poor prognosis for glioblastoma multiforme patients. [...] could sensitize highly temozolomide-resistant glioblastoma multiforme tumors to temozolomide." (S. Y. Lee, 2016)

Example 2 resolved: “[...] clinical risk factors associated with emergency room (emergency room) visits and diabetes-related hospitalizations.” (Doubova et al., 2018)

Performance of such abbreviation resolution by the Schwartz and Hearst algorithm has been previously evaluated, and is expected to achieve up to 96% precision and 85% recall (Schwartz & Hearst, 2003).

Shortened or abbreviated species names are also a common practice in medical publications. Unlike acronyms, species name are commonly shortened after the first mention without explicit indication of the name mapping. For example, “Staphyloccous aureus is studied here. S.aureus was cultured […]". The shortened version is not enclosed within parenthesis following the full form, as a typical acronym definition. Therefore, the Schwartz and Hearst algorithm is not able

to resolve this out-of-the-box. This algorithm was therefore extended through regular expressions to solve many possible forms of name shortening, such as:

Example 3: “Staphylococcus aureus (SA) [...]. S. aureus or SA was found to be [...]”
Example 3 resolved: “Staphylococcus aureus [..]. Staphylococcus aureus and Staphylococcus aureus was found to be [...]”.

Example 4 : “Staphylococcus aureus (St. aureus) is [...]. St. aureus and was found to be [..]”
Example 4 resolved: “Staphylococcus aureus is [...]. Staphylococcus aureus and was found to be [...]”

Example 5 : “Cyriocosmus elegans, Centruroides elegans, and Caenorhabditis elegans […]. Cy. elegans refers to Cyriocosmus elegans, Ce. elegans refers to Centruroides elegans while Ca. elegans refers to Caenorhabditis elegans”
Example 5 resolved: “Cyriocosmus elegans refers to Cyriocosmus elegans, Centruroides elegans refers to Centruroides elegans while Caenorhabditis elegans refers to Caenorhabditis elegans”

This improvement however does not resolve generic names such “humans” which refers to “Homo sapiens” and “yeast” referring to “Saccharomyces cerevisiae”, as performed in the LINNAEUS algorithm implemented in Java (Gerner et al., 2009). However, such alternatives do not classify as abbreviations or acronym and therefore do not fall in the scope of this task. Nonetheless, this synonymy is captured during dictionary graph compiling. We attempted to use a subset of the LINNAEUS corpus to evaluate the implemented extension to the acronym resolution of species names however the LINNAEUS corpus only provides mention level

entities and does not include the reference between the shortened species name to the normalized/full form.

5.5.2 Negation Cue and Scope Detection

Initially, negation detection filter was considered as a binary classifier to discriminate between positive and non-positive associations. Benchmarking on the PolySearch corpus, we recorded marginally improved performance with up to 3.31% mean accuracy increase over PolySearch implementation when using "no" and "not" tokens as detection tokens, achieving a global F-score across datasets of 90.90% (Table 21).

Table 21. Evaluation of negative association detection by negation cue detection filters on the PolySearch datasets.

	HASKEE			PolySearch		
	F-score	**Recall**	**Precision**	**F-score**	**Recall**	**Precision**
Drug/Adverse	0.992302	0.998592	0.986092	0.8585	0.8022	0.9233
Toxin/Adverse	0.989831	1	0.979866	0.7689	0.6822	0.8808
Toxin/Disease	0.978378	1	0.957672	0.8417	0.7864	0.9054
Disease/Gene	0.794233	0.918182	0.699769	0.8987	0.9238	0.875
Drug/Gene	0.821843	0.840782	0.803738	0.9179	0.8746	0.9658
Gene/Gene	0.929231	0.92638	0.932099	0.9379	0.9326	0.9432
Metabolite/Gene	0.857143	0.918367	0.803571	0.9074	0.8619	0.9579

Given precision is in most cases/datasets the lowest (compared to recall), this suggests that the filter is falsely identifying associations as positive. Similar results were achieved when evaluating on the DDIcorpus, achieving 98.26% recall and 23.44% precision (F-score of 37.86%), and the GAD corpus achieving 87.89% recall and 47.16% precision (F-score of 61.38%). These results imply there is room for improvement in the filter to capture negations. However, the limitation of such evaluation is that false and negative associations are treated the same. Here, a "false association" is considered as a sentence with at least a pair of entities where such sentence does not imply a link between the pair (positive or negative link). On the other hand, a "negative association" sentence is one which specifically claims that entities are

not linked; therefore, it does describe an association, just with negative polarity. When evaluating with the PolySearch dataset, where false and negative associations are treated the same, performance is highly dependent on the distribution of negative and false associations – where higher performance is expected by the negation filter (and PolySearch's implementation) when negative associations are the dominant type of non-positive associations, and vice-versa when false associations dominate the evaluation dataset.

As with HASKEE the aim is to retain negative associations as negative results/proof of no association, not discriminating between negative and false associations or treating them equally would be inadequate.

The GAD corpus (Becker et al., 2004) distinguishes between false and negative associations. As the negation filter is aimed at discriminating between positive and negative associations, we evaluated it on the GAD corpus subset of associations. This achieved 91.18% F-score (87.91% recall, 84.70% precision) when only unique sentences were considered. The corpus also contains multiple identical sentences which contain multiple annotations. Evaluating on this achieved 89.43% F-score (85.60% recall, 93.62% precision). These high performing results, in contrast with 61.38% GAD F-score when false and negative associations are considered as a single class, indicate that negation detection performs well and therefore detection of false associations is the performance bottleneck that requires improvement. This was confirmed by additionally evaluating false and positive associations from the GAD corpus which achieved 61.99% F-score (85.60% recall, 48.59% precision). On manual inspection of the misclassifications, incorrect annotations were observed. This suggests results may not be accurate, although quantification of such errors and inter-annotator agreement is to be calculated to quantify the impact of this. A Flask app was developed to aid the re-annotation

and validation of the dataset, however manual validation with such tool is to be carried out in future work.

Evaluation on the EU-ADR corpus (Mulligen et al., 2012) achieved similar high negation detection performance, recording 93.77%, 95.02% and 92.42% F-scores for drug-disease, drug-target and target-disease datasets, respectively, when discriminating positive associations from negative associations (Table 22). As EU-ADR also annotates speculative associations, when/if these are considered as positive associations, F-score decreased to 80.42%, 81.14% and 77.80% for drug-disease, drug-target and target-disease subsets, respectively. The latter was as a result of a decrease in precision (Table 22).

Finally, evaluation on the BioScope corpus (Vincze et al., 2008) was also run. BioScope only annotates speculative associations and negative associations and therefore for this purpose speculative associations were considered as positive associations. Performance was determined at 72.89% F-score for full papers and 80.17% F-score for abstracts (Table 23). This is similar performance to that determined for EUADR when speculative associations are considered (Table 22).

Table 22. Evaluation of the negation module based on the datasets from EU-ADR corpus. SA = speculative associations; PA = positive associations; NA = negative associations.

Associations considered	Dataset	F-score	Recall	Precision
SA/PA vs. NA	Drug-disease	0.804178	0.885057	0.736842
	Drug-target	0.81137	0.918129	0.726852
	Target-disease	0.777982	0.872428	0.701987
PA vs. NA	Drug-disease	0.937705	0.882716	1
	Drug-target	0.950166	0.910828	0.993056
	Target-disease	0.924205	0.887324	0.964286

Table 23. Negation evaluation with the BioScope corpus.

Dataset	F-score	Recall	Precision
Full papers	0.728856	0.864307	0.630108
Abstracts	0.801657	0.908579	0.717252

Preliminary negation scope detection with a BiLSTM-CRF architecture achieved up to 86.18% token match (Table 24). This is lower than the current state-of-the art of 90.3-92.1% (Hao Li & Lu, 2018). Comparing the use of different embedding models from Section 4, once again fastText outperforms word2vec even in the negation scope detection task.

Table 24. Negation scope detection accuracies achieved by the BiLSTM-CRF architecture under 4 evaluation methods: exact scope, token match, left margin match and right margin match.

Evaluation method	word2vec	fastText
Exact scope	70.65 ± 4.63%	71.54 ± 1.89%
Token match	85.65 ± 3.09%	86.18 ± 1.48%
Left margin	85.77 ± 2.31%	86.14 ± 1.44%
Right margin	78.03 ± 4.92%	79.88 ± 2.45%

Scope detection is expected to improve performance in general. Based on sentence-level negation detection results (Table 23), negation detection is mostly limited by precision i.e. false positives. This therefore predicts that restricting negation to a specific scope (compared to a whole sentence level) is likely to boost performance, although this is variable based on the benchmark dataset evaluated on.

5.5.3 Sentence Classification

Predicting the PIBOSO classes for NICTA-PIBOSO abstract sentences achieved an average F-score of 63% when training a CRF model with minimal feature engineering (Table 25). This is lower than challenge entries, however these were evaluated using ROC AUC where presumably class imbalance was not accounted for. Nonetheless, as predicting PIBOSO classes was not the aim of this work, we utilized the NICTA-PIBOSO dataset to train a section

predictor. As NICTA-PIBOSO corpus consists of structured and unstructured abstracts, we utilized the structured abstracts which have the headings: objectives, methods, results and conclusions as training data for a paper section classifier. With minimal feature engineering, this achieved a micro-average of 83.5%; with 89% for objectives, 83% for methods, 89% for results and 73% for conclusions.

Table 25. Abstract sentence classification into PIBOSO classes.

Class	F-score
Population	0.44
Intervention	0.15
Background	0.77
Outcome	0.86
Study design	0.69
Other	0.85
Micro-average	**0.63**

As for the neural model, predicting paper sections for abstracts with padded tokens and padded sentences, a total of 67% F-score was achieved. This is likely due to the fact that pre-padding or post-padding zeros to a sentence results in network memory loss due to the increase in distance from the last tokens, which are truly informative.

When sentences were represented as mean vectors and with a batch size of 1, this did not require token padding and allowed variable number of sentences per abstract. This achieved 88.91% F-score accuracy after a single epoch on the development set.

Consistent with the results from Section 4.5.5, fastText outperformed word2vec in both training (90.53% fastText vs. 90.40% word2vec) and validation accuracy (91.40% fastText vs. 90.94% word2vec). Therefore, fastText was used throughout this task.

Optimizing the number of epochs, number of layers, and number of hidden units achieved a final test accuracy of 90.9% with 3 BiLSTM layers run for 2 epochs with 50 hidden units per

layer (Figure 30). This lies close to the 91.6% reported by (J. Y. Lee & Dernoncourt, 2016) and exceeds the overall performance obtained by the CRF model. (Reimers & Gurevych, 2017) reported the optimal hyper-parameters for LSTM sequence tagging are: 100 hidden units per layer, 2 LSTM layers, dropout and Nadam optimizer. This hyper-parameter configuration did not achieve the highest performance here; reporting 90.25% classification accuracy on development at 2 epochs (compared to 90.9% achieved with the determined optimal hyper-parameter set).

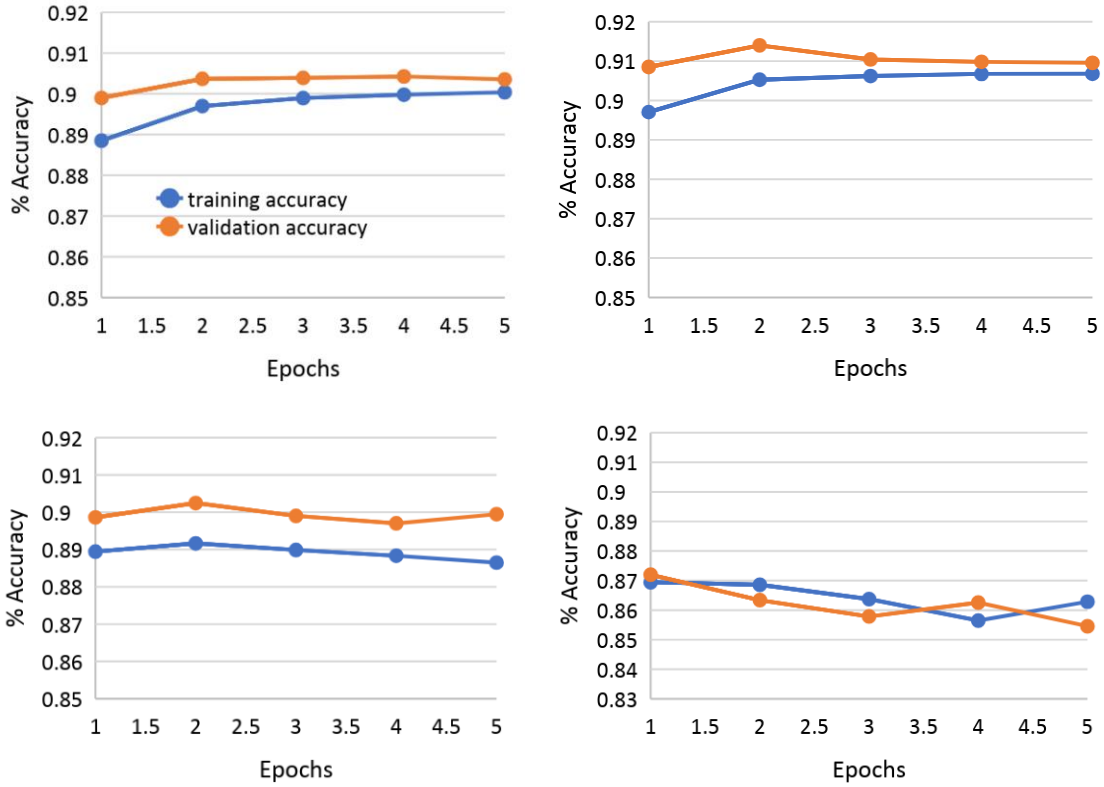


Figure 30. Training and development sequential sentence classification accuracies under different hyper-parameter and architecture configurations. A) 1 BiLSTM layer each with 50 hidden units with word2vec embeddings; B) 3 BiLSTM layers each with 50 hidden units with fastText embeddings; C) 2 BiLSTM layers with 100 hidden units (as reported by (Reimers & Gurevych, 2017)) using fastText embeddings; and D) 3 BiLSTM layers each with 50 hidden units with fastText embeddings on the ~4 million abstract corpus. A-C were trained on the 200k PubMed RCT corpus.

The new corpus constructed by extracting PubMed structured abstracts with NLM categories (i.e. objectives, conclusions, methods, background and results) totalled to 3.7 million abstracts. Model trained on the new corpus achieved an overall accuracy of 86.43% (Table 27). When

this model was applied to the test set of the 200k PubMed RCTs, this achieved 88.48% accuracy (Table 28). While this is lower than the 90.9% achieved when trained on the 200k RCT training dataset (Table 26), as this is likely to generalize better across broader type of publications, this model was utilized and integrated in the final pipeline.

Table 26. Per-class performance metrics (precision, recall, F-score and support) for sentence classification model trained on 200k PubMed RCTs.

Section	Precision	Recall	F-score	Support
Background	0.74	0.73	0.74	2661
Conclusions	0.96	0.93	0.94	4424
Methods	0.95	0.96	0.95	9748
Objective	0.72	0.71	0.72	2377
Results	0.94	0.95	0.95	10271
TOTAL	**0.91**	**0.91**	**0.91**	**29481**

Table 27. Per-class performance metrics (precision, recall, F-score and support) for sentence classification model trained on 3 million PubMed structured abstracts and applied to its equivalent test set.

Section	Precision	Recall	F-score	Support
Background	0.73	0.51	0.60	318324
Conclusions	0.92	0.92	0.92	553848
Methods	0.92	0.90	0.91	784010
Objective	0.60	0.79	0.68	288279
Results	0.91	0.93	0.92	1173504
TOTAL	**0.87**	**0.87**	**0.86**	**3117965**

Table 28. Per-class performance metrics (precision, recall, F-score and support) for sentence classification model trained on 3 million PubMed structured abstracts and applied to the PubMed RCT test set.

Section	Precision	Recall	F-score	Support
Background	0.77	0.47	0.58	2661
Conclusions	0.93	0.93	0.93	4424
Methods	0.94	0.93	0.94	9748
Objective	0.60	0.85	0.70	2377
Results	0.93	0.94	0.94	10271
TOTAL	**0.89**	**0.89**	**0.88**	**29481**

The per-class accuracies (Table 26-27) indicate that predicting background sentences is the poorest-performing class with F-scores ranging from 58-74%, followed by the objective class. Other classes were classified with accuracies exceeding 90%. Looking at the confusion matrices (Figure 31-31), this appears to be the result of misclassifications between background and objective sentences. Trends/misclassified classes are identical in all 3 models: train/test on 200k RCT (Figure 31), train/test on extracted 4 million abstract new corpus (Figure 32), and train on 200k RCT and testing on new corpus test set (Figure 33). In terms of absolute performance/misclassifications, background/objective sentences were misclassified mostly when the model was trained on the new structured abstracts corpus and applied to the RCT dataset test set, achieving as low as 47% accuracy with 48% of background sentences misclassified into the objective class. In the RCT train/test model, background was classified with 71% accuracy (Figure 31), therefore this suggests potential dataset preparation bias in the latter, particularly in the background sentences. This may be due to the highly variant vocabulary used in the background, whereas other paper sections are more consistent.

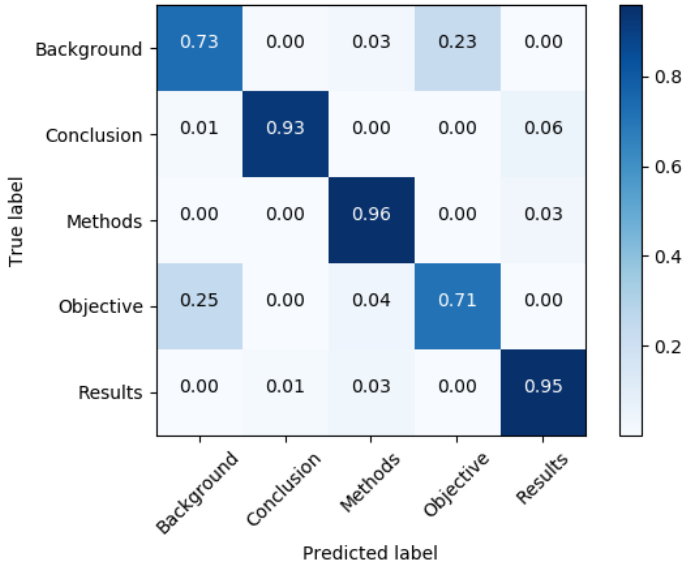


Figure 31. Confusion matrix for the sentence classification model trained on the PubMed RCT training set and applied to the PubMed RCT test set.

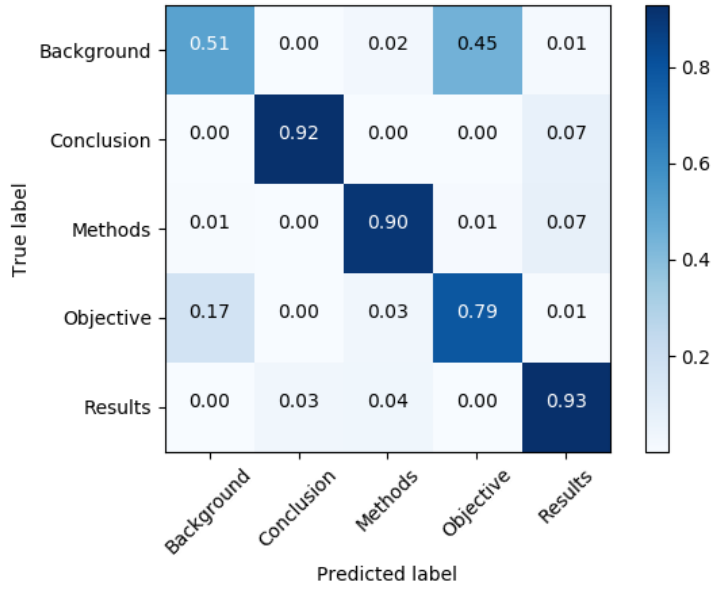


Figure 32. Confusion matrix for the sentence classification model trained on the PubMed structured abstracts training set and applied to the PubMed structured abstracts test set equivalent.

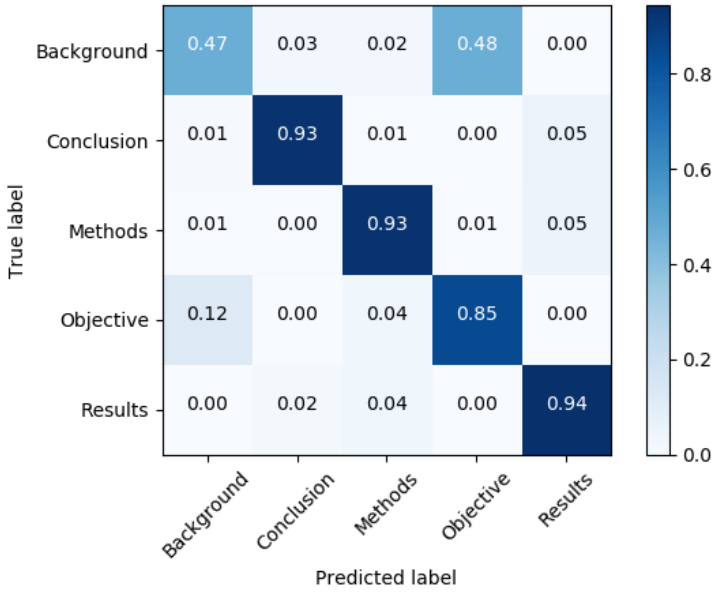


Figure 33. Normalized confusion matrix for the sentence classification model trained on the PubMed structured abstracts training set and applied to the PubMed RCT test set.

5.6 Conclusion(s) and Future Direction(s)

In this work, a module was developed to parse PubMed articles and structural information along with metadata. With this objective achieved, this enables downstream scoring of the articles. Such module also performs trivial pre-processing such as sentence and word

tokenization as pre-requisite steps for downstream tasks. As per the findings in (Barrett & Weber-Jahnke, 2011; Cruz Diaz & Maña López, 2015), training specific biomedical sentence/word tokenizers might improve pre-processing quality and therefore can be the subject of future work.

Through extending and integrating an abbreviation resolution system that identifies shortened species names, this work also enabled and improved on recalling associations between entities that have been abbreviated, while avoiding ambiguity if shortened species names were considered as synonyms at the dictionary level. Future work can quantitatively evaluate and subsequently integrate additional acronym abbreviation resolvers.

By implementing a rule-based negation detection approach that achieves high performance when compared to existing methods, this allows HASKEE to negatively score associations. Preliminary work on negation scope detection for more fine-grained negation detection shows promising results for future research and integration.

Finally, an optimized sequential sentence classifier was developed for paper section prediction that achieves quasi state-of-the-art performance. This enables HASKEE to give more weight to associations found in “results” sections as compared to “background” information citing previous findings. A more generalizable model, compared to current published models, was trained here and is therefore more appropriate for PubMed-scale translation. As a result of this, we also present a new corpus for PubMed sentence classification with about 4 million abstracts.

Chapter 6 – Implementation of biomedical named entity recognition methods and power analyses

6.1 Abstract

Identifying entities such as genes, chemicals and drugs from unstructured text is the information extraction task referred to as named entity recognition (NER). NER can simplistically be performed by string matching text to a curated list of terms. By making use of and extending the trie data structure, we integrate longest string match and nested string match as part of the HASKEE pipeline. This is coupled with a graph-based dictionary and database that serves as an inter-map between 13 difference sources which we compiled for the bioentities: species, genes/proteins, toxins, drugs, metabolites, anatomy, disease, fungi, food and food compounds. This graph-based dictionary approach enables to capture hyponymy and hypernymy between entity terms.

Modern NER approaches utilize machine learning (ML) methods based on supervised machine learning techniques. These have achieved state-of-the-art performances on standardized benchmark datasets. However, evaluation may be optimistic as models may not generalize well and are therefore not reflective of true performance when utilized in tools for large-scale processing of documents such as PubMed. Here, biomedical corpora published to date were collated, overlapping entities identified, and models were trained and cross-validated between different datasets in a number of approaches. Models trained on single corpora are found to perform perform poorly when predicting other corpora, likely due to annotation standard differences as well as model overtraining. By combining multiple corpora for training, it is shown that model are more generalizable, while achieving reasonably comparable performance. Performing power analyses, it is further demonstrated that the current quantity of annotated training data is not the limiting factor for higher performance.

6.2 Aims and objectives

Aim(s):

- To integrate/implement string matching algorithms for dictionary-based named entity recognition and develop a dictionary of biomedical entities;
- To benchmark machine learning-based biomedical named entity recognition models, their generalizability, and determine the effect of training quantity on performance;

Objective(s):

- To integrate existing dictionary-based named entity recognition approaches as part of the HASKEE pipeline and implement alternative approaches;
- To build a queryable graph-based dictionary that merges different sources while retaining granularity and source information to serve as an inter-map between sources;
- To collate training data from different sources for assessing machine learning-based approaches;
- To perform power analyses and determine if training data is the bottleneck in performance;
- To determine machine learning-based model generalizability across different datasets by:
    - (i) training models on single source data and testing on all other data;
    - (ii) training models on multiple data sources except one and predicting the left-out corpus (leave-one-out cross-validation);
    - (iii) training models on all training data and predicting test data for all sources.

6.3 Introduction

Named entity recognition (NER) is an information extraction task that identifies named entity mentions in unstructured text. String-matching approaches are simple yet effective means of

performing NER, utilized by pipelines such as PolySearch2 (Y. Liu, Liang, & Wishart, 2015b). In PolySearch2, string matching is performed using the ElasticSearch technology which is reputable for its speed and scalability, and is widely used in industry for big data. ElasticSearch enables recalling text which matches entities of interest (and the respective synonyms) in quasi-real-time, and has the capability of performing fuzzy matching to account for syntax errors. ElasticSearch has highly versatile and advanced string-matching algorithms, and is adequate for performing closed-discovery searches where both entities of interest are known, as it is able to match, extract and score the text that contains both entities. Aiming towards developing a platform that can also perform open-discovery search, where all other entities potentially associated/co-occurring with an entity of interest are recalled, ElasticSearch and real-time search techniques would be computationally inefficient and inadequate.

Pre-identifying entities in text requires a one-time processing expense, allowing for faster subsequent searches as well as computing an association graph; the ultimate goal of this project. Therefore, in this module we utilize and extend string matching algorithms such as the trie data structure (Section 2.6) to achieve this.

String matching to a list of terms and synonyms is a trivial task, however none of the existing solutions to our knowledge handle “hypo-” and “hypernymy”. For example, while “cancer” is synonymous to “(malignant) tumour”, it is a more generic term (hypernym) to “breast cancer”. Similarly, “breast cancer” and “colon cancer” are co-hyponyms. Therefore, when querying for associations with “cancer”, all related hyponyms (more specific terms) need to be recalled, however differentiation between the hyponyms needs to be retained as querying specifically for “breast cancer” should not recall “colon cancer”. In our approach, we develop a graph-based dictionary that retains this information.

Beyond dictionary-based approaches, machine learning methods have been developed and studied extensively to perform NER (Lample et al., 2016; L. Liu et al., 2017; Ma & Hovy, 2016; Peters, Ammar, Bhagavatula, & Power, 2017; Yang, Salakhutdinov, & Cohen, 2017; Ye & Ling, 2018). Such approaches provide a number of advantages over dictionary-based approaches, such as: identifying new entities that have not been reported/seen before, and disambiguating words based on context or semantics. This is however at the expense of requiring training data, which may introduce new limitations; as analyzed and identified here.

In the biomedical domain, the GENIA project (J.-D. Kim et al., 2003) was amongst the first major efforts for the development and optimization of machine learning-based named entity recognition systems for bioentities, creating the GENIA corpus and initiating the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004) (Galea, Laponogov, & Veselkov, 2018b; Jin-Dong Kim et al., 2004). Initial results for DNA, RNA, cell line and cell types were reported at 72.55% accuracy using HMM and SVM models (GuoDong & Jian, 2004) (see Section 2.2). Similar models and architectures were used in the BioCreAtIve challenges, where for the GENETAG corpus, a maximum F-score of 83.2% was reported by (Yeh, Morgan, Colosimo, & Hirschman, 2005). Since then, several publications have reported equivalent or improved performance, and several NER tools are currently available (Campos et al., 2013b; Finkel et al., 2005; McCallum, 2002; Settles, 2005). This improvement can mostly be attributed to efforts in improving the feature selection and extraction.

Using CRF-based models, the open-source NER tool Gimli (Campos et al., 2013b) reported 72.23% F-score for the JNLPBA corpus and 87.17% for the GENETAG corpus. This is comparable to the highest reported accuracies for these corpora with closed source software,

where NERBio (Tsai et al., 2006) report 72.9% for the JNLPBA corpus while (Hsu et al., 2008) report 88.30% for the GENETAG corpus (Galea et al., 2018b). These results for JNLPBA are also similar to others reported in literature (GuoDong & Jian, 2004; Rei, Crichton, & Pyysalo, 2016). Genes and diseases were also reported to be identified with over 90% F-score by the NER module of DTMiner (Xu et al., 2016). However, training and evaluation of the latter was performed on a custom corpus (Galea et al., 2018b).

More recently, with the increase in computational power and open-source deep learning platforms such as tensorflow, pytorch, and keras, neural networks are heavily researched and optimized to perform the NER task. To mention a few, studies such as (Crichton, Pyysalo, Chiu, & Korhonen, 2017), (Gridach, 2017), (Zeng et al., 2017) have applied neural approaches (see Section 2.3) to biomedical NER corpora. At the time of writing, BiLSTM-CRF (see Section 2.3.2) provides state-of-the-art performances. Using this architecture, (Dang et al., 2018) reported a 93.14% F-score for chemical recognition in the BC5CDR corpus, (Lou et al., 2017) reported 86.23% F-score for diseases in the BC5CDR corpus, and (Habibi, Weber, Neves, Wiegandt, & Leser, 2017) reported 84.64% F-score for disease recognition in the NCBI Disease corpus (Doayan, Leaman, & Lu, 2014).

With the efforts on developing new machine learning approaches and architectures, performance has incrementally increased throughout the years. However, despite this ever-increasing promising performance, a number of investigations are outstanding and require to be addressed for more informed next steps in the field:

i) how generalizable and robust are the models?;

ii) how realistic and translatable is the performance evaluated?;

iii) does the size of the training data limit the performance?; and consequently,

iv) would acquiring more data improve the results further?

Commonly, individual corpora such as GENETAG or GENIA are used to train and test machine learning-based NER models, resulting in models that are optimized to such individual corpora, consequently introducing potential bias and overfitting. This reduces the models' generalizability when applied to unseen text. In the work by (Campos et al., 2013b), when NER for genes and proteins was trained on the GENETAG corpus and tested on the CRAFT corpus, a 45-55% F-score was achieved; up to 45% lower performance compared to the GENETAG test F-score of 87.17% by Gimli (Campos et al., 2013b). There are a number of potential factors contributing to such performance discrepancies including differences in the quality and annotation standards between the different corpora. However, the variability in the style of writing of unseen text is also likely to increase when compared to the much smaller corpora, and thus the accuracy quoted for the models may not be representative (Galea et al., 2018b). The difference between gold-standard performance and translational performance has indeed been previously shown for mutations (Caporaso et al., 2008).

In (Galea et al., 2018b), we recognize that several available corpora share the same/related entity classes: OSIRIS (Furlong, Dach, Hofmann-Apitius, & Sanz, 2008), SNPcorpus (Thomas, Klinger, Furlong, Hofmann-Apitius, & Friedrich, 2011), BioInfer (Sampo Pyysalo et al., 2007), various BioNLP 2011 subsets (Sampo Pyysalo, Ohta, Rak, et al., 2012), CellFinder (Neves, Damaschun, Kurtz, & Leser, 2012), GETM (Gerner, Nenadic, & Bergman, 2010), IEPA (Ding, Berleant, Nettleton, & Wurtele, 2002), HPRD50 (Fundel et al., 2007), GREC (Thompson, Iqbal, McNaught, & Ananiadou, 2009) and GENIA (J.-D. Kim et al., 2003) all contain gene/protein-related entities; GENIA (J.-D. Kim et al., 2003), CellFinder (Neves et

al., 2012) and AnEM (Ohta, Pyysalo, Tsujii, & Ananiadou, 2012) contain cell line/type and tissue information; BioNLP2011 (Sampo Pyysalo, Ohta, Rak, et al., 2012), DDI corpus (Herrero-Zazo et al., 2013) and GENIA (J.-D. Kim et al., 2003) share chemical/drug entities; and GENIA (J.-D. Kim et al., 2003), GREC (Thompson et al., 2009), CellFinder (Neves et al., 2012) and BioNLP2011 ID (Sampo Pyysalo, Ohta, Rak, et al., 2012) contain annotated species terms. Despite the common entities, availability of such data is very dispersed and formats and not standardized, varying from CONLL, to BioC, leXML and several others. As a result, these resources have not been collectively exploited. Here, as published in (Galea et al., 2018b), biomedically-related corpora currently available are collated, standardized to the BioC format (Comeau et al., 2013), corpus-independent models are generated and the role of the data size on NER performance is assessed; a task referred to as power analysis. We therefore show how representative (and generalizable) the calculated metrics are if current algorithms were to be applied in a large-scale corpus such as PubMed.

6.4 Methods

6.4.1 String matching implementations

HASKEE incorporates 2 dictionary-matching approaches: longest string match and trie-based matching. Longest string match was initially developed by matching tokens sequentially to a hierarchical python dictionary, however since the release of FlashText, this has been replaced with the FlashText (Singh, 2017) library. Secondly, the trie-based implementation was developed by utilizing the pygtrie python library (*Python library implementing a trie data structure.*, 2014/2018) implementing the trie data structure in python. This was utilized and modified so that each node represents a token. The list of terms from Section 6.4.2 was used to build such a trie graph.

Given a list of sentence pre-processed tokens from Chapter 5, the implementations return the keywords that match with the trie dictionary and their corresponding position indices (start and end boundaries). These are appended to the JSON file and a new tagged JSON file is returned on completion of the NER module. All possible entity matches are saved for a given sentence, irrespective of overlap. Overlap is taken into account during the computation of the associations (see Section 7.4.1).

6.4.2 Vocabulary compilation

Keywords were compiled from 13 sources: (i) CHEBI (Degtyarenko et al., 2008); (ii) Catalogue of Life (Roskov et al., 2013); (iii) Entrez (Maglott, Ostell, Pruitt, & Tatusova, 2005); (iv) T3DB (Lim et al., 2010); (v) EMA ("Medicines | European Medicines Agency," n.d.); (vi) DrugBank (Wishart et al., 2008); (vii) ECMDB (Sajed et al., 2016); (viii) YMDB (Jewison et al., 2012); (ix) HMDB (Wishart et al., 2018); (x) MESH (ROGERS, 1963); and (xi) ICD10 (World Health Organization, 1992, p. 10); (xii) Mycobank (Robert et al., 2013); and (xiii) FooDB (Scalbert et al., 2011). These cover the entity classes: metabolite, gene/protein, species, chemical, toxin, drug, disease, food, food compounds and anatomy. As more specific subsets, these cover: E. coli metabolites, yeast metabolites, and human metabolites. From each of these sources, the following terms were extracted and compiled:

- **CHEBI**: IUPAC names, compound names, traditional names, INCHI key, brand name;
- **Catalogue of Life**: taxonomy (kingdom, phylum, class, order, superfamily, family, genus), species names, vernicular (common names);
- **Entrez**: symbol, name, synonyms;
- **EMA**: drug name, common names, and previous drug name;
- **DrugBank**: drug name, synonyms, product names, classification (direct parent, alternative-parent, kingdom, superclass, class, subclass);

- **T3DB**: common name, IUPAC name, traditional IUPAC name, synonyms, taxonomy (direct parent, kingdom, superclass, class, subclass, hmdb class, alternative parents);
- **ECMDB**: name, IUPAC, INCHI key, SMILES;
- **YMDB**: name, IUPAC name, traditional IUPAC name, synonyms;
- **HMDB**: metabolite name, traditional IUPAC name, IUPAC name, SMILES, INCHI key, classification (direct parent, kingdom, superclass, class, subclass, alternative parents);
- **MESH**: anatomy and diseases ontologies and their respective concept names and term lists;
- **ICD10**: chapter names, section class name and subsequent headings;
- **Mycobank**: taxon, current name;
- **FooDB**: food ontologies, name, scientific name, and compound names;

Pre-processing of these terms involved replacing double spacing with single spacing in synonym names, removing within token new line delimiters, and stripping new line delimiters, tabs and dashes. For Catalogue of Life, additional synonyms were computed to obtain variants for each entry that conform with both International Commission on Zoological Nomenclature (ICZN) ("ICZN," 1999) and International Code of Nomenclature (ICN) (Turland et al., 2018) codes. This was performed by concatenating: genus, subgenus, specific epithet, and intraspecific epithet, when available. Additionally, we constructed a synonym with genus, specific epithet and (author). Shortened name versions such as G. species (where "G." is a shortened representation of the respective Genus) were considered but are handled in the preprocessing module (refer to Section 5.4.2).

6.4.3 Compiling dictionary graphs

The vocabulary list compiled in Section 6.4.2 provides no information on the relationship between terms such as synonymy or hierarchical relationship in the case of species names belonging to a genus name. To retain this information, we built a graph database dictionary with nodes for term names and identifiers and edges to indicate synonymy (such as "AMBIGUOUS SYNONYM", "BRAND NAME", "IUPAC NAME", "NAME", and "SYNONYM"), ancestory ("IS A") and mapping from the string name to the knowledgebase identifier ("ID"). These edges depended on the information available from the original source, with CHEBI and Catalogue of Life providing most fragmented information (Table 29), whereas other sources such as Entrez, EMA, and YMDB only included mapping from the term name to their respective database identifier (Table 29). An examplary graph structure is shown in Figure 34 for the term "caffeine" from CHEBI. The same pre-processing and cleaning of the terms was carried out as in Section 5.4.1.

Table 29. Dictionaries compiled from different sources as graphs. Ontological relationships and synonymy information is retained through respective edges. Terms are represented by name nodes and their respective source identifier.

Source	Nodes	Relationships	Entity Class(es)
CHEBI	name, id	SYNONYM, NAME, IUPAC_NAME, IS_A, INN, ID, BRAND_NAME	Chemical
Catalogue of Life	name, id	ID, IS_A, AMBIGUOUS_SYNONYM, MISAPPLIED_NAME, SYNONYM	Species
Entrez	name, id	ID	Gene/protein
T3DB	name, id	ID, IS_A	Toxin
EMA	name, id	ID	Drug
DrugBank	name, id	ID, IS_A	Drug
ECMDB	name, id	ID	Metabolite, E.coli metabolite

YMDB	name, id	ID	Metabolite, Yeast metabolite
HMDB	name, id	ID, IS_A	Metabolite, human metabolite
MESH	name, id	SYNONYM, ID, IS_A	Anatomy, Disease
ICD10	name, id	ID, IS_A	Disease
Mycobank	name, id	ID, IS_A	Fungus
FooDB	name, id	ID, SCIENTIFIC_NAME, IN_FOOD, IS_A, SYNONYM	Food, Food compound

The primary identifier node for each knowledgebase that maps to the string name was defined as the primary knowledgebase accession number. In the case of The Catalogue of Life, were additional synonym terms were computed, these were assigned a new identifier prefixed with 'COL:' term. The primary identifier used for this knowledgebase was the acceptedNameUsageID. In knowledgebases such as T3DB where ontologies and taxonomy is also provided, parents and alternative parents were considered as ancestral nodes.

Graphs for each source were built using the NetworkX python package (Hagberg, Schult, & Swart, 2008) and using a custom script exported to csv for building a queryable graph database.

Ultimately, a single merged graph was constructed to allow cross-mapping between knowledge-bases. This was built by a function that stacks the individual graphs, while checking if terms already exist from previous knowledge-bases to avoid duplicates. Based on the source, each term was assigned a class (Table 29). Where an entity is found in more than one knowledgebase, the entity classes were appended (therefore if a metabolite term is found in both ECMDB and HMDB, this is assigned both the label "human metabolite" and "E. coli metabolite"). Similarly, each node is assigned the attribute "source" which holds the dictionary source; allowing for filtering by source in downstream querying.

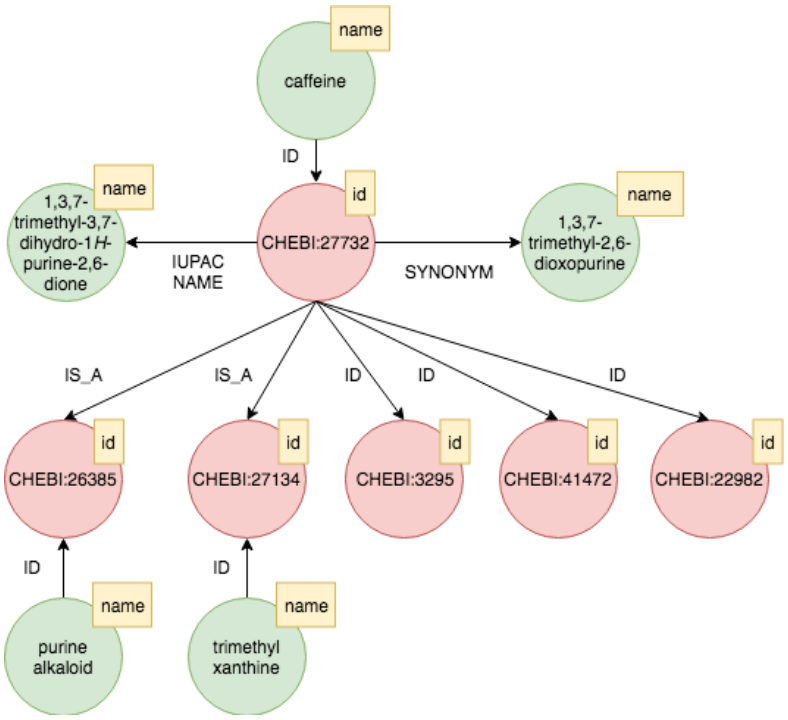


Figure 34. Dictionary graph structure example for the term "caffeine" from CHEBI. String terms are assigned the node type "name" and match the keyword list compiled in Section 6.4.2. This is linked to an "id" node which contains the database accession id for the term through an "ID" edge. Synonymous terms and IUPAC name for "caffeine" are linked through the respective edges. Secondary accession identifiers such as "CHEBI:3295", "CHEBI:41472", "CHEBI:22982" are linked to the primary id through another "ID" edge. Ontologies for the term "caffeine" are assigned by linking the primary accession id through "IS A" relationships to the primary accession id of the ontological term. In this example, the terms "purine alkaloid" and "trimethyl xanthine" are two ontological terms with "CHEBI:26385" and "CHEBI:27134" as their primary identifiers, respectively. These are both ontologies for "CHEBI:27732" (caffeine) and therefore are linked by an "IS_A" relationship.

6.4.4 Visualizing dictionary graph(s)

To visualize (and validate) the network population script and structure, D3 (Bostock, Ogievetsky, & Heer, 2011), Cytoscape (Shannon et al., 2003) and Gephi (Bastian, Heymann, & Jacomy, 2009) network visualization libraries were tested. Ultimately Gephi was determined the most adequate for visualizing networks such as HMDB (the largest network). As HMDB provides taxonomic classification of each metabolite but only up to the first 5 hierarchy levels, the chemical ontologies were obtained from classyfire (Djoumbou Feunang et al., 2016). The

ontology network was initially built and the HMDB metabolites were added to it, including the synonyms. Secondary accessions/ids were also taken into account.

6.4.5 Building dictionary graph(s) databases

The constructed graph was exported to csv; a format suitable for import in a neo4j graph database.

Population of the database was performed using the bulk csv importer built-in in the neo4j console through execution of the following command:

```
SET DATA=parent\\directory\\to\\csv\\files
.\\bin\\neo4j-admin import ^
--mode=csv ^
--id-type string ^
--database=database_name.db ^
--nodes %DATA%\nodes_ids.csv ^
--nodes %DATA%\nodes_names.csv ^
--relationships %DATA%\edges_AMBIGUOUS_SYNONYM.csv ^
--relationships %DATA%\edges_BRAND_NAME.csv ^
--relationships %DATA%\edges_ID.csv ^
--relationships %DATA%\edges_INN.csv ^
--relationships %DATA%\edges_IS_A.csv ^
--relationships %DATA%\edges_IUPAC_NAME.csv ^
--relationships %DATA%\edges_MISAPPLIED_NAME.csv ^
--relationships %DATA%\edges_NAME.csv ^
--relationships %DATA%\edges_SYNONYM.csv
```

Code Snippet 1. Population of a neo4j database with the dictionary graph.

To optimize look-up speed of terms, nodes and their attributes were indexed. A hash list is created for each unique term and attribute value for constant time look-up; where querying time becomes independent of the number of dictionary terms.

6.4.6 Compiling training data

Biomedically-related corpora were collated from primary and/or secondary sources. Irrespective of the format, corpora annotating biomedical entities or entity relationships such as drug/drug or protein/protein interactions, were considered in this work if the entities were specifically annotated. Annotations without indices or multiple nested entities were excluded.

As newly published corpora commonly increase data by annotating further examples and adding to previously published corpora, in such instances the subset corpus was excluded while the superset corpus was used to avoid biases. For example: MLEE (Sampo Pyysalo, Ohta, Miwa, et al., 2012) and AnEM (Ohta et al., 2012) are subsets of the bigger AnatEM corpus (Sampo Pyysalo & Ananiadou, 2014), thus were excluded.

6.4.7 Corpora pre-processing

Machine learning-based NER algorithms and packages such as Stanford's NER module (Finkel et al., 2005) require the data to be in IOB2 format. In such format, the initial entity token is labeled as B-LABEL (B for "Beginning"), internal entity tokens in a multi-phrase entity are labeled as I-LABEL (I for "Inside") (Krishnan & Ganapathy, 2005), while all other null tokens are labeled as "O". This format however requires text to be tokenized and therefore data is tokenizer-dependent. Furthermore, as the data is expected to be a single token per line, with a tab-separated entity label, additional article information is not retained. For further validation, transparency and to retain all information from the original corpus including document source, passage and annotation position, selected corpora were converted to an intermediate BioC format (Comeau et al., 2013). Corpora in the BRAT format were converted using the Brat2BioC Java module (A. M. N. Jimeno Yepes & Verspoor, 2013).

Annotation string position indices included in the BioC format were checked and corrected automatically when the offset/mismatch was between 1-5 characters, whereas larger offsets were manually validated and corrected (Galea et al., 2018b). Subsequently, corpora were converted to IOB2 with a custom script utilizing the python pyBioC library (Marques & Rinaldi, 2013). Unless a custom DTD (Document Type Definition) was used and provided by the corpus (as for tmVar corpus (Wei et al., 2018)), the default DTD was used to process BioC

documents. As part of the conversion, following sentence tokenization, word tokenization was carried out using the NLTK regular expression tokenizer with the expression: \\w+|[\S\w]". This was chosen over other python NLTK tokenization methods as 'TreebankTokenizer', 'WordPunctTokenizer', 'PunktWordTokenizer', and 'WhitespaceTokenizer' as these contract some (or all) of the punctuation, creating a token with embedded punctuation which does not match the entity/annotation when the annotation is part of a punctuated token. For example: "[…] gene X-associated […]" tokenizes to "gene" and "X-associated" using "TreebankWordTokenizer" and "PunktWordTokenizer", however the gene entity in this case is only "X" or "gene X". In the case of a terminal entity ("[…] gene X."), the punctuation is contracted using the "PunktWordTokenizer" (Galea et al., 2018b).

As different corpora annotate entities by various labels, to allow for merging different corpora we devised and assessed 8 initial new classes based on ontologies: (i) Chemical/Drug; (ii) Gene/Protein/Variant; (iii) Cell; (iv) Anatomy; (v) Organism; (vi) Tissue; (vii) RNA; and (viii) Disease. Classes such as tissue, cells, organisms, anatomy and diseases are predicted to be well-recognized using dictionary matching approaches due to their consistent nomenclature; with organisms/species requiring mandatory registration upon discovery, and diseases documented in registers (Galea et al., 2018b). Therefore, these classes were not considered further for machine learning training/prediction.

On the other hand, classes such as drugs and chemicals, genes, RNA, proteins and particularly mutations, are highly variable entities. This results in a greater chance for an entity not to have been mentioned/documented before and therefore highly appropriate to recognize with machine learning methods (Galea et al., 2018b).

Different stratifications of the initial super-classes were devised and tested by generating different versions of the corpora with one entity class per corpus copy. The remapped classes, original entity classes and respective number of entities per corpus entity classes are shown in Table 32.

6.4.8 Model training and prediction

We trained the models using the Stanford NER CRF algorithm based in Java (Finkel et al., 2005). A python wrapper was developed to allow executing the shell commands (Code Snippet 2).

Training:
javacp stanford-ner.jar;lib/*;. edu.stanford.nlp.ie.crf.CRFclassifier prop trainProp- File

Prediction:
javacp stanford-ner.jar;lib/*;. edu.stanford.nlp.ie.crf.CRFclassifier -loadClassifier trainedModel prop testPropFile

Code Snippet 2. Shell commands to train and apply a CRF classifier using the Stanford NER tools.

Training and test data were provided by listing their directories in the .prop file (properties file) as an alternative to providing the directories as the input argument in the command line. The former option was chosen as command line imposes a character count limit which is exceeded when the number of files is large.

The Stanford NER CRF package was also used for feature extraction, including: capitalization, symbols, disjunction and word tag features. This choice of features was based on feature assessment through backward elimination by (Campos et al., 2013b).

6.4.9 Power analyses

Power analyses was performed by generating learning curves to determine the effect of training data size on performance. Corpora were split into 80/20 training/test, with samples from the same publication considered in the same split (training or test) to avoid model bias. Performance was measured by the F-score (Equation 29) which was calculated for each corpus individually. To provide a single overall metric, two statistics were computed: (i) a document-weighted average was also calculated, where the F-score from each corpus is weighted by the number of documents it contains to compute an overall average; and (ii) an equally-weighted mean where corpora contributed equally to the overall average (Galea et al., 2018b).

$$Fscore = 2 \cdot \frac{precision \cdot recall}{precision + recall} \qquad \text{Equation 29}$$

By varying the corpus sizes, training, and predicting the test set, the performance-dependence on training size was quantified. Learning curves were in turn represented by an inverse power law function (Figueroa, Zeng-Treitler, Kandula, & Ngo, 2012), where the prediction F-score (Yfs) is defined as a function of the product of training sample size and minimum achievable error (a), learning rate (b) and decay rate (c) (Equation 30). An initial decay rate of -0.1 and learning rate of 0.2 were used. Error was defined as the root mean squared error (RMSE) (Galea et al., 2018b).

$$Y_{fs}(x) = f(X; a, b, c) = (1 - a) - b \cdot x^{c} \qquad \text{Equation 30}$$

3 approaches were used to generate learning curves (Galea et al., 2018b):

- Corpus-specific training: training was done per corpus and the model was used to predict the test split of the other corpora to determine model generalizability;

- Merged corpora training: the training data from all the corpora was merged by incremental addition while the same fixed test set was used. This determined the added value of increasing data size by adding more corpora;
- Leave-corpus-out training: a model was trained using all corpora except for one corpus and applied to the test data of the left-out corpus.

Varying data size was carried out in 2 approaches. (i) In the leave-corpus-out scenario, when all corpora except one corpus are used for model training and the left-out corpus is fixed test data, the latter was a random subset of all documents from each left-out corpus (Galea et al., 2018b). Data from the same corpus were excluded from model training to avoid biasing the analysis. The training set was constructed by merging and randomly shuffling the documents. The left-out-corpora were predicted one at a time. (ii) When all corpora were used for training, corpora were added sequentially. This allowed to establish the added predictive value of each corpus.

6.4.10 Orthographic feature analysis

To gain insight into which orthographic features are contributing to the predictive power, univariate tests comparing entity classes features were executed. Feature extraction was carried out on each entity class tokens whereas the rest of the tokens were considered as 'nulls'. Random sub-sampling was done on the bigger class to balance the classes. Regular expressions and the GIMLI (Campos et al., 2013b) package were utilized to extract 31 morphological features, including: punctuation, case sensitivity (initial caps, end caps, all caps), digits (number of digits) and number of characters (Galea et al., 2018b).

Features were represented as a binary representation, with the summation indicating total occurrence of a feature in the 'null' class and 'entity' class (Galea et al., 2018b). Statistical

significance was determined by Fisher exact tests followed by Benjamini-Hochberg FDR-correction for multiple testing. This was repeated n-times [where n = size(largest class)/size(smaller class)] and a mean q-value standard deviation was computed for each feature. When a feature was determined to be significantly different between the classes ($q < 0.05$; including the upper deviation boundary), and the percentage difference was positive between the entities class and the null, such feature was considered characteristic of that class (compared to null tokens).

6.5 Results and Discussion

6.5.1 Dictionary graph

Commonly, matching strings with synonyms is performed by replacing all synonym terms with a standard unique identifier or a primary term. This is possible with algorithms such as FlashText where multiple synonyms are assigned a primary key and if found are replaced with such primary key. While this is computationally efficient, this normalization results in the loss of information on the original term. With some knowledgebases being curated, while others being populated automatically, the level of noise for keywords and synonyms varies. Therefore, if all knowledgebase terms are merged and mapped to a single standard identifier, this loses the ability to filter out some of the noisy synonyms, or unselect a specific knowledgebase which may have noisy synonyms for a specific term. With the proposed graph structure, we maintain the highest fragmentation possible while enabling to: (i) recall the identifier for a string term from a specific database; (ii) recall the identifiers for all knowledgebases from one of the knowledgebase identifiers (identifier inter-map between knowledgebases); (iii) recall synonyms for a given term from selected knowledgebases; (iv) recall synonyms based on mapping confidence. In the latter capability, as identifiers may map to other alternative identifiers and ultimately the term name, the distance can be used as a confidence of synonymy. Terms which directly map to single identifier from a single database

are considered more "robust" compared to terms which are indirectly linked by 3 or more alternative identifiers.

These capabilities can be achieved by running the below queries:

Get synonyms for a term (e.g. “ciprofloxacin”):

```
MATCH (n:_NAME_ {name: "ciprofloxacin"})-[:ID|SYNONYM|”BRAND NAME”|”IUPAC NAME”|NAME*1..3]-(syns:_NAME_) RETURN DISTINCT(syns)
```

Get synonyms excluding a source (e.g. DrugBank):

```
MATCH paths=(:_NAME_ {name:"ciprofloxacin"})-[:ID|SYNONYM|”BRAND NAME”|
“IUPAC NAME”|NAME*1..3]-(syns:_NAME_ {type:"name"})
WHERE ALL(path in relationships(paths) WHERE path.source <> "drugbank")
RETURN DISTINCT(syns)
```

Get all database identifiers for a term (e.g. “ciprofloxacin”):

```
MATCH (:_NAME_:drug {name:"ciprofloxacin"})-[:ID|SYNONYM|”BRAND NAME”|
“IUPAC NAME”|NAME*1..3]-(syn:_ID_)
RETURN DISTINCT(syn)
```

Get all direct children for a parent (e.g. the genus "Klebsiella"):

```
MATCH (:species:_NAME_ {name:"klebsiella"})-[:ID]->(parentID:_ID_)<-[:IS_A]-(child:species:_ID_)<-[:ID]-(child_name:species:_NAME_) RETURN child_name
```

Get direct parent(s) for a term/id (e.g. "HMDB00001"):

```
MATCH (n:_ID_ {id: "hmdb:hmdb00001"})-[:IS_A]-(parents) RETURN parents
```

As mentioned above, this graph structure makes it easier to cross-map between different sources, while maintaining the original mapping from each source. This maximizes the recall of synonyms for a given term. For example: When querying 1-methylhistidine, we retrieve identifiers from YMDB, HMDB, ECMDB and CHEBI. From CHEBI, we find the IUPAC name which is 2-amino-3-(1-methyl-1h-imidazol-4-yl)propanoic acid, and we have the synonym 1-methyl-dl-histidine. Doing the same for YMDB, we get synonyms such as l-3-methylhistidine and 3-methyl-l-histidine which are not present in the other databases.

Similarly, from ECMDB we get (2s)-2-amino-3-(1-methyl-1h-imidazol-4-yl)propanoic acid which according to PubChem (Pubchem, n.d.) is the same compound. Furthermore, the non-tree nature of the graph structure can be seen to be advantageous when looking at an example from the Catalogue of Life graph (Figure 35). In this example, Klebsiella genus has 5 species (K. pneumoniae, K. singaporensis, K. oxytoca, K. granulomatic, and K. variicola). Each species has multiple variants of the name which are directly attached to the species ID by the :ID relationship. However, K. granulomatis has a sister synonym with ID 16961901 which in turn has other variants. This structure enables filtering these variants should these be noisy mappings, without having to re-run named entity recognition and pre-processing the documents.

6.5.2 Dictionary visualization

A number of network visualization libraries were researched to visualized the constructed networks, including: D3 (Bostock et al., 2011), Cytoscape (Shannon et al., 2003) and Gephi (Bastian et al., 2009). D3 provides excellent customizability and interactivity, however, as the data is loaded in the Document Object Model (DOM) of the web browser, it is too slow for such big data. Although Cytoscape is specified JSON support (Table 30), it was unable to do so successfully, and while the network was converted to the supported csv format as edge lists, it appeared to be unable to handle such big data. GraphML is one of the formats that retains most network features and is widely supported (Table 30). Therefore, network was modified to export the network to such format. HMDB was successfully loaded and visualized in Gephi (Figure 36). A mismatch between the identifiers in the classified ontologies and the metabolites from HMDB was identified for 600 elements. This was also the case for 410 identifiers in T3DB, and 260 identifiers in DrugBank. This was communicated with the authors.

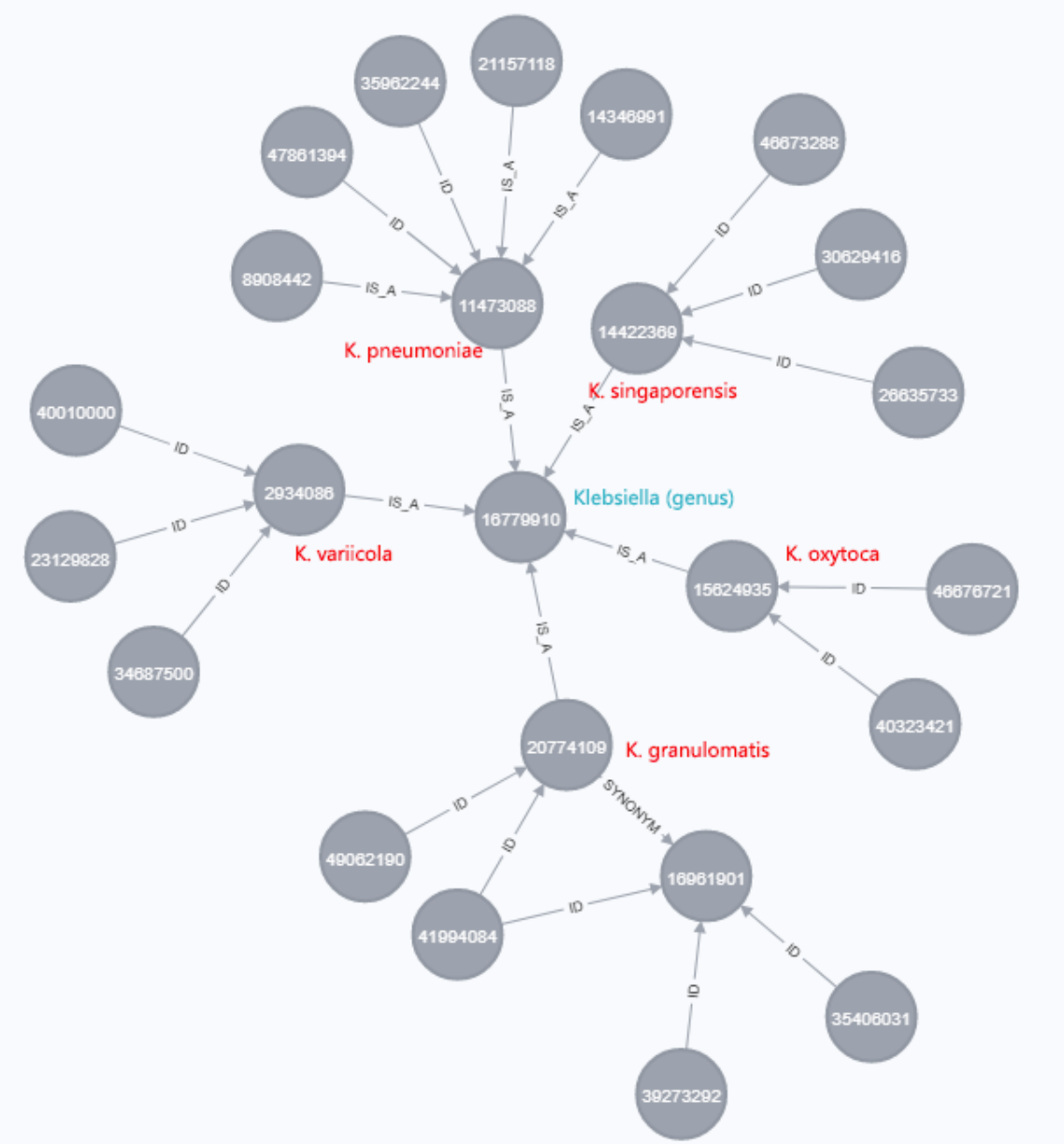


Figure 35. Part of the Catalogue of Life dictionary graph showing 5 species (*K. pneumoniae*, *K. singaporensis*, *K. oxytoca*, *K. granulomatic*, and *K. variicola*) for the Klebsiella genus. Each species has multiple variants of the name which are directly attached to the species ID by :ID relationship. However, *K. granulomatis* has a sister ID 16961901 as synonym, which in turn has other variants. Given the graph structure, we can retrieve these identifiers which are not directly linked to 20774109. This allows to maintain the original IDs in the dictionary i.e. 16961901. We also note that 11473088 (*K. pneumoniae*) also has further sub-children which are sub-species. These should also be included when querying "Klebsiella" as a genus therefore we add infinite depth to the IS_A relationship.

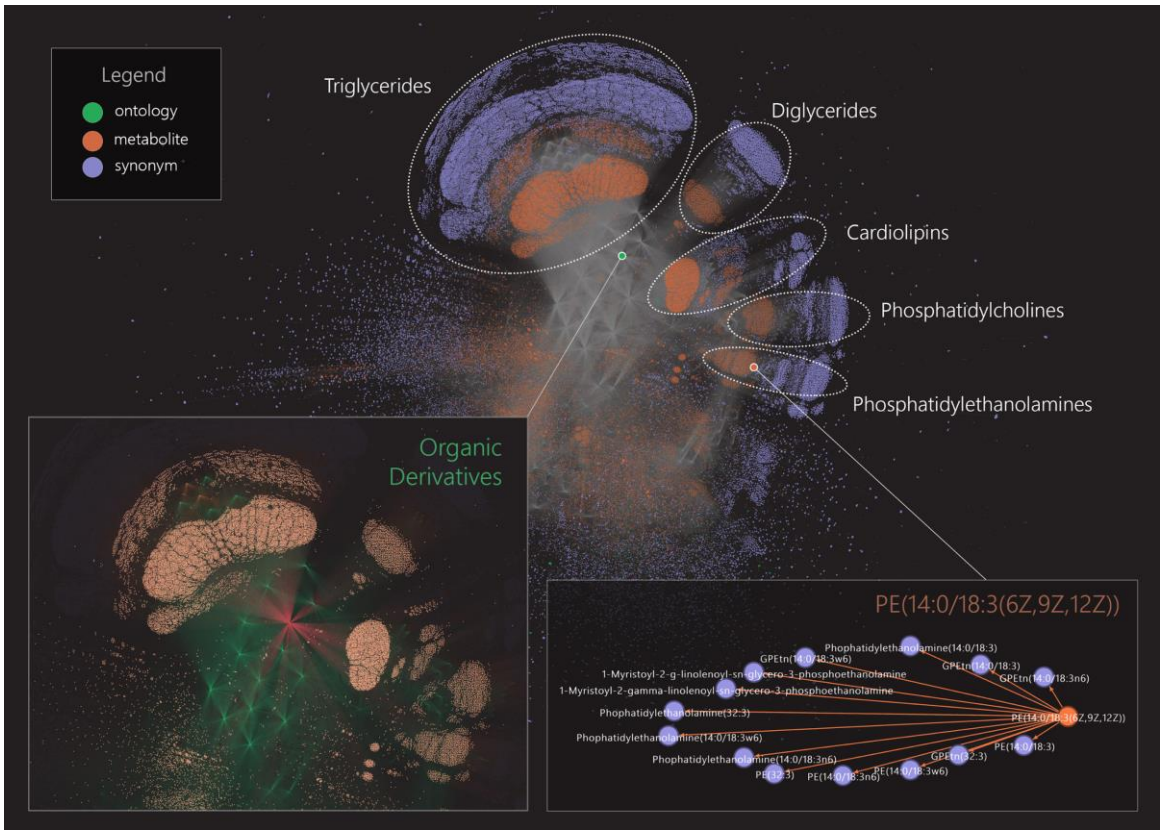


Figure 36. Network of human metabolites constructed from the human metabolite database (HMDB) classified into chemical ontologies and linked to their corresponding synonyms. Triglycerides, diglycerides, cardiolipins, phosphatidylcholines and phosphatidylethanolamines are encircled, and the metabolites in the ontology 'Organic Derivatives' are highlighted. Synonyms are shown for 'phosphatidylethanolamine PE(14:0/18:3(6Z, 9Z, 12Z)). Figure published in (Galea, Laponogov, & Veselkov, 2018a).

Table 30. Data formats supported by a number of network visualization packages, the respective programming languages developed in, and usage license.

Software	Edge	GEXF	GML	Pickle	GraphML	JSON	LEDA	YAML	Sparse Graph6	Pajek	GIS	GDF	Pajet	UCINET	Tulip	Netdraw	PSI-MI	BioPAX	SBML	XGMML	NNF	SIF	GT	DOT	VisML	DIMACS	LEDA	Language(s)	License
Networkx	x	x	x	x	x	x	x	x	x	x	x																	Python	BSD
Graph-tool			x		x																		x	x				Python, C++	GPL
Cytoscape(JS)	x		x		x	x											x	x	x	x	x	x						Java, Javascript	LGPL
Gephi	x	x	x		x							x	x	x	x	x								x				Java, OpenGL	CDDL, GPL2
GraphViz																								x				C	EPL
iGraph	x		x		x					x																x	x	C	GNU, GPL2
VisANT	x		x														x	x	x						x			Java	not declared
D3						x																						JavaScript	BSD3
SigmaJS		x				x																						JavaScript	MIT

6.5.3 Power analyses: Model training strategy

75 biomedically-related corpora were collated from primary and/or secondary sources and compiled along with respective statistics and additional information such as tagged entity classes, annotation formats, year of publication, and number of documents (Table 31). Due to licensing, retrieval links are provided, and as the hosts may change over time and new corpora published, Table 31 is made available to the community on GitHub: https://github.com/dterg/biomedical_corpora . Annotation formats varied widely from stand-off (.ann), IOB, BioC (Comeau et al., 2013) or otherwise. Where the same corpus was available in multiple formats, these were all compiled for cross-reference.

During the pre-processing of the corpora, each corpus was duplicated to annotate one entity class only, generating as many versions of the corpus as it contains entity classes. This approach was taken as opposed to a multi-class sequence labeling problem for a number of reasons, published in (Galea et al., 2018b):

1. **Scalability**: with the availability of new corpora annotating new entity classes, recognizing the new entity class would require retraining the whole model in the case of an existing multi-class model. However, with multiple single class models for each class, recognizing a new entity class would only require to train a new model for the new class and integrating with the existing models.
2. **Multiple acyclic inheritance**: An entity is not exclusive to one class and may thus belong to multiple classes. Classification and prediction in a multi-class model would not be straightforward with the current implementation. For example, on a single level, proteins are (a subset of) chemicals but not all chemicals are proteins, thus a protein

entity belongs to both the class proteins as well as chemicals if these are considered separate.

3. **Corpora available**: training corpora available are highly varied; from specific corpora such as DDI (drug-drug interaction) corpus to broader chemical classes such as CHEMDNER which annotate chemicals including drugs and proteins into a single class.
4. **Frontend**: with the ultimate aim of providing a realistic evaluation and training of machine learning-based NER models for deployment in a scalable end-to-end tool, applications were considered. With single class classification models, an entity may have multiple annotations. This is favorable over a single annotation as if an entity such as interleukin is listed and classified as only a chemical, a user will not recall it if querying proteins. Having it labeled as both a chemical and protein will allow for such entity to be recalled in both instances.

Table 31. List of compiled biomedically-related corpora, corresponding year of publication, different formats of availability and a brief description of the data, if available. Where corpora are available from multiple sources, size may differ and each document may be defined differently in different corpora (e.g. title, whole manuscript document, abstract). Originally published in (Galea et al., 2018b). For compactness, sources have been excluded from this version. A more complete and updated version is also available on GitHub: https://github.com/dterg/biomedical_corpora, and in Supplementary Table 1.

Corpus	Year	Format	Documents
Ab3P (Abbreviation Plus P-Precision)	2008	BioC	1250 PubMed Abstracts
AIMed	2005	BioC	~ 1000 MEDLINE abstracts (200 abstracts)
AnatEM (Anatomical entity mention recognition)	2013	CONLL, standoff	1212 docs (500 docs from AnEM + 262 from MLEE + 450 others)
AnEM	2012	BioC	500 docs (PubMed and PMC); abstracts and full text drawn randomly
AZDC (Arizona Disease Corpus)	2009	IeXML, .txt	2856 PubMed abstracts (2775 sentences). Other source says 794 PubMed Abstracts
BEL (BioCreative V5 BEL Track)	2016	BioC	
BioADI	2009	BioC	1201 PubMed abstracts
BioCause	2013	standoff	19 full-text documents
BioCreative-PPI		XML	
BioGRID	2017	BioC	120 full text articles
BioInfer	2007	BioC	1100 sentences from biomedical literature
BioMedLat	2016	standoff	643 BioASQ questions/factoids
BioText	2004	txt	100 titles and 40 abstracts

CDR (BioCreative V)		BioC	
CellFinder 1.0	2012	BioC	10 full documents from PMC from (Loser et al. 2009) on "Human Embryonic Stem Cell Lines and Their Use in International Research"
CG Cancer-Genetics (BioNLP-ST 2013)	2013	BioC, standoff	
CHEMDNER (BioCreative IV Track 2)	2013	BioC / standoff	
Chemical Patent Corpus	2014	standoff	200 patents
CoMAGC	2013	XML	821 sentences on prostate, breast and ovarian cancer
CRAFT	2012		97 full OA biomedical articles
Craven (Wisconsin corpus)	1999	other	1,529,731 sentences (automated)
CTD (BioCreative IV Track 3)		BioC	
DDICorpus	2011 2013	BioC	792 texts from DrugBank and 233 Medline abstracts
DIP-PPI (Database of Interaction Proteins)		other	Only proteins from yeast.
EBI:diseases	2008	other	856 sentences from 624 abstracts
eFIP	2012 2015	xlsx	
EMU (Extractor of Mutations)	2011	other	
EU-ADR	2012	other	300 PubMed abstracts (drug-disoder, drug-target, gene-disorder, SNP-disorder)
Exhaustive PTM (BioNLP 2011)			
FlySlip	2007	CONLL	82 abstracts, 5 full papers
FSU-PRGE	2010	leXML	3236 MEDLINE abstracts (35,519 sentences)
GAD	2015	csv	
GeneReg	2010	BioC	314 Abstracts
GeneTag (BioCreative II Gene Mention)	2005	BioC	20,000 sentences MEDLINE
GENIA (BioNLP Shared Task 2009)			
GENIA (BioNLP Shared Task 2011) GE, EPI, ID, REL, REN, CO, BB, BI		BioC, standoff	
GENIA (term annotation)	2003	BioC, XML	
GETM	2010	BioC, standoff	
GREC (Gene Regulation Event Corpus)	2009	BioC, standoff, XML	240 MEDLINE (167 on E.coli and 73 on Human)
HIMERA	2016	standoff	
HPRD50 (Human Protein Reference Database)	2004	BioC	50 abstracts
IEPA	2002	BioC	slightly over 300 MEDLINE abstracts
iHOP	2004	other	~ 160 sentences
iProLINK / RLIMS	2004	other, XML, BioC	
iSimp	2014	BioC	130 MEDLINE abstracts (1199 sentences)
Linnaeus	2010	standoff	
LLL (Learning Language in Logic)	2005	BioC	
MEDSTRACT		BioC	199 PubMed citations
MedTag	2005	other	
Metabolite and Enzyme	2011	BioC, XML	296 abstracts
miRTex	2015	BioC, standoff	350 abstracts (200 development, 150 test)
MLEE	2012	CONLL, standoff	262 PubMed abstracts on molecular mechanisms of cancer (specifically relating to angiogenesis)
mTOR pathway event corpus (BioNLP 2011)	2011	standoff	

MutationFinder	2007	other	305 abstract (development data set), 508 abstract test set
Nagel		XML, standoff	
NCBI Disease	2012	other	6881 sentences in 793 PubMed abstracts
OMM (Open Mutation Miner)	2012	other	40 full texts
OSIRIS	2008	BioC, XML, standoff	105 articles
PC (Pathway Curation) (BioNLP-ST 2013)	2013	BioC	
PennBioIE-oncology	2004	leXML	1414 PubMed abstracts on cancer
pGenN (Plant-GN)	2015	BioC	104 MEDLINE abstracts
PICAD	2011	XML	1037 sentences from PubMed
PolySearch (includes v1. and v2.)		other	
ProteinResidue		other	
SCAI_Klinger	2008	CONLL	
SCAI_Kolarik	2008	CONLL	
SETH	2016	standoff	630 publications from The American Journal of Human Genetics and Human Mutation
SH (Schwartz and Hearst)	2003	BioC	1000 PubMed Abstracts
SNPCorpus	2011	BioC	296 MEDLINE abstracts
Species	2013	standoff	800 PubMed abstracts
T4SS (Type 4 Secretion System)	2011	CONLL	
T4SS Event Extraction (BioNLP 2010)	2010	other	
tmVar	2013	BioC	500 PubMed abstracts
VariomeCorpus (hvp)	2013	BioC	
Yapex	2002	other	99 training, 101 test MEDLINE abstracts

Table 32. Basic statistics on the number of entities and unique entities in each corpus, the original entity classes and the new entity class to which they were remapped in this study. As published in (Galea et al., 2018b).

Corpus	Entity Class (original)	Entity Class (Remapped)	nEntities	nUniqueEntities
AIMED	protein	GeneProtein	4236	1138
BioGrid	Gene	GeneProtein	6489	1068
CellFinder	GeneProtein	GeneProtein	1750	734
VariomeCorpus	gene	GeneProtein	4613	453
IEPA	Protein	GeneProtein	1117	130
miRTex development	Gene	GeneProtein	1266	484
	Complex	GeneProtein	24	7
	Family	GeneProtein	57	28
miRTex test	Gene	GeneProtein	922	368
	Complex	GeneProtein	32	9
	Family	GeneProtein	78	31
mTor	Tag	GeneProtein	3	2
	Receptor	GeneProtein	1	1
	Protein	GeneProtein	1483	297
	Complex	GeneProtein	201	69
OSIRIS	gene	GeneProtein	799	260
SETH	Gene	GeneProtein	2315	969
VariomeCorpus	mutation	GeneProtein	1690	429
OSIRIS	variant	GeneProtein	551	369
SETH	SNP	GeneProtein	895	689
	RS	GeneProtein	9	3
SNPCorpus	NSM	GeneProtein	244	230
	PSM	GeneProtein	278	216
tmVar test	SNP	GeneProtein	39	29
	ProteinMutation	GeneProtein	205	137
	DNAMutation	GeneProtein	220	156
tmVar train	SNP	GeneProtein	96	58
	ProteinMutation	GeneProtein	440	254

	DNAMutation	GeneProtein	431	305
CHEMDNER - Development	MULTIPLE	ChemicalDrug	188	175
	NO CLASS	ChemicalDrug	32	15
	FAMILY	ChemicalDrug	4223	1573
	ABBREVIATION	ChemicalDrug	4521	812
	SYSTEMATIC	ChemicalDrug	6813	2756
	FORMULA	ChemicalDrug	4137	839
	IDENTIFIER	ChemicalDrug	639	240
	TRIVIAL	ChemicalDrug	8970	2268
CHEMDNER - Training	MULTIPLE	ChemicalDrug	202	177
	NO CLASS	ChemicalDrug	40	13
	FAMILY	ChemicalDrug	4086	1444
	ABBREVIATION	ChemicalDrug	4536	822
	SYSTEMATIC	ChemicalDrug	6655	2820
	FORMULA	ChemicalDrug	4448	840
	IDENTIFIER	ChemicalDrug	672	231
	TRIVIAL	ChemicalDrug	8823	2172
DDI - negative	DrugName	ChemicalDrug	826	416
DDI - positive	DrugName	ChemicalDrug	240	176
VariomeCorpus	Chemicals_Drugs	ChemicalDrug	1	1
Metabolites	Entity	ChemicalDrug	2454	653
mTOR	Ion	ChemicalDrug	5	2
	Simple_molecule	ChemicalDrug	26	13
	Drug	ChemicalDrug	42	3
miRTex - development	MiRNA	RNA	1539	469
miRTex - test	MiRNA	RNA	1217	353
mTOR	RNA	RNA	12	7

6.5.4 Power analyses: Identifying genes and proteins

Proteins, genes and variants are ontologically related and therefore were initially grouped into a single superclass to determine the overlap between annotations across different corpora. SNPcorpus and tmVar did not provide a high predictive capacity, as prediction performance was low before introducing OSIRIS/SETH training data (Figure 37). SNPcorpus and tmVar contain mutation annotations, whereas no other training corpus introduced prior to OSIRIS/SETH contained such entity types, accounting for the poor predictive performance. This concerms that mutation entities are significantly different from the gene/protein classes and were therefore considered as a separate class in subsequent tasks (Galea et al., 2018b). Contrastingly, VariomeCorpus contains variants (1690 entities) annotations as well as genes (4613 entities). This accounts for the better prediction of its test split compared to mutation-specific corpora such as SNPcorpus and tmVar. These differences are visible in the significant

univariate features shared between genes and proteins from different corpora, whereas the variants class is much less consistent throughout (Figure 42) (Galea et al., 2018b).

Excluding the variants annotations, we next determined the generalizability of the models for the gene/protein class by applying models trained on individual corpora to each other corpus (not seen in the training - leave-corpus-out cross-validation) (Figure 38). Increasing quantity of training samples improved performance in all corpora, however the best predictions were achieved when the test samples originated from the same corpus, whereas the predictive capacity varied widely for other corpora (Figure 38 A-H). IEPA was the hardest to predict (Figure 38 A-C; E-H). Inversely, IEPA-trained models performed poorly on other corpora's test splits (Figure 38D). This suggests that the data is incompatible and there is high corpus bias. Looking further into the orthographic features, IEPA can be seen to be the most inconsistent when compared to other GeneProtein corpora (Figure 42). This gives the IEPA corpus a very different feature 'fingerprint' compared to other corpora within the same class (Galea et al., 2018b).

In the second approach, all the corpora for training and testing were merged. This resulted in an increase in the overall performance and consistency (Figure 40). Generally, there was a dependence between sample size and performance, therefore increasing quantity of training data increased performance. Good generalizability was achieved for some corpora, where corpora left out from training were predicted by other corpora with similar performance to their own (Galea et al., 2018b). SETH is one such example, where an F-score of 64% was achieved using all corpora for training, and a similar 63% was achieved when leaving SETH out of training. Similarly for miRTex, all training data and training without MiRTex samples

converged at an F-score of 76% at 450 documents. However, in the latter case, additional documents improved the performance up to 84% (Galea et al., 2018b).

In general, maximum absolute performance for a given test set was achieved when training data included the training set of the same corpus. Nonetheless, in the majority of cases, a relative performance plateau was reached with 1000 training samples, achieving a maximum weighted average of 78.32% F-score (Figure 39).

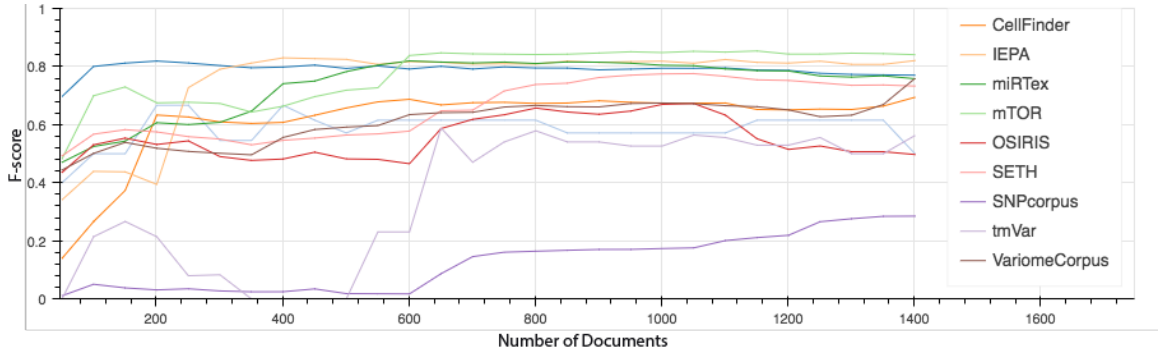


Figure 37. Raw learning curve plots when genes, proteins and variants were considered as a single "GeneProteinVariants" superclass. Different corpora may differ in the annotation standards for the same entities, resulting in poor/no predictive performance on other corpora. However, overall performance generally does not decrease substantially with the introduction of new data from other sources. Figure published in (Galea et al., 2018b).

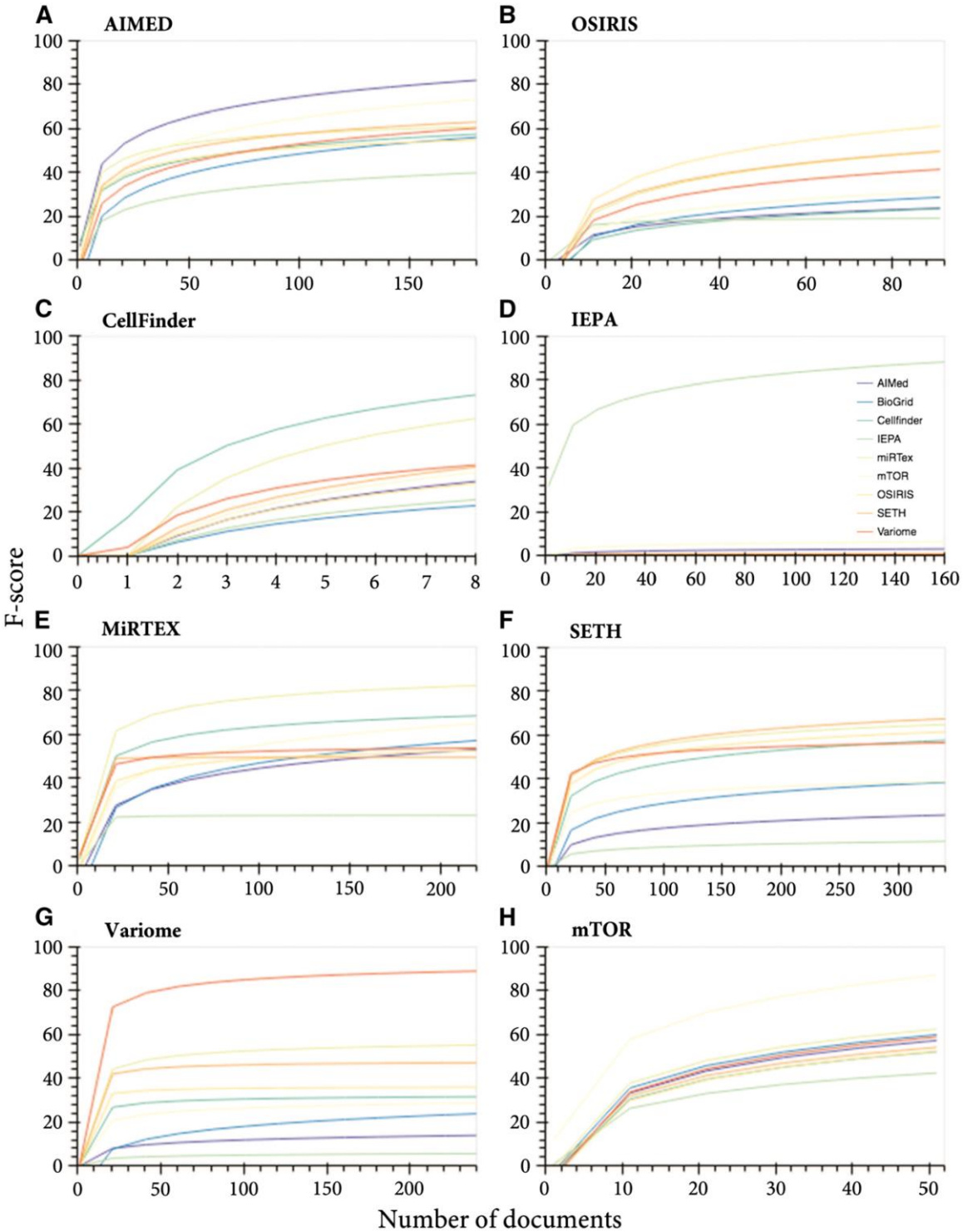


Figure 38. 'GeneProtein' class learning curves obtained by each corpus. Learning curves for models trained on each corpus and applied to test data from all corpora to test the generalizability of each corpus. (A) AIMed; (B) OSIRIS; (C) CellFinder; (D) IEPA; (E) miRTeX; (F) SETH; (G) VariomeCorpus; (H) mTOR. Figure published in (Galea et al., 2018b).

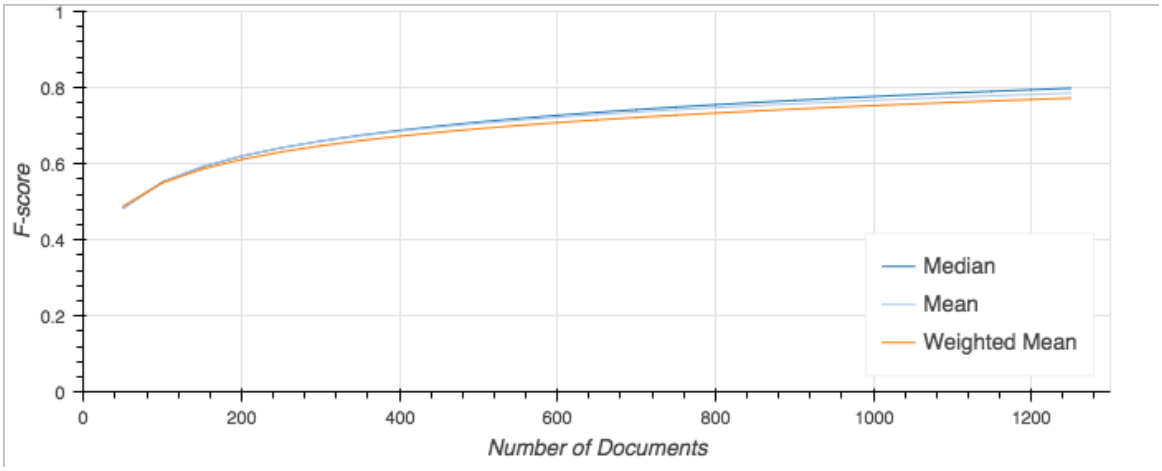


Figure 39. Average accuracy measured by F-score for the "GeneProtein" class when entities are predicted with the model trained on all merged data. Corpora annotating genes and/or proteins were merged and split for training and testing. Mean, median and weighted mean F-score obtained when applying the trained model to the test data is shown, where performance appears dependent on the training size up to 1200 documents. Figure published in (Galea et al., 2018b).

6.5.5 Power analyses: Identifying variants

As a result of the highly variant orthographic features and the poor predictive power observed, variants were considered as a separate class. When performing leave-corpus-out cross-validation, where test data of a given corpus is predicted by other corpora except that corpus, VariomeCorpus achieved poor predictive performance by any other corpus. Looking at the raw data, most entities are observed to be genes followed by the term "mutant" rather than mutation terms which followed the standard nomenclature (Galea et al., 2018b). With entities following the structure 'gene X mutant', this may account for the prediction of the VariomeCorpus test data as genes when genes, proteins and variants were collectively considered as a superclass (Figure 37). Such difference is also indicated in the orthographic feature map (Figure 42) where OneDigit, OneCap, threeCap and Length3-5 were identified as significant features for VariomeCorpus but not in any other mutation corpus. On the otherhand, whereas the '+' character was identified as significant in all other variant corpora, VariomeCorpus did not share this property. Recently, (Cejuela et al., 2017) reported VariomeCorpus annotates vague mentions such as 'de novo mutations' and 'large deletion', with only a subset mentioning

position-specific variants (Galea et al., 2018b). Due to this, VariomeCorpus was excluded from subsequent power analyses. If more training data is identified to be required to improve performance, the subset mentioning position-specific variants can be extracted as described by (A. Jimeno Yepes & Verspoor, 2014).

Merging SETH, SNPcorpus, tmVar, and OSIRIS achieved 82% for SETH and tmVar test data at about 500 documents. Performing leave-corpus-out cross-validation identified tmVar to give generalizable performance as its test data was predicted with equal performance. This was not the case for other corpora, where unseen corpora test data was poorly predicted when using merged training data (Figure 38B). This indicates that a subset of the entities are specific to these corpora (Galea et al., 2018b). OSIRIS obtained the lowest plateaued performance. Inspecting its annotations, indeed they appear to contain non-standard nomenclature, with annotations such as: codon 72 (CCC/proline to CGC/arginine, (TCT TCC) in codon 10, -22 and -348 relative to the BAT1 transcription start site, A at positive -838, C in -838 (Galea et al., 2018b). Such mutation-related entities however are still valid and would thus require more training data similar to OSIRIS, although standardization of nomenclature in more recent publications may render this unnecessary (Galea et al., 2018b).
Omitting OSIRIS from the analyses and plotting learning curves for SETH, SNPcorpus and tmVar, we observe that tmVar and SETH obtained lower performance. This backs up the positive contribution of OSIRIS training data to the predictive performance of tmVar and SETH (Galea et al., 2018b).

Plotting the learning curves per corpus (Figure 41), we observe that tmVar training split (Figure 41C) only predicts tmVar's test split with 50% accuracy. The learning curve trend indicates that increasing the size of the training data is expected to improve performance further. Indeed,

performance increase up to >80% when the other corpora were added to the training dataset (Figure 40B). High model generalizability can be inferred from the similar performance obtained when tmVar was out of the training split (Figure 40B) (Galea et al., 2018b). SETH corpus achieved 86.02% accuracy both when using only its training data as well as when merging the training data (Figure 40B; Figure 41B). Removing the SETH from training achieves a lower 67.09%, indicating a contribution of 18.93% by SETH data. In terms of generalizability, tmVar test data was predicted at 66.67% by SETH data (Figure 41B) (Galea et al., 2018b).

As for SNPcorpus and OSIRIS, performance trends were similar when merging corpora (Figure 40B) and when corpus-specific training data was used (Figure 41A, D). In the merged training data scenario, the absolute performance was slightly lower, indicating potential noise introduction to the model (Galea et al., 2018b). The inconsistent performance and limited model generalizability for identifying variants can be seen in the orthographic feature analysis, where significant features vary by source/corpus. Most consistent feature includes three or more digits, however there is no overall distinctly evident 'fingerprint' of univariate significant features across the different corpora (Galea et al., 2018b).

A more detailed exploration of such variation has been reported in literature (Cejuela et al., 2017), where mutation mentions were classified as 'standard', 'semi-standard', and 'natural language', with SETH and tmVar sharing a subset of standard mutations while only SETH captured natural language mentions (Cejuela et al., 2017; Galea et al., 2018b).

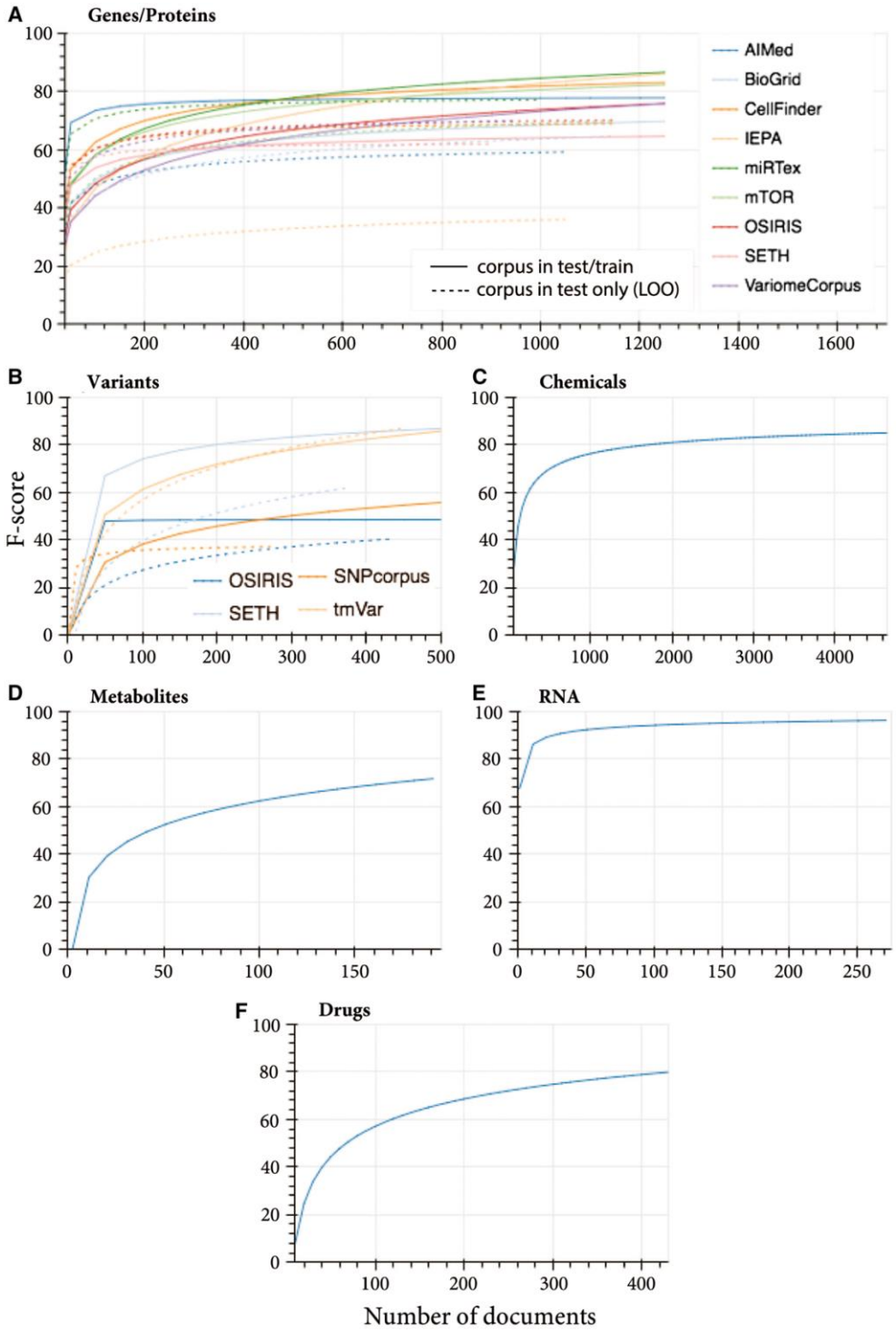


Figure 40. Learning curves obtained when: (i) multiple sources are used as training data to predict test data from each other corpus, individually; and (ii) each corpus is excluded from training and its test data is predicted by the other corpora (leave-corpus-out cross-validation). Each subplot represents training and testing of the different entity classes: (A) Genes and proteins (dashed lines represent leave-corpus-out validation approach); (B) variants; (C) chemicals; (D) metabolites; (E) RNA; and (F) drugs. Figure published in (Galea et al., 2018b).

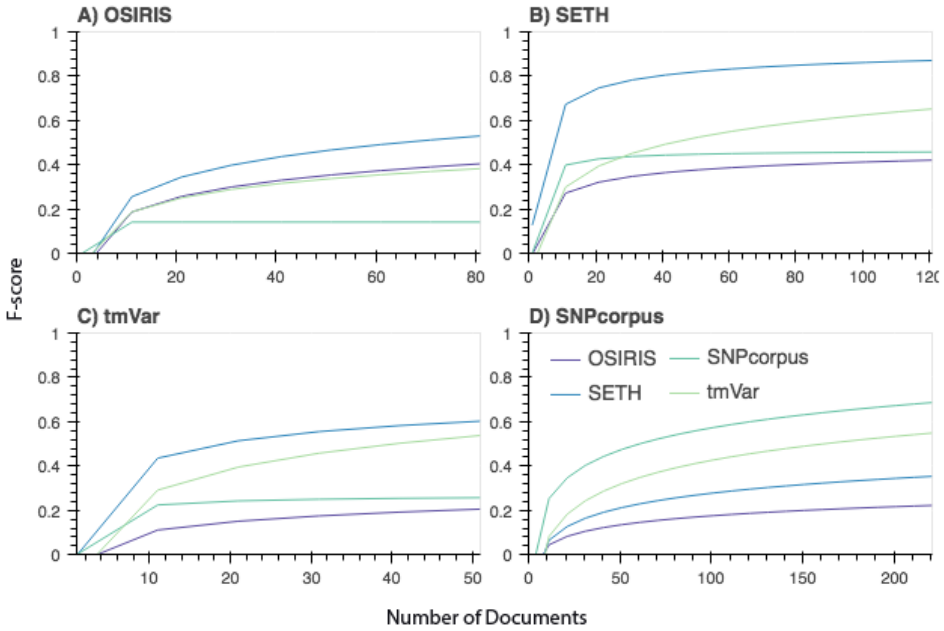


Figure 41. "Variants" class learning curves obtained by each corpus. Learning curves for models trained on each corpus and applied to test data from all corpora to test the generalizability of each corpus. (A) OSIRIS; (B) SETH; (C) tmVar; and (D) SNPcorpus. Figure published in (Galea et al., 2018b).

6.5.6 Power analyses: Identifying chemicals, drugs and metabolites

Drugs and metabolites are a subset of the chemical ontology, therefore these three classes were remapped into a single 'ChemicalDrug' superclass. Corpora like CHEMDNER annotate genes as chemicals whereas more specific corpora such as DDI and metabolites corpus only annotate the specific entity classes: drugs and metabolites, respectively. Therefore, this predicts poor predictive capacity from these corpora for the genes in the CHEMDNER test set. Given such mismatches, we devised new classes (Galea et al., 2018b).

CHEMDNER is the biggest corpus with over 58,000 chemical entities annotating formulae and multiple alternative names such as: systematic names and chemical families. As such comprehensive annotations are only exclusive to this corpus, this corpus was considered on its own (Galea et al., 2018). From an applied perspective, as genes/proteins such as 'Interleukin-2' are annotated in the CHEMDNER corpus, this entity would be annotated multiple times

when the GeneProtein model and CHEMDNER model are used to predict such token (Galea et al., 2018b). As such entity is a child of the GeneProtein ontology as well as the Chemical ontology, this is an appropriate approach. Furthermore, this further justifies the single entity class models approach used in this study versus a multi-class classification approach (Galea et al., 2018b). A plateaued performance was achieved after 1200, 160 and 400 documents for chemicals, metabolites and drugs, respectively. Overall, an F-score of 84.8% was achieved by CHEMDNER with a plateau reached around 1500 documents (Figure 40C) (Galea et al., 2018b). Similar performance was reported previously (Krallinger, Leitner, et al., 2015).

As for drug named entity recognition, while both DDIcorpus and mTor annotate drug entities, mTor only annotates 3 unique drug entities (Table 32) and therefore was excluded. Plotting a learning curve for the DDIcorpus achieved an average performance of 78.48% at 380 training documents (Figure 40F) (Galea et al., 2018b).

Finally, while metabolites are annotated in CHEMDNER and the metabolites corpus, such entities are labeled broadly as chemicals in the former and hence cannot be distinguished from non-metabolite entities, including drugs and proteins. As a result, the learning curve for metabolites was based on the metabolites corpus (Figure 40D), achieving a stable 71.98% F-score at 160 training documents (Galea et al., 2018b).

6.5.7 Power analyses: Identifying RNA

The corpora miRTex and mTor annotate RNA entities (Table 32), with the former containing over 2700 entities whereas the latter only annotates 7 unique entities. As the quantity of data in mTor is insufficient for representative power calculations and, this was excluded (Galea et al., 2018b). 91% F-score was achieved by miRTex with only 21 documents (Figure 40),

increasing marginally to a maximum 96.17% when using all the training data. Such high performance and stability is likely due to the highly consistent RNA nomenclature (Galea et al., 2018b).

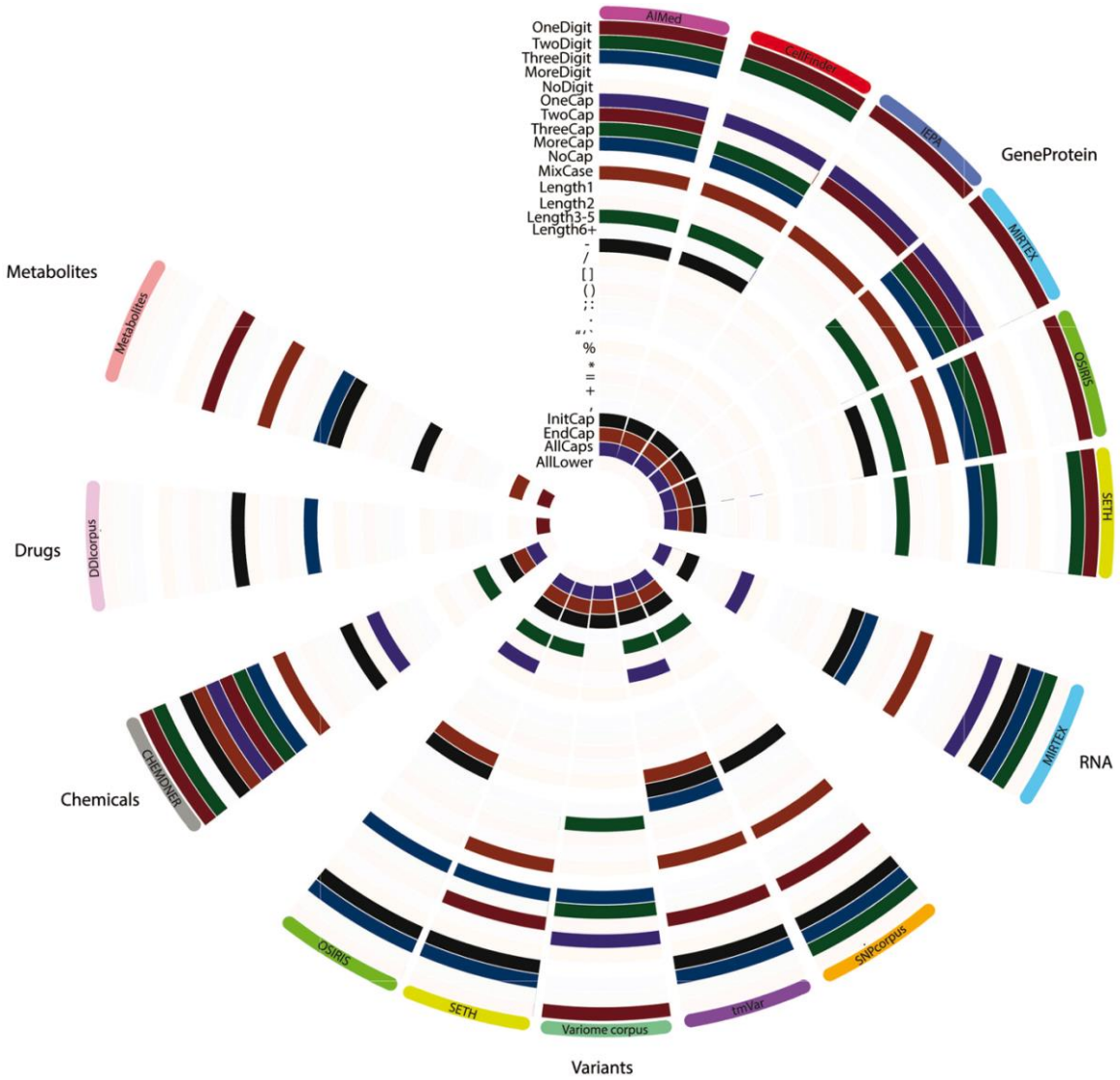


Figure 42. Orthographic features identified as significant to the entity classes: Gene-Protein, RNA, variants, chemicals, drugs and metabolites. Highlighted features were identified to be univariately significant in the orthographic feature analysis for a given entity class in a given corpus. Rows represent such orthographic features while sectors/columns represent corpora; grouped by the entity classes. Figure published in (Galea et al., 2018b).

6.6 Conclusion(s) and Future Direction(s)

Through this work, 2 dictionary-based string-matching implementations are provided as part of the HASKEE pipeline as well as a standalone queryable graph-based dictionary that serves as an inter-map between multiple sources and maximizes information retention such as hypo- and hypernymy. This is likely to maximize recall for nested entities and fine-grained control on synonyms. Nonetheless, future work could quantitatively evaluate and compare the performance of such approach, requiring developing appropriate benchmark datasets that are currently not available.

Beyond string-matching, we showed that machine learning-based models suffer from generalizability issues, proving that offline metrics are not reflective of the PubMed-scale performance as would be observed when integrated with HASKEE. Training models on individual sources achieves lower performance when applied to different datasets, likely due to annotation standard differences and corpus over-fitting. Merging training data provides a more generalizable model, although performance is slightly lower than models trained on single sources. Power analyses showed that performance is generally not limited by the quantity of training data. However, improving the annotation standards across multiple sources may improve generalizability and overall performance – providing realistic translational performance for named entity recognition. As usage and popularity of neural models has increased, outperforming absolute performance of traditional methods, useful future direction could involve carrying out similar generalizability and power analyses done in this work.

Beyond quantitative evaluation, from a translation perspective, dictionary-based and machine learning-based approaches can be compared and/or merged. The modular design of HASKEE enables the integration of any approach, irrespective of whether a neural or traditional model

is used. This is critical as providing and integrating a fixed ‘single optimal solution’ is unfeasible and unrealistic given the fast speed of development in the machine learning field.

Chapter 7 – Biomedical associations:
Extraction, database, pipeline and search engine

7.1 Abstract

Relation extraction is an important task in biomedical natural language processing, involving the identification of associations between biomedical entities. Latest research utilizing supervised machine learning approaches achieves state-of-the-art performance on benchmark datasets for entity pairs such as gene-disease and drug-drug associations. However, training data is limited when performing relation extraction at a large scale, with a large number of different entity classes and existing models do not generalize well as they tend to overfit the training data. In this work we avoid training specific entity pair models and consider co-occurrence as a simplistic approach for candidate pairs, determined by a set of filters. To further filter false positive associations, the application of a generic sentence claim identification classifier is tested. This approach is not domain-specific and therefore more generalizable. Quantitative evaluation shows competitive results for chemical-disease associations and similar performance differences to adversarial training. Qualitative evaluation shows improved results when compared to simple PubMed searches.

Parsing, pre-processing and named entity recognition are pre-requisite steps to relation extraction; for which a number of different approaches are available. Here we present HASKEE: a modular, highly customizable and documented pipeline. Applying this pipeline to PubMed, we extract biomedical associations and compile them in a devised scalable graph model that enables easy exploration through graph traversal while allowing to extract all evidence from source. Finally, we show a proof of concept/prototype for a complementary search engine frontend. The whole pipeline will be available on https://www.github.com/dterg.

7.2 Aims and objectives

Aim(s):

- To extract potential biomedical associations and compile them in a queryable database.

Objective(s):

- Devise an association extraction approach for candidate biomedical associations between several bioentity types;
- Develop a graph model for representing biomedical associations;
- Compile extracted associations in a queryable database;
- Develop a proof of concept for a user-friendly and highly customizable frontend.

7.3 Introduction

Identification and classification of relationships between entities such as proteins, drugs and diseases is critical in biomedical research as it facilitates the fundamental understanding of biological processes and may lead to the discovery of novel treatments. Efforts such as ClinVar (Landrum et al., 2014), OMIM (Hamosh, Scott, Amberger, Bocchini, & McKusick, 2005), COSMIC (Forbes et al., 2017) and UniProt (Bateman et al., 2017), compile and curate medical association knowledgebases, however with the rapid growth of research, text mining techniques are required to facilitate this by automating extraction from unstructured text. Natural language processing techniques have been utilized and proven successful in work extracting and collating gene/mutation-disease associations, such as: DISGENET (Piñero et al., 2017, 2015), tmVar (Wei et al., 2018), and DISEASES (Pletscher-Frankild, Pallejà, Tsafou, Binder, & Jensen, 2015). Such projects often target a limited number of binary entity associations, such as gene-disease associations, and update the repositories annually by running the NLP pipeline internally. With a large number of biomedical entities such as: food, drugs,

chemicals, toxins, proteins, metabolites, and mutations, a customizable, modular, scalable and open-source pipeline is needed. In this project we initiate and present the work towards such an end-to-end pipeline that extracts biomedical associations.

Numerous tools such as SNPShot, PolySearch2, MEMA, and MuteXt utilized simple heuristics or a set of trigger words to extract associations. With advancements in general machine learning techniques and latest developments in neural networks, relation extraction performance has increased considerably.

Typical machine learning based approaches involve a supervised classifier that distinguishes between an association and not an association given a pair of entities. State-of-the-art methods such as (Tikk, Thomas, Palaga, Hakenberg, & Leser, 2010), (H. Zhou et al., 2016) and (Panyam, Verspoor, Cohn, & Ramamohanarao, 2018) have achieved 40.4-72.3% for protein-protein interaction corpora: AIMED (43.8%), BioInfer (40.4%), HPRD50 (69.7%), IEPA (59.6%), and LLL (72.3%), and 61.3-65.1% for chemical-disease interactions from the CID corpus (compared to the original BioCreative V shared task highest performer reporting 57.03%). While these results show improvements and are promising on paper, such approaches are highly dependent on expensive training data which is very limited, particularly in the biomedical domain. Additionally, these models are not generalizable as they tend to overfit the domain of the training data. For example, training for protein-protein associations on the PPI corpus, the model is likely to pick up trigger words such as “phosphorylates”, “binds to”, and “catalyzes”. These are specific to proteins and therefore such model does not generalize to drug-disease associations and other term pairs. Corpora available in the medical field are available for: drug-drug, gene-gene, protein-protein, and chemical-disease interactions, therefore training a model for each entity pair could be a solution. However, this is not a

feasible approach to derive associations between combinations of a large number of entity classes. In this work, we realize these limitations and devise an alternative broad and hybrid approach that enables performing relation extraction at scale for a large number of classes. Finally, after applying the devised approach, we design a graph model that enables retrieving n-ary relations without requiring training separate models. This provides the potential to explore and traverse through a network of biomedical associations rather than a static table of binary relations as commonly compiled by existing tools.

7.4 Methods

7.4.1 Relation extraction

Following named entity recognition, relation extraction was developed as a set of filters/heuristics:

- **Filter 1**: sentences which contain at least 2 entities were considered as candidate associations

- **Filter 2**: with a trie implementation, recognized entities may share terms/tokens (e.g. for the sentence: gene X was linked with colon cancer, the recognized entities may be: gene X, colon and colon cancer, with the last 2 entities overlapping the token colon). In this case, such entities these should not be considered for associations. The intersect of the entity token indices was computed for each combination (in case of more than 2 entities). Entities intersecting with each other were not considered for associations. Otherwise, all pairwise combinations were computed.

- **Filter 3**: an association cannot be between 2 entities of the same term, both within the same dictionary as well as between dictionaries. E.g. DNA from dict1 and DNA from dict2, or "DNA is […] but DNA is not […]" (where DNA would be mapped twice to the same ID from the same dictionary but has different index and is thus not captured by the intersection filter).

- **Filter 4**: each candidate sentence with a pair of entities is predicted with the novel claim detection classifier, and labelled accordingly (see Section 5.4.5).

7.4.2 Database

7.4.2.1 Database Management System and Graph Model

Initially, ElasticSearch and MongoDB were considered and tested for integration with HASKEE. Ultimately, the graph-based database neo4j was used and integrated. This choice is discussed further in Section 7.5.1.

When defining the graph structure, a number of models were considered (Figure 43) with a tradeoff between redundancy of information and faster expected response times. Going forward, an intermediary model was chosen where documents, sentences, associations and entities are represented as nodes. Sentence nodes are linked to documents by an 'IN_DOC' edge, association nodes are linked to their originating sentence by an 'IN_SENT' edge and entity nodes are assigned to an association node via a 'IN_ASSOC' edge.

To assess the scalability of neo4j and obtain indications of query times, the graph-based model was defined and a varying number of (dummy) associations with the defined model structure were used to populate the databases.

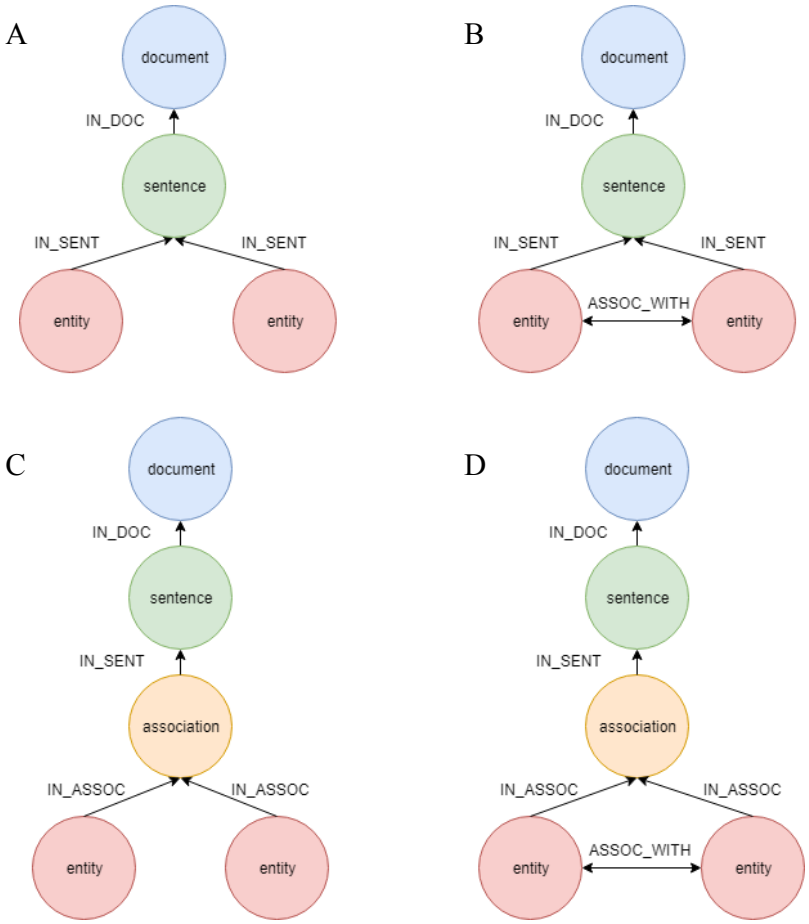


Figure 43. Graph model iterations. Different graph structures considered to model the association data. A) Basic unit of the graph-based model for the database management system. Each association is represented by a single node at the center of the graph that is linked to nodes representing the entities and the document(s) supporting this association claim. B) Alternative model structure that introduces additional edges between the entities themselves. While this information introduces redundancy to the model, it may improve traversing performance for obtaining directly and indirectly related entities. C) Graph model that represents associations with a node. This allows direct look-up of an association and allows for storing additional attributes such as scores, and type (e.g. in-silico predicted association). D) A graph model equivalent to B) with the structure of C). This introduces redundancies but may improve traversing and look-up speed.

7.4.2.2 Exporting the graph

Export of the network from python object to csv format, as compatible with neo4j, was implemented with a number of options: (i) include duplicates and export node by node (line-by-line); (ii) include duplicates, store in memory and write in chunks; and (iii) export only uniques in chunks or node-by-node. Where duplication is allowed, de-duplication can be done after exporting csv files with duplicates.

When de-deduplicating, association directionality is ignored. i.e. triplets (A, parent assoc, B) and (B, parent assoc, A) are considered identical.

The export module has a number of additional features/configurations including:

- Sentences with a single entity are still retained. Rather than IN_ASSOC, they are linked to a sentence node via a IN_SENT edge. This allows for such entities to be still be recalled. Additionally, if a user queries a free-text term, if it co-occurs in the same sentence, it is extracted as an association.
- Configure whether to keep all sentences irrespective of how many associations they have or entities they have is defined by the 'keep all sents' parameter (that filters sentences in the last module of the export). Whether to keep unpaired entities is also defined in a separate config parameter 'keep all entities'. Entities which are not paired in a sentence are linked to a SENT directly via a IN_SENT rather than an IN_ASSOC edge.

Associations are exported into 2 file types (nodes and edges) and a total of 7 files. For nodes: documents, sentences, associations and entities; and edges are saved in: IN_DOC, IN_SENT and IN_ASSOC.

Each publication/document is assigned the following attributes: year, PMID, pmc, doi, journal, title, filename, article_types, score, and DOC label. Similarly, sentences are each assigned the following attributes: id (a concatenation of pmid, section, and the sentence number), raw_txt, processed_txt, predicted_section, negation, and significance.

7.4.2.3 Populating the database

Populating the neo4j database with the associations is done through the provided neo4j command line CSV importer tool. This was chosen over using the py2neo API to execute cypher queries and populate the database in batches.

The command required to populate the database with the pipeline-generated data files is shown in Code Snippet 3. This can be executed through the neo4j desktop application.

```
SET DATA=path\to\neo4j_json\directory
.\bin\neo4j-admin import ^
--mode=csv ^
--database=medline.db ^
--ignore-duplicate-nodes true ^
--nodes %DATA%\nodes_assocs.csv ^
--nodes %DATA%\nodes_docs.csv ^
--nodes %DATA%\nodes_entities.csv ^
--nodes %DATA%\nodes_sents.csv ^
--relationships:IN_ASSOC %DATA%\edges_IN_ASSOC.csv ^
--relationships:IN_DOC %DATA%\edges_IN_DOC.csv ^
--relationships:IN_SENT %DATA%\edges_IN_SENT.csv
```

Code Snippet 3. Command to populate the neo4j database with the association graph using the built-in bulk csv importer.

To allow quick look-up, we index the entity name attribute using the command in Code Snippet 4.

```
CREATE INDEX ON :ENTITY(id)
```

Code Snippet 4. Cypher command to create an index on a labeled property.

7.4.3 Overall Pipeline

7.4.3.1 Structure and features

In HASKEE, we represent each of the tasks previously defined and described in detail as python modules, therefore creating an end-to-end python pipeline. Such pipeline has the following structure:

1. Parsing of PubMed files
2. Preprocessing: sentence tokenization, word tokenization, contraction expansion, sentence prediction, article scoring, negation cue detection
3. Named entity recognition: multiple dictionary-based approaches
4. Relation extraction
5. Graph construction

Each of these modules can be replaced entirely with custom functions or configured through the configuration file. In the latter, different pipeline parameters can be used and tested, including: whether or not to remove stopwords, expand abbreviations, expand species names, which model to use to predict sentences, the database configuration details etc. A snippet of the code is found in Code Snippet 5.

```
# default import and export paths
[import]
import_base_path = Z:/PubMed/baseline
import_path_pmc = D:/PubMed_PMC_oa_bulk_23_aug_2017/
import_path_medline = ${import:import_base_path}
import_path_api = ${import:import_base_path}api/
export_path = ${import:import_base_path}_processed/original_json/

# define settings to send notifications and/or progress to a slack channel
[notifications]
send = True
type = slack
email = dietergalea@gmail.com
legacy_slack_token = xoxp-310357240534-310357240598-323328518438-
309ae71bff014690fb721e8f534ad1c2
legacy_channel = log
posting_name = HASKEE Status

# default parameters for the preprocessing module
# Parameters: import/export paths, tokenizer, stopwords, contractions, and
# resources for sentence section prediction: embeddings, trained model and
# label encoder
[preprocessing]
import_path = ${import:export_path}
export_path = ${import:import_base_path}_processed/preproc_json/
tokenize = True
tokenizer = penntreebank
lowercase = True
remove_stop = True
stop_source = Default
expand_contractions = True
expansion_source = Default
expand_abbrev = True
```

```
expand_species_names = True
predict_sent = True
embed_path = Z:/word2vec/optimized/fasttext_optimized_extrinsic
sent_classifier = C:/Users/Dieter/Documents/GitHub/HASKEE/backend/haskee/models/sent_pred/BiLSTM-CRF_4M_optimized.h5
architecture = bilstm-crf
```

Code Snippet 5. Snippet of the HASKEE pipeline configuration file.

In addition to this, the developed HASKEE pipeline is embarrassingly parallelized using multiprocessing pathos library and uses the slack-progress library that in turn utilizes the Slack API to enable tracking the progress of each module through a Slack channel.

We have also compiled a number of useful resources with HASKEE. This includes both resources that are required as part of the pipeline by default, such as: stopword lists, synsets, and dictionaries; as well as benchmark and training data that can be used to re-train any of the machine learning models.

Technical and extensive documentation is also available for HASKEE, powered by MKDocs ("MkDocs," n.d.) and the Material for MkDocs theme ("Material for MkDocs," n.d.).

7.4.3.2 Quantitative Evaluation

In previous sections, we have quantitatively assessed the performance of different modules such as: negation detection, named entity recognition, and sentence prediction. To evaluate the ultimate results of the platform qualitatively, we perform a case study where we 'validate' the results obtained from Section 3.5.1. In Section 3.5.1, a number of drugs were identified to have the potential to be repurposed for anti-cancer therapy therefore we query the database with processed articles for associations between these drugs (and their respective synonyms and hyponyms) and cancer and manually review the returned output.

7.4.4 Proof of Concept

We develop a proof of concept for a search engine frontend based on the Flask framework. The prototype connects to the database compiled in Section 7.4.2.3, retrieves synonyms from the synonym graph for a set of entities of interest, and queries for associations between the synonym sets. For demonstration purposes, we query for the synonyms and hyponyms of "cancer" and "cetrorelix" (Section 3.5.1). The returned results are grouped by article, where the results page displays the article title, article type (randomized clinical trial, journal article etc), the sentence(s) containing the claim(s) within the article, and the year of publication.

7.5 Results and Discussion

7.5.1 Database Management System

Given the structure of PubMed articles, the raw and processed data can intuitively be viewed and represented as unstructured and structured documents, respectively. Databases such as MongoDB are built to store documents and therefore were initially tested. With the devised document structure for processed articles, preliminary developments showed complex query construction (Code Snippet 6) and slow response times.

```
    {
        "$match": {
            "pairs": {
                '$elemMatch': {
                    '$all': [
                        [idA, idB]
                    ]
                }
            }
        }
    }, {
        "$project": {
            "metadata.doi": 1,
            "metadata.publication_year": 1,
            "metadata.authors": 1,
            "metadata.title": 1,
            "metadata.filename": 1,
            "sentences": {
                "$filter": {
                    "input": {
                        "$map": {
                            "input": "$sentences",
                            "as": "sentence",
                            "in": {
                                "_id": "$sentence._id",
                                "processed_txt": "$sentence.processed_txt",
                                "associations": {
                                    "$filter": {
                                        "input": "$sentence.associations",
                                        "as": "association",
                                        "cond": {
                                            "$and": [{
                                                "$in": [idA, "$association.ids"]
                                            }, {
                                                "$in": [idB, "$association.ids"]
                                            }, ]
                                        }
                                    }
                                }
                            }
                        }
                    },
                    "as": "sentence",
                    "cond": {
                        "$and": [{
                            "$ne": ["$sentence.associations", None]
                        }, {
                            "$ne": ["$sentence.associations", []]
                        }, ]
                    }
                },
            }
        }
    }
```

Code Snippet 6. A MongoDB query that retrieves documents and the respective sentences containing a pair of entities (defined as an association). Initially the query matches documents containing both entities (anywhere in the document), and subsequently filters through the sentences to identify and return sentences that contain the entity pair. The highly convoluted query is inefficient for real-time querying and would require optimization of the document structure for improved performance.

ElasticSearch enables querying for keywords from unstructured text with real-time processing and synonym matching. However, the aim of this project was beyond look-up of entities and their respective synonyms. One of the aims of this project is to find indirectly linked associations to infer potential associations; a task and feature that is native to a graph model.

A more structured (and more pre-processed) graph model was determined to be both adequate to achieve all the aims and likely to be more efficient over less structured document-based repositories or unstructured engines. Furthermore, one of the advantages of a graph-based over a relational one is that relationships are first class citizen rather than entities (table rows in a relational database) and therefore traversing along relations is expected to be more efficient. A number of graph databases are available. Based on the open source nature, documentation, user base, and support through python with a py2neo API, we chose Neo4j as the database management system for further development of HASKEE.

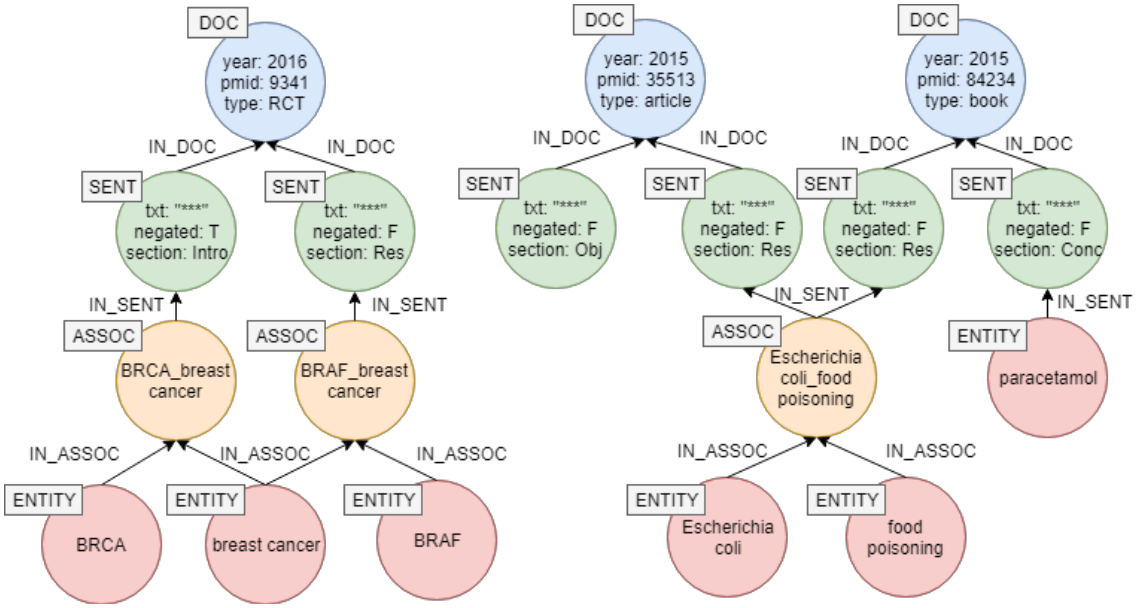


Figure 44. Final graph model. Detailed property graph model used in production, with metadata properties assigned to documents and raw sentence strings and negation assigned as properties to the sentences. The document type attribute is derived from parsed publication/document type (e.g. clinical study or in silico prediction), negation is detected by the negation cue detection module, and section represents the predicted paper section for each sentence.

7.5.2 Graph Model

The basic model unit was centered around an association node which links 2 entities by a IN ASSOCIATION relationship (Figure 43A). This allows finding entity B given entity A (and vice-versa) by querying for entities within a 1-edge distance of the node of interest. Similarly, this also enables to infer associations by querying for entities within n-edge distance of the

node of interest (Figure 43C). Each entity node will contain a unique identifier and 2 label types: 1 labeling the node type (entity) and another labeling the entity type (e.g. gene, disease, drug) to enable filtering of associations if the user queries for specific entity types at the frontend level. An alternative model structure that introduces additional edges between the entities themselves is shown in Figure 43B. While this information introduces redundancy to the model, it may improve traversing performance for obtaining directly and indirectly related entities.

In order to obtain the document(s) which state this association, a document is represented as a node linked to the association node by an IN DOC relationship. Therefore, in the final database a pair of entities is represented by one association node that can have multiple documents supporting it, and an entity node may be linked to multiple associations. This would create a chain of associations that would enable inferring indirect associations. Pre-computed weights for an association, derived from experimental design, publication type, and (predicted) paper section can either be individually stored as a property to the IN DOC edge for each document, or aggregated into a single score and saved as a property of the association node. The latter would require re-computing this weight every time a new document is added or upon deletion of a document. The former would be preferred from a maintenance point of view as the weight is calculated on the fly, however choosing which to employ depends on the added latency in the execution time and thus requires to be tested once the real dataset has been modeled and benchmarked.

The full graph model that includes document metadata, individual weights for the document, sentence and associations, and sentence negation information is shown in Figure 44.

7.5.3 Model Benchmarks

Benchmarking the preliminary model using dummy data obtained linear O(n) execution time with the number of entity nodes, up to 11 seconds for 20 million entity nodes prior to indexing the queried property (the entity id) (Figure 45Ai,ii). O(n) execution time is the best performance achieved due to the index-free adjacency property of native graph database (Weinberger, 2016), in comparison with O(n log n) complexity if using any other index. Due to potential caching, cold start (first time) queries and subsequent queries were kept separate (Figure 45A and Figure 45B). This provides a realistic execution time if the graph does not fit in the memory and therefore not all nodes and edges can be cached. A 1 second execution time difference was observed between first time queries and subsequent queries (Figure 45Ai,ii), while a O(n) execution time trend was maintained (Figure 45A). After indexing, a maximum 3000-fold improvement was achieved, averaging at 11 ms first time query execution time for 20 million entity nodes (and 2.8 ms for subsequent repeated queries) (Figure 45B). The typical brute-force computing linear trend is no longer applicable after indexing, obtaining a relatively constant execution time with increasing number of nodes. With increasing linkages, network complexity and traversing operations, the execution times are expected to be higher. This has been shown to still surpass relational database management systems execution times with increasing depth (Table 33). Furthermore, the structure of the query executed is crucial for the execution times. The preliminary benchmarks were performed by running the Cypher (neo4j's query language) query in Code Snippet 2.

Table 33. Comparison of query execution times in a relational database and neo4j with variable relation depth. Source: (Robinson, Webber, & Eifrem, 2013).

Depth	RDBMS execution time (s)	Neo4j execution time (s)	Records returned
2	0.016	0.01	~2,500
3	30.267	0.168	~11,000
4	1543.505	1.359	~600,000
5	*unfinished*	2.132	~800,000

These preliminary benchmark results provided promising indications of efficient query times. Nonetheless, further query optimizations (particularly as the graph grows) would be crucial.

As indicates in the preliminary benchmarks, to enabled quick look-up of nodes, some nodes and properties were indexed. As in querying associations the starting point of graph traversal is always the entity name (saved in the graph model as the entity id), we index this attribute. Additional indexing of document properties such as pmid may also be useful. Indexing provides quick look-up at the trade-off of graph size.

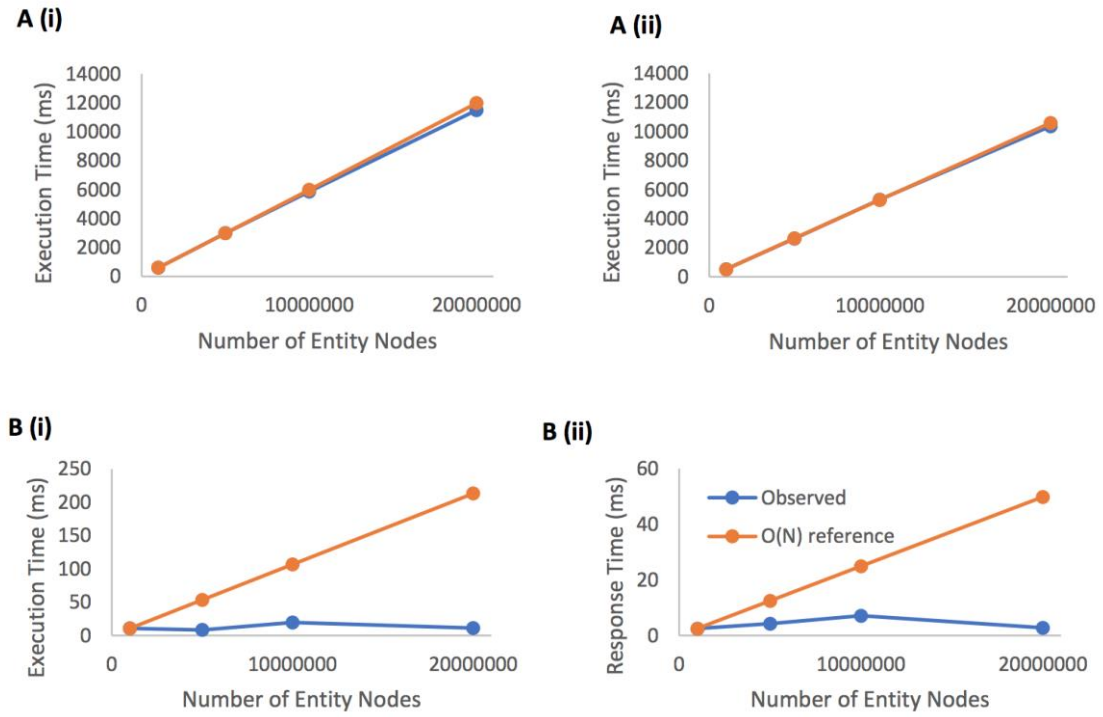


Figure 45. Scalability of neo4j. Preliminary benchmarks for the effect of node size on simple query execution time A) before indexing; and B) after indexing of the queried entity 'id' property. Identical queries were run and averaged. Due to caching of queries and results, the timings for the first queries (Ai and Bi) and subsequent repetitions (Aii and Bii) were kept separate. The difference is evident in the scale of the execution time. A linear O(n) reference line is shown with respect to the average observed execution time.

7.5.4 Populating the database

Populating the neo4j database with the associations from the associations module was initially done through a developed py2neo module that executes cypher queries in batches (or transactions) of 100 documents at a time. This would have enabled a more streamlined database population where the python pipeline would have created and populated a database following the extraction of the associations without requiring intermediary files. However, this was determined to be very inefficient, taking more than 8 hours to populate associations from only 6,800 documents.

Neo4j provides a command line executed CSV importer tool that enables import of large data in bulk. The associations were therefore exported to CSV files, with separate CSV files for each node type and edge type/label. While conversion to CSV is relatively time consuming, importing into the database now takes 30 minutes for 173 million nodes, 844 million node properties, and 766 million edges - a total of 94.56GB disk space. This makes this approach more feasible for populating the database from scratch. However, the python module can still be used for incremental additions and modifications.

7.5.5 Overall Pipeline

The overall HASKEE workflow with all its constituent modules from parsing and pre-processing, to named entity recognition, relation extraction and subsequently sub-modules is shown in Figure 3. The approach taken to develop the HASKEE pipeline renders it highly configurable. This allows users/developers to not only replace complete modules with custom functions but also varying parameters for existing implemented functions.

The pre-processing module can be customized by varying settings such as: the tokenizer, the lowercasing strategy, the removal of stopwords, the source of the stopword list, whether to expand contractions and the source of such list, the sentence classifier model directory, whether to perform negation cue detection, which sentiment algorithm to use ('match', 'vader', 'dependency' , 'sentisynsets', or a combination of any multiple of these), the significance scoring file, and the significance recognition regular expression.

The pipeline generates json files after each module. This fragmented approach requires users to re-run only specific modules if the configuration for a given module is modified or replaced, rather than running the whole pipeline. This is also useful when testing newly-developed modules as it enables to compare and contrast the module input and output json files. Ultimately, a structured json file which contains annotated entities, sentence predicted sections, significance, document metadata, and negations is generated.

The included HASKEE documentation goes beyond configuration. The documentation maximizes transparency, reproducibility and understanding of the pipeline. This is done through a user/developer-friendly set of web-pages that can be modified and updated through git (Figure 46).

Finally, to make it easier to monitor the progress of the pipeline, while logs of the current status are available through the python console, integration of the slack-progress enabled monitoring each pipeline module from anywhere through a desktop or mobile device (Figure 47).

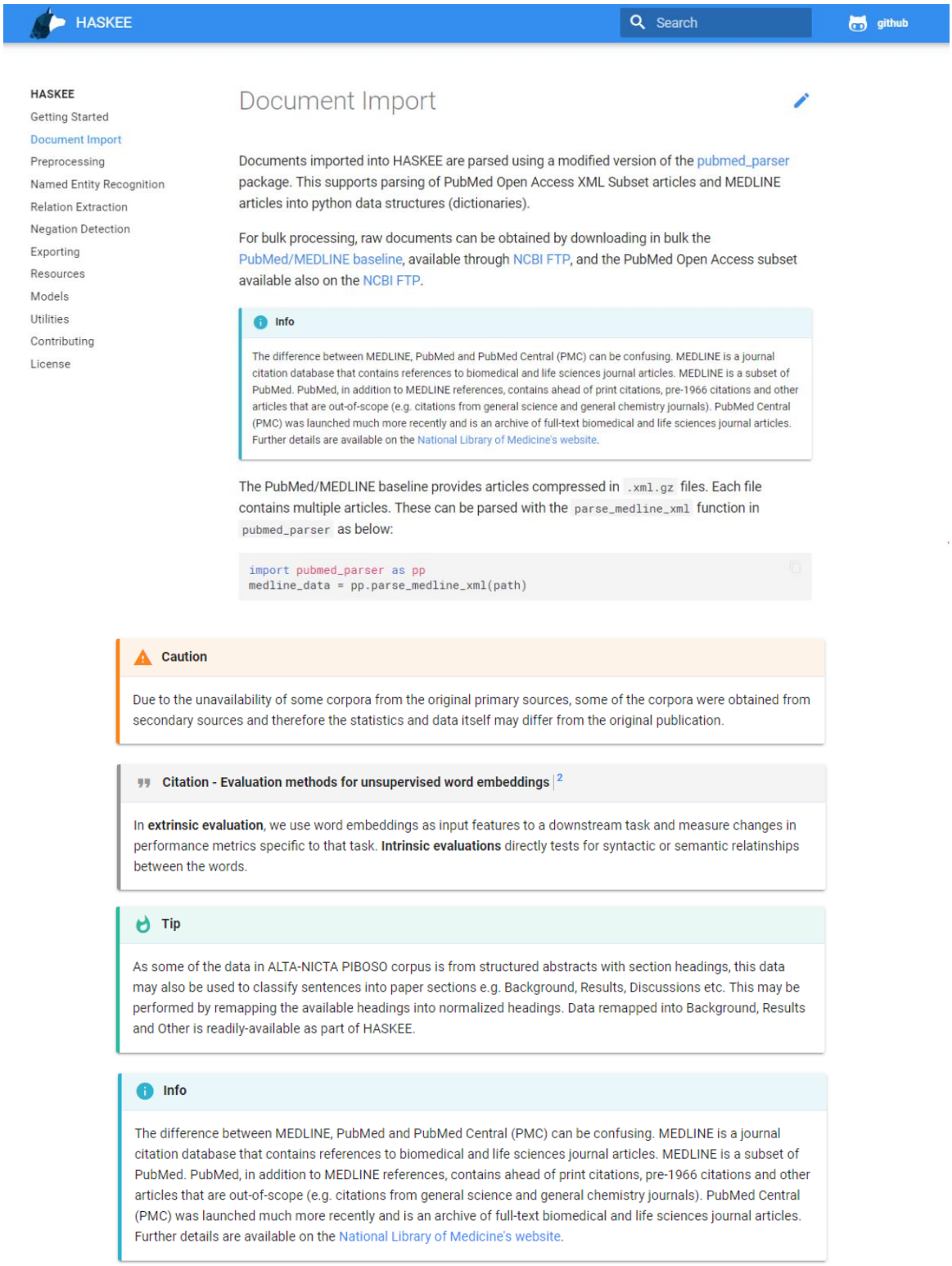


Figure 46. HASKEE documentation on each of the pipeline's modules, available resources and utilities. In addition to technical usage example, documentation includes background information, practical suggestions and warnings, and citations to original resources or publications utilizing the mentioned resource or algorithm.

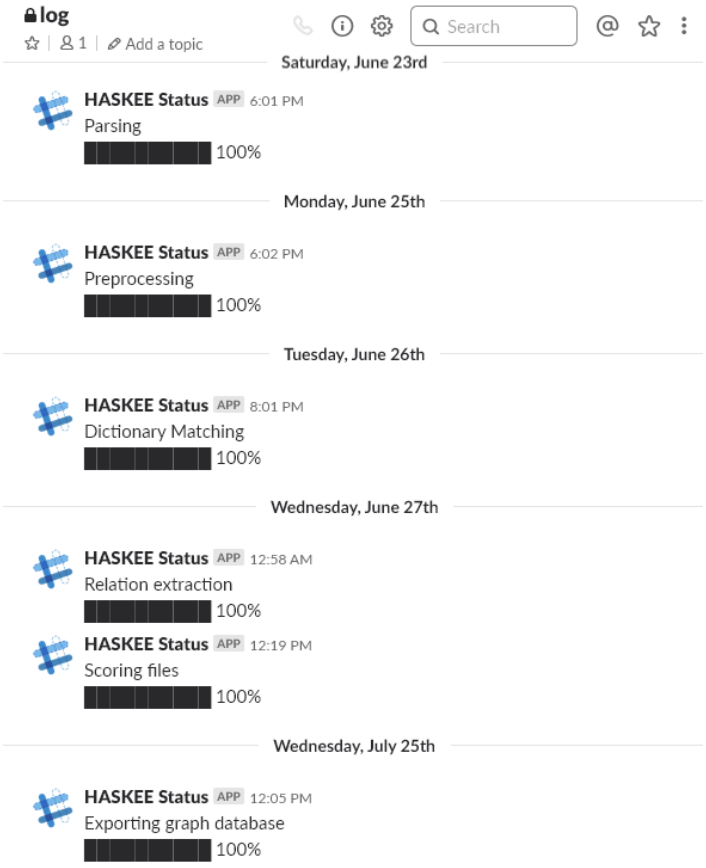


Figure 47. Slack pipeline progress monitoring. HASKEE integration of slack progress and status logging using the slack-progress library for desktop and mobile monitoring of each pipeline module.

7.5.6 Evaluation

In this work, we take a different approach to traditional rule-based methods such as in PolySearch2 (Y. Liu et al., 2015), where associations are determined by the presence of trigger words, or pure machine learning-based approaches as heavily researched on benchmark datasets. Sentences with at least a pair of entities are here considered as candidate associations. This simple co-occurrence maximizes recall however predicts poor precision due to false positives. The candidates are passed through a filter from Section 5.4.3 to determine the polarity of the association (positive or negative).

This approach mitigates the need for training data and separate models for each pair of entity classes. Fundamentally based on co-occurrence, this enables extracting associations for

unlimited number of entity class pairs, and therefore an adequate approach to be integrated with HASKEE which currently supports 10 entity classes (metabolite, gene/protein, species, chemical, toxin, drug, disease, food, food compounds and anatomy) at the NER stage (Section 6). This approach therefore allows for easy expansion for new entity classes without requiring new training data. Additionally, training corpora often do not provide labels for negated associations and therefore a trained model is unable to pickup/discriminate negative associations either – an important task which has been commonly overlooked.

Nonetheless, our approach has a number of limitations: (i) directionality is lost and therefore there is no differentiation between source and target entity; (ii) as with any binary classifier, its broad and coarse nature does not allow for differentiation between specific events such as "phosphorylates", "binds to", "causes" etc; (iii) is expected to underperform in cases where a number of entities are present in a single sentence. The latter limitation is expected to be minimized by considering frequency of entities claimed to be an association as a sorting score. To mitigate the second limitation, a fine-grained association extraction model can be subsequently applied following binary classification. This allows exploitation of the advantages of a coarse generalizable binary classifier and specificity of a fine-grained model for entity pairs with such available training datasets.

While the lack of annotated data limits the full end-to-end evaluation of the developed approach, we attempt to perform evaluation on 3 datasets: CDR (J. Li et al., 2016), DDIcorpus (Herrero-Zazo et al., 2013), and GAD (Àlex Bravo, Piñero, Queralt-Rosinach, Rautschka, & Furlong, 2015). When using co-occurrence, CDR test split obtained 40% precision and 100% recall, resulting in an F-score of 57%. This is similar performance to neural-based (CNN) models' baseline performance of 57.2% and rule-based approaches reporting an F-score of

60.8% (Lowe, O'Boyle, & Sayle, 2016). We predicted that the sentence classifier model would reduce the false positives and hence boost precision by removing sentences which do not describe an association between entities. Applying the model obtained a 32% F-score due to a lower 28% recall. This drop in recall was identified to be due to a drop in true positives as the CDR corpus considers positive associations even for sentences which are not claims/results, for example:

> "This study investigated behavioral phenotypes and AChE activity in male mice following BPA exposure during puberty."

> "OBJECTIVE: The objective of this study was to determine the risk of lifetime and current methamphetamine-induced psychosis in patients with methamphetamine dependence."

> "METHODS: This was a cross-sectional study conducted concurrently at a teaching hospital and a drug rehabilitation center in Malaysia."

To provide a more realistic performance metrics for our approach, we re-evaluate performance by only considering claims. As the sentence classifier was determined to be minimum 96% accurate for conclusion and result sentences collectively (Figure 33), we consider this as the recall rate for positive associations. The precision estimate therefore becomes dependent on the distribution of negative and positive associations in the benchmark dataset. For the CDR corpus, this results in an F-score of 54.4%. Evaluation on the GAD corpus was performed similarly. As this corpus only contains claims (conclusions), we estimate recall to be 96% based on claim-detection model accuracy for conclusions and results. Based on the proportion of

positive to negative samples, this results in 52.6% precision for the co-occurrence strategy, resulting in an overall F-score of 68.0% (in comparison with the machine learning-based approaches by BeFree reporting a maximum F-score of 81.9%, although for the depression subset an F-score of 64.2% was reported) (À Bravo, Cases, Queralt-Rosinach, Sanz, & Furlong, 2014; Àlex Bravo et al., 2015).

In terms of absolute performance metrics, chemical-disease association results are comparable to state-of-the-art methods, however, other results such as gene-disease associations may appear much inferior to latest research, with a difference of up to 13.9% in F-score. However, given the utilized approached is not supervised, does not require training data and is therefore most generalizable, this sub-optimal performance is not poor. This is further backed up by results obtained with adversarial techniques. In adversarial methods, training is done on available data that is then transferred to a new domain with no data available, mitigating the need for training data. (Rios, Kavuluru, & Lu, 2018) have shown that such approach results in a loss of 18-33.57% F-score when performance is compared to a model trained on the same dataset. This performance loss can be seen as overfitting or bias to the training data and therefore comparing this lower accuracies to our results is a more fair comparison.

Beyond benchmark datasets and quantitative evaluation, we performed qualitative assessment. In initial pre-runs, before integration of the sentence classifier model, we identified that a large number of returned associations did not contain a claim of finding an association. Rather, returned sentences were hypothesis, aims or background. For example, when querying association between pantoprazole and cancer, one of the outputs was:

> "To compare safety and efficacy of **pantoprazol**, metoclopramide, ondansetron, as compared to placebo, in controlling gastrointestinal (GI) complaints of thyroid **cancer** patients […]."

This justified the development and highlighted the importance of the sentence classifier. In the final version, we observe that filtering the output for sections predicted to be 'results' or 'conclusion', outputted sentences only containing claims such as:

> "Our study indicates that LH-RH antagonist **Cetrorelix** may inhibit the growth of DU-145 human androgen-independent **prostate cancers** by […]"

In Section 3.5.1 we report drugs with the potential for anti-cancer properties, including: phenmetrazine, fluticasone furoate, cefradine, flunisolide, cetrorelix, gentian violet, and celecoxib. Cetrorelix identified 7 results statements and 16 conclusion statements, which upon review were confirmed to be all valid; claiming there is a potential association between cetrorelix and cancer. A subset of which is listed here:

> "Our findings indicate that the anti-proliferative effects of GHRH antagonist MZ-J-7-138 and LHRH antagonist **Cetrorelix** on **prostate cancers** involve p53 and p21 signaling."

> "Our results indicate that the bombesin antagonist RC-3095 and the LH-RH antagonist **Cetrorelix** inhibit effectively the growth of ES-2 **ovarian cancers** in nude mice."

> "The ability of **Cetrorelix** to downregulate EGFR signalling and subsequently reverse the antiadhesiveness found in metastatic **prostate cancer** highlights a novel potential target for therapeutic strategies."

Similar results were obtained for celecoxib, with 147 result statements and 254 conclusion statements claiming an association. Indeed, this is a compound that has been well-reported for its anti-cancer properties (Koki & Masferrer, 2002; B. Song, Shu, Du, Ren, & Feng, 2017; Winfield & Payton-Stewart, 2012).

Gentian violet is commonly used as a histological dye in cancer assays. A total of 9 statements were extracted. However, as our relation extraction is based on co-occurrence of entities, as expected, recalled statements stated the use of crystal violet as an assay in determining an association between cancer and other entities. This is demonstrated by the below example:

> "The in vitro inhibitory activity of BBPW-2 was measured using MTT and **crystal violet** assays, which suggested that BBPW-2 had direct cytotoxic effects on the **cancer** cell lines HeLa and HepG2 (particularly HeLa cells), and had a long-term antiproliferative effect on MCF-7 cells, respectively."

In the above example, while the sentence is indeed a claim, this is not between the entities of interest (i.e. gentian violet and cancer) but rather between BBPW-2 and cancer. This case demonstrates that in some cases, there is a need for a more fine-grained relation extraction model that is entity-specific rather than at the sentence level, as currently performed by HASKEE.

HASKEE did not return any results claiming a link with cancer for flunisolide. Querying PubMed returns 3 results (Table 34).

Table 34. Results returned by PubMed for a query "flunisolide cancer".

Paper title	PMID	Statement(s)
Molecular mechanisms of increased nitric oxide (NO) in asthma: evidence for transcriptional and post-translational regulation of NO synthesis.	10820280	*NA*
Drug solubilization in lung surfactant	10699268	"The aqueous solubility and the extent of solubilization in Survanta, a native extract of bovine lung, of budesonide, triamcinolone acetonide, dexamethasone, and **flunisolide** were determined as a function of temperature by a dialysis technique."
Prophylactive treatment with flunisolide after polypectomy	6753092	"Prophylactic treatment with **flunisolide** can be recommended as a complement to other treatment after surgery of nasal polyps"

These search results indicate no relevance to cancer in the case of PMID:10820280, or very indirect link through solubility of lung surfactant (PMID:10699268) or nasal polyps, which may not necessarily be cancerous (PMID:6753092), validating the results returned by HASKEE.

When querying for an association between cefradine and cancer, initial versions of HASKEE returned 5 statements. These statements were recalled due to the CED synonym for cefradine. However, in all 5 cases, CED was an abbreviated form for another term. This justifies the need for an abbreviation solver module, which upon completion, filtered out such statements. Performing the same query in PubMed returns 17 items, however these reported the link between cephradine and antibiotics, rather than cancer. Cephradine targets bacterial receptors

and a link to cancer may potentially be through shared pathways; a finding which has not yet been reported in literature.

HASKEE also did not return any output for a link between phenmetrazine and cancer. PubMed returns 1 article which claims other studies have previously reported a link; the source of which is not reported and therefore couldn't be traced. HASKEE identified this claim as background and therefore did not return such result when querying only results and conclusions.

Fluticasone furoate returns 4 results in PubMed when queried for a link with cancer (Table 35). On inspection, none of these statements recalled define such association.

Table 35. Results returned by PubMed for a query "fluticasone fuorate cancer".

Paper title	PMID	Statement(s)
Pharmaceutical Approval Update	26185402	"Deoxycholic acid (Kybella) for submental fat; codeine polistirex/chlorpheniramine polistirex extended-release (Tuzistra XR) for coughs, allergies, and colds; ramucirumab (Cyramza) for colorectal cancer; and **fluticasone furoate**/vilanterol (Breo Ellipta) for asthma."
Discovery of a highly potent glucocorticoid for asthma treatment	27066265	"VSGC12 also showed a higher potency than **fluticasone furoate** in repressing most asthma symptoms."
Pharmaceutical approval update	24049424	"**Fluticasone furoate**/vilanterol inhalation powder (Breo Ellipta) for chronic obstructive pulmonary disease; atorvastatin/ezetimibe tablets (Liptruzet) for reducing low-density lipoproteincholesterol; and radium 223 dichloride (Xofigo) injection for late-stage, castration resistant prostate **cancer**."
Gateways to clinical trials	17805439	"Caldaret hydrate, **Cancer** vaccine, Cediranib, CHAMPION everolimus-eluting coronary stent […] Everolimus; **Fluticasone furoate**; Glucarpidase; […]"

These results collectively indicate that whereas quantitatively HASKEE performs lesser compared to state-of-the-art models trained on benchmark datasets, when considering model bias, generalizability, and versatility, performance is promising. This can be seen from the quantitative results, where in comparison to popular engines like PubMed which is a pure co-occurrence-based approach, HASKEE achieves a reasonable recall while filtering out false associations. Nonetheless, improving HASKEE further to mitigate the limitations discussed, a hierarchical approach can be taken in future work, where a broad classifier is applied first, followed by a more fine-grained state-of-the-art model for which training data is available. Rather than discarding associations when identified as false, they can be labelled appropriately and stored in the graph model. This is the approach taken with HASKEE which allows to adjust recall/precision at the querying stage and therefore does not require to re-process the whole data or re-run the pipeline.

7.5.7 Frontend proof of concept

The developed proof of concept shows an exemplary interface where given a query for associations between entities such as “cetrorelix” and “cancer”, this connects to the dictionary graph database to retrieve their respective synonyms and hyponyms, and subsequently query the associations graph for articles claiming such an association.

This two-step process enables users to have fine-grained control on the synonyms and therefore we show an exemplary implementation as how a user can add/modify synonyms for each entity (Figure 49). This is useful in cases where the dictionary graph populated has noisy or ambiguous synonyms for a specific entity, therefore rather than rendering the results completely false, the user can control this. By disabling a synonym, the association query is updated and therefore the list of results is updated respectively. In Figure 49 we show how additional control on the synonyms can be implemented; i.e. controlling synonyms based on

the source, or even relationship type. These features are fully supported with the devised dictionary graph structure.

The returned associations results (Figure 48) list the claims supporting the association between the identified entities in form of sentences grouped by article. Each article is represented by the article title, article type(s), PubMed identifier (PMID), sentences containing the claim, as well as the year of publication. To enable user feedback to be registered, a feedback button can be integrated that allows users to report cases where an article does not in fact support a claim i.e. in cases of false positives. Additionally, a dropdown button can enable additional details about the article.

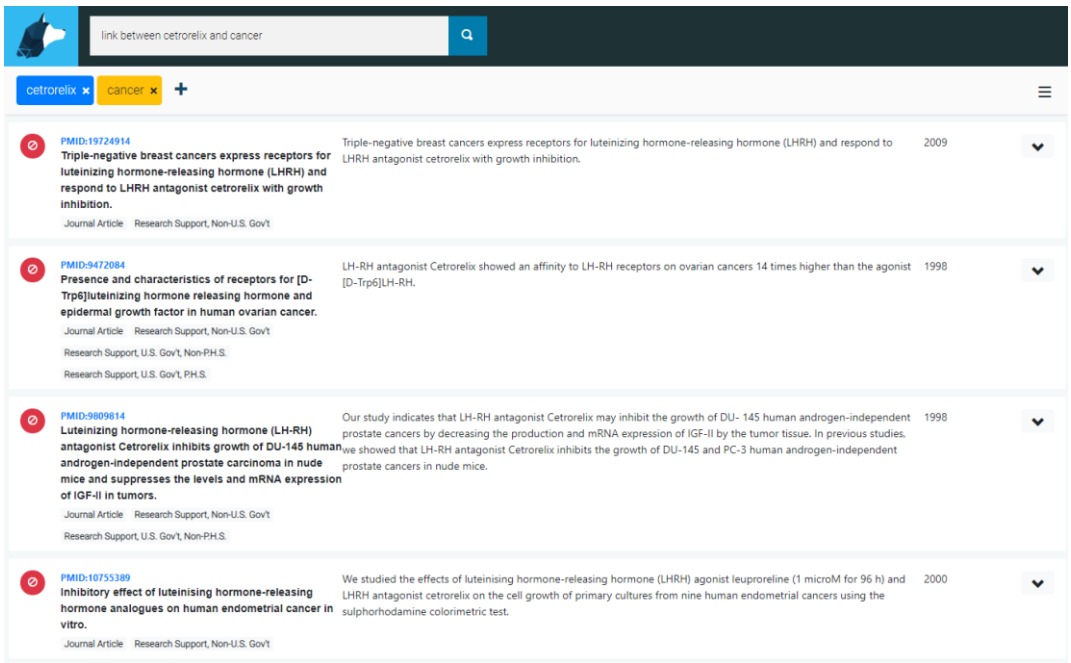


Figure 48. Screenshot of the proof of concept for the results page when querying for a link between "cancer" and "cetrorelix". The recognized entities are listed and articles claiming an association are listed below. Recognized entities can be added and/or modified by the user in case of false positive and false negative entities recognized from the inputted statement. Articles are represented by their PubMed identifier (PMID), their article type, the sentences claiming such association, and the year of publication. Each entry could have a button (represented by a red circular button in the screenshot here) that enables to capture user feedback in instances where the article is falsely recalled. Additionally, a dropdown button can provide additional metadata and article information.

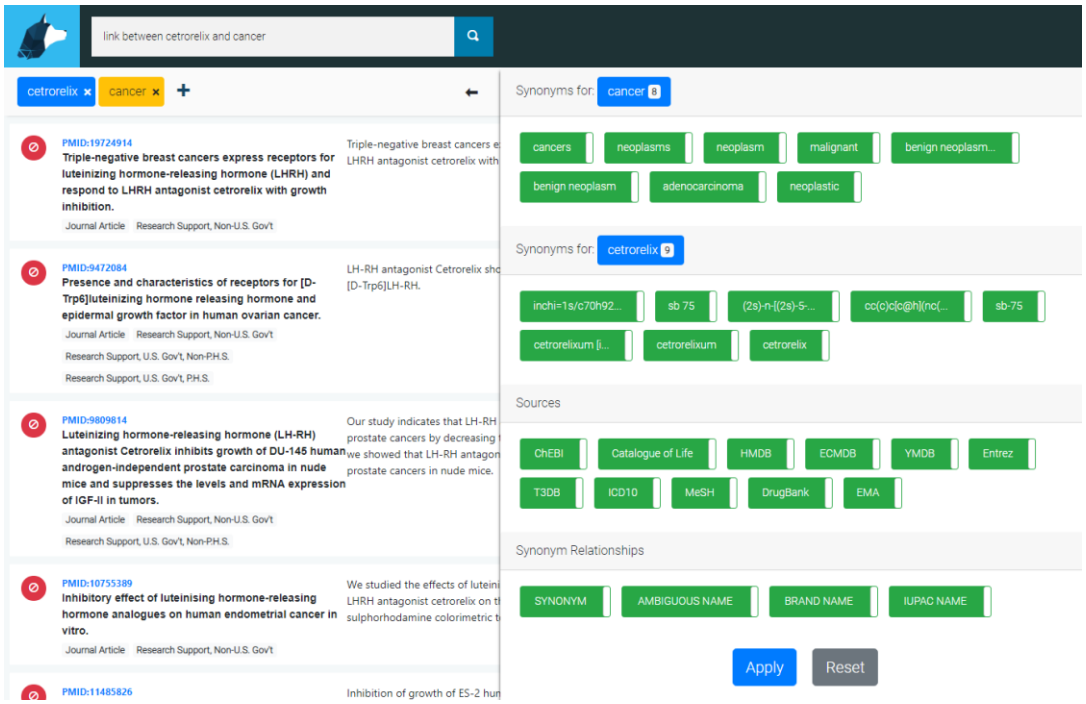


Figure 49. Screenshot of the proof of concept for the results page when querying for a link between "cancer" and "cetrorelix" showing the retrieved synonyms from the dictionary graph for each of the recognized entities.

7.5.8 Towards automating systematic reviews

In Section 3.4.3 we introduced systematic reviews, with their automation as one of the ultimate aims for utilizing text mining tools. With the development, discussion and promising results achieved by HASKEE, automating the extraction of associations between entities such as biomarkers and cancer can be seen as an achievable task. However, automating systematic reviews as manually performed in Section 3.4.3 requires additional research developments. Screening of titles and abstracts is required to determine if a study is eligible to be included in a systematic review. This involves identifying the PIBOSO elements (Population, Intervention, Background, Outcome, Study Design, Other) and determining if these match the requirements and guidelines implemented by bodies such as Cochrane Library. Furthermore, whereas in HASKEE (and BioNLP research in general) we have developed and implemented a named entity recognition module, phrase/chunk recognition is required in order to reproduce work like

(Poynter et al., 2019) looking at broader concepts (rather than entities) such as immunotherapy response.

On a broader level, systematic reviews only utilize randomized clinical trials. This therefore requires an additional layer in the pipeline for the identification of randomized clinical trials from other publications. While PubMed labels such articles as part of their respective metadata, mislabelling of such studies is not uncommon (James Thomas, personal communication) and therefore requires improvements.

The developments required to automate systematic reviews have been detailed in the Vienna principles (Beller et al., 2018). In addition to the specific tasks, these include the requirement for the processes of each task to be continuously improved, to be more efficient and accurate. This principle was employed during the development of HASKEE as is fundamental in incremental tool development and to maximize usage versatility.

7.6 Conclusion(s) and Future Direction(s)

In this work we have developed a graph-based repository for querying associations extracted from literature using the documented HASKEE pipeline developed throughout this project and demonstrated a proof of concept for a complementary frontend. Qualitative evaluation indicates the effectiveness of the abbreviation solver module implemented, as well as the sentence classifier in reducing noisy sentences recalled as 'associations'. Preliminary quantitative evaluation shows promising generalizable results. Future work can utilize additional machine learning techniques to recall cross-sentence n-ary associations, as well as

combine state-of-the-art machine learning methods to the current approach by providing and storing probability of association as an additional score in the association node.

While complete in terms of an end-to-end pipeline, this work can be subjected to a number of improvements and additional research. A lot of effort has been put to maximize transparency, define the pipeline with a modular structure, and provide developer-friendly documentation. This is expected to facilitate future developments and incremental improvements of the open-source HASKEE pipeline. Further work on the frontend can maximize the highly diverse and flexible underlying model structure while providing a user-friendly interface.

Chapter 8 – Conclusions and future work

In this work we started by demonstrating how different methods can make use of a natural language processing pipeline to automate biomedical entity link searches (Chapter 3) and subsequently investigated different natural language processing tasks required to achieve this.

Fundamentally, in Chapter 4, we have investigated and established state-of-the-art biomedical word representations trained in an unsupervised fashion. Word representations are critical for any downstream machine learning task, and we have shown how through optimization of hyper-parameters, our models out-perform previous solutions. Because of this, these models were incorporated with HASKEE to be utilized in all the modules. Nonetheless, the current approach does not exploit the information-rich knowledge base resources available in the biomedical field, therefore future work can investigate the role of integrating knowledge base information during training to boost performance further. This may increase performance of subsequent tasks such as sentence classification and named entity recognition.

In the first pipeline module tackling article pre-processing, detailed in Chapter 5, a number of sub-tasks are handled, including: article and metadata parsing, negation detection and abbreviation resolution. Extending parsing to metadata enables HASKEE to score biomedical association based on multiple variables, including: recency, journal types (where an association extracted from randomized clinical trial can be given higher score over other article types), etc.

Furthermore, by incorporating a negation detection module, HASKEE allows for recognizing negated sentences/associations. Although publication bias for positive results over negative ones is a well-known issue in research, recognition of negative results is critical for factors

such as reproducibility, and robustness of findings/claims. HASKEE has been evaluated to be highly performant in recognizing negations, however a limitation is that negation is currently done at the sentence-level. This implies that if multiple associations are described in a sentence with different polarities, all the associations are considered negated. This issue can be mitigated in the future by integration of a negation scope detector which has been preliminarily investigated here.

An additional feature/advantage of HASKEE is the ability to resolve abbreviations. Through extending the Schwartz algorithm, HASKEE recalls entity mentions even in the abbreviated and shortened forms, including shortened species names. These are fairly common in biomedical literature and therefore enables an improvement on recalling such entities. Unfortunately, due to lack of evaluation data, the impact of the extension presented here was unable to be quantified. In future work, developing a dataset would be required to quantify this as well as to evaluate the pre-processing module developed end-to-end.

In the second pipeline module, we have investigated and implemented a named entity recognition module (Chapter 6). Needing a platform that scales to a large number of entity classes, we achieved this by implementing dictionary matching approaches. Dictionary matching approaches are relatively simple but also static, and are therefore unable to recognize, for example, more dynamic entities such as variants. This can be mitigated through machine learning models which however make scalability a more challenging task, due to predominantly availability of data and data bias. We have shown that while machine learning approaches provide highly promising results, these are biased by the dataset that the models are trained on. Generalizing the models by collating training data from different sources

decreases absolute performance but is expected to be more realistic. Through power analyses we have also shown that collecting more data is unlikely to increase performance.

This work has focused on traditional sequential machine learning models whose performance is highly dependent on manual feature extraction. With newer techniques such as embeddings and neural models, feature extraction is no longer required and therefore the findings presented here may not hold fully. For example, as neural models are known to require a large number of training samples, it is likely that more training data is required to reach the optimal performance.

Nonetheless, naturally, the next step for machine learning-based named entity recognition would be to integrate this as part of the HASKEE pipeline. Integration is trivial with HASKEE's modular structure, however there are a number of additional investigations that would be required with the current implementation: (i) how scalable is feature extraction to apply to the PubMed scale?; (ii) would a neural model with pretrained embeddings be more efficient as the feature vector is pre-computed and therefore obtaining the feature vector is only a look-up?; (iii) would a neural model require more training data? i.e. does the power analyses executed for a CRF-based model still apply?

Additionally, from an end-to-end pipeline perspective, with a machine learning-based NER approach, recognized entities require a "normalizing" module that maps a discovered entity to a knowledge base/dictionary. This is advantageous when entities discovered are not mapped one-to-one to a knowledge database (e.g. because of typos), however the extent at which this occurs is unknown. If a large number of the discovered entities are indeed in the knowledgebase/dictionary, this is expected to achieve similar performance to a dictionary-

matching implementation. One potential approach could utilize both machine learning-based methods as well as dictionary-based matching. The proposed approach is to perform NER with both machine-learning approaches as well as dictionary-based approaches, however use the former to help curate the latter. If ML-based NER identifies entities which are not in the dictionary, these can be reviewed and added to the dictionary accordingly. Additionally, with the implemented HASKEE graph model, a tag can be added to the entity whether this was identified by machine learning methods or dictionary methods. This provides a highly flexible filter especially if NER performance is entity class dependent.

With both dictionary and ML-based approaches investigated here, to enable a fair comparison as well as robustness of the implementations, evaluation of the trie-based implementation is required. The trie-based approach enables recalling nested entities and therefore is expected to recall a much larger number of entities. However, this requires extensive evaluation effort as benchmark datasets for nested entities at the scale of HASKEE is not available, to our knowledge. A dataset is required to cover a diverse range of entity classes to realistically evaluate the HASKEE dictionary-based NER module. For example, considering the example string: “human colon cancer”, “human” is a species entity, “colon” is an organ/anatomy entity, and “cancer” and “human colon cancer” are disease entities. This therefore requires a benchmark dataset which considers all 3 entity classes in a nested fashion. In the HASKEE scale, this would require a dataset which considers all 10 entity classes.

As for associations (Chapter 7), we have developed and shown that a claim detection model is a highly generalizable and performant, achieving up to 89% accuracy. This has the clear advantage of being highly generalizable and therefore scales to a large number of entity classes. However, because this is a sentence-level model, multiple associations within a sentence or

multiple entities within a sentence are treated equally. Additionally, the sentence-level approach taken here loses information in the directionality of the association (i.e. A causes B and B causes A are not differentiated). While here we attempted to avoid training a model per entity class pair due to high specialization (and therefore poor generalizability) and expense of training data, an alternative future approach would be to have a hierarchy of models. The claim detection model can be used as an initial screen, filtering out sentences with co-occurring entities that are not describing a claim, and subsequently, based on the entity class of the entities, a more specialized model is applied. The advantage of such approach is that if training data is not available for a specific entity class pair (e.g. toxin-drug), associations can still be computed for such pair as the claim detection model is not entity class specific.

An extensive amount of effort was done during the HASKEE pipeline design and development in order to ensure optimal structure for wide and diverse capabilities from a frontend perspective. Fully developing a complimentary frontend would make the computed association graph more readily accessible to users. The devised graph model structure enables very diverse queries to be executed. We envision a free text search which enables queries such as: “what treats headaches?” and “vaccines cause”. These are open-ended queries where in the first case the system identifies headaches as a condition/disease and executes a query for the associations with it. In the second case, this is an autocomplete task, where given “vaccines” is recognized as a drug followed by the verb “cause”, an autocomplete model predicts the next entity to be “side effect” or “disease”. Subsequently, having determined this, at the backend of the search engine, a query is executed to extract associations between the drug entity “vaccine” and all other “side effects”. This autocomplete model can be trained by utilizing sentences identified as claims during the construction of the association graph. Given sentences such as “celecoxib can cause an upset stomach”, identifying “celecoxib” as a drug and “upset stomach” as a side-

effect, replacing the original tokens by the identified entity class would generate the sentence: “DRUG can cause a SIDE-EFFECT”. Truncating the sentence before the mention of the second entity would generate the training sentence “DRUG can cause a” where the label/word to be predicted is “SIDE-EFFECT”.

Other examples of queries that are supported by the devised graph model are closed queries. This has been demonstrated in the proof of concept which can be extended further to provide additional claim information such as negation information, additional article metadata as well as a ranking score.

Additionally, one of the major advantages of pre-computing associations and building a network is the ability to traverse the graph. So far, query examples recalled direct links between a pair of entities where such entities were mentioned in literature. However, the devise graph model allows to find associations that have not been explicitly mentioned in literature and therefore associations can be inferred. Having a frontend which supports a “discovery” mode would allow for novel associations to be discovered. Such mode would require to execute queries where given entity “X”, links to “Y” are extracted as long as there is no direct links between X → Y. The confidence of such inference can also be adjusted by defining a path distance threshold. Considering the path: A → B → C → D, where → represents a direct link (i.e. co-occurring entities), setting a threshold distance of 2 would extract entity C, given entity A as a seed. Adjusting the threshold distance to 3 would also extract entity D as potentially associated with A.

The devised dictionary graph also allows traversal through ontologies. Given the exemplary association pair (“Staphylococcus aureus”, “skin infection”) which would recall sentences

claiming a link between “Staphylococcus aureus” and “skin infection”, as “Staphylococcus aureus” is in the species graph linked by an IS_A relationship to the genus “Staphylococcus”, this association search can be interactively broadened by traversing through the IS_A relationship and therefore modifying the association pair to be: (“Staphylococcus”, “skin infection”) where “Staphylococcus” now recalls all mentions of species belonging to that genus. Similarly, for chemicals, for example: (“methanol”, “blindness”), as “methanol” is in the “alcohol” ontology, a “go broader” feature would traverse through the IS_A link and modify the association query to (“alcohol”, “blindness”), where the “alcohol” entity now recalls recursively all entities such as “ethanol” and “methanol”.

The advantage of having a generic graph model to fit the diverse queries comes with a likely disadvantage of sub-optimal processing times. Devising an optimal model for the different queries is likely to outperform a generic model in terms of query time, however this would in turn result in data redundancy, requirement for additional resources, and makes maintenance more demanding.

Finally, in attempt to measure the overall end-to-end performance of the HASKEE pipeline, we queried the database of compiled associations for closed associations between cancer and specific drugs identified to be potentially anti-cancerous. This was a semi-quantitative approach that enabled to identify the strengths of HASKEE but more rigorous evaluation is required through development of an adequate dataset. Additionally, evaluation of new ‘undocumented’ discoveries enabled by HASKEE’s traversal of pre-computed associations would require validation by an expert panel of researchers.

To conclude, the developed HASKEE pipeline is a modular and highly extendible framework that along with the structure of its components, enables its utility in diverse applications. The open-source nature of the pipeline enables integration of existing solutions, enabling for a continuously improved end-to-end pipeline for extracting biomedical associations.

References

Achakulvisut, T., Acuna, D. E., Cybulski, T., Hassan, T., Badger, T. G., H-Plus-Time, & Brandfonbrener, D. (2016). *titipata/pubmed_parser: Pubmed Parser*. https://doi.org/10.5281/zenodo.159504

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., & Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, *9*(11), S2. https://doi.org/10.1186/1471-2105-9-S11-S2

Alharbi, E., & Tiun, S. (2015). A hybrid method of linguistic features and clustering approach for identifying biomedical named entities. *Asian Journal of Applied Sciences*, *8*(3), 210–216. https://doi.org/10.3923/ajaps.2015.210.216

Alshaikhdeeb, B., & Ahmad, K. (2016). Biomedical Named Entity Recognition: A Review. *International Journal on Advanced Science, Engineering and Information Technology*, *6*, 889. https://doi.org/10.18517/ijaseit.6.6.1367

Amini, I., Martinez, D., & Molla, D. (2012). Overview of the ALTA 2012 Shared Task. In *Proceedings of the Australasian Language Technology Association Workshop 2012* (pp. 124–129). Dunedin, New Zealand. Retrieved from http://aclweb.org/anthology/U12-1017

Ayoub, N. (2018). *Optimisation of word representations and neural architectures for biomedical named entity recognition*. Imperial College London, London.

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*(7604), 452–454. https://doi.org/10.1038/533452a

Banerjee, S., & Pedersen, T. (2002). An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In A. Gelbukh (Ed.), *Computational Linguistics and*

Intelligent Text Processing (pp. 136–145). Berlin, Heidelberg: Springer Berlin Heidelberg.

Barbosa-Silva, A., Fontaine, J.-F., Donnard, E. R., Stussi, F., Ortega, J. M., & Andrade-Navarro, M. A. (2011). PESCADOR, a web-based tool to assist text-mining of biointeractions extracted from PubMed queries. *BMC Bioinformatics*, *12*(1), 435. https://doi.org/10.1186/1471-2105-12-435

Barrett, N., & Weber-Jahnke, J. (2011). Building a biomedical tokenizer using the token lattice design pattern and the adapted Viterbi algorithm. *BMC Bioinformatics*, *12*(3), S1. https://doi.org/10.1186/1471-2105-12-S3-S1

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An Open Source Software for Exploring and Manipulating Networks. Presented at the International AAAI Conference on Weblogs and Social Media.

Bateman, A., Martin, M. J., O'Donovan, C., Magrane, M., Alpi, E., Antunes, R., … Zhang, J. (2017). UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, *45*(D1), D158–D169. https://doi.org/10.1093/nar/gkw1099

Batista-Navarro, R., Rak, R., & Ananiadou, S. (2015). Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics. *Journal of Cheminformatics*, *7*(Suppl 1), S6. https://doi.org/10.1186/1758-2946-7-S1-S6

Becker, K. G., Barnes, K. C., Bright, T. J., & Wang, S. A. (2004). The genetic association database. *Nature Genetics*, *36*(5), 431–432. https://doi.org/10.1038/ng0504-431

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720. https://doi.org/10.1109/34.598228

Beller, E., Clark, J., Tsafnat, G., Adams, C., Diehl, H., Lund, H., … On behalf of the founding members of the ICASR group. (2018). Making progress with the automation of systematic reviews: principles of the International Collaboration for the Automation of Systematic Reviews (ICASR). *Systematic Reviews*, *7*(1), 77. https://doi.org/10.1186/s13643-018-0740-7

Bhasuran, B., Murugesan, G., Abdulkadhar, S., & Natarajan, J. (2016). Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases. *Journal of Biomedical Informatics*, *64*, 1–9. https://doi.org/10.1016/j.jbi.2016.09.009

Bitola (Biomedical Discovery Support System). (2008). In *Encyclopedia of Genetics, Genomics, Proteomics and Informatics* (pp. 219–219). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-1-4020-6754-9_1861

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017a). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017b). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, *17*(12), 2301–2309. https://doi.org/10.1109/TVCG.2011.185

Bravo, À, Cases, M., Queralt-Rosinach, N., Sanz, F., & Furlong, L. I. (2014). A Knowledge-Driven Approach to Extract Disease-Related Biomarkers from the Literature [Research article]. https://doi.org/10.1155/2014/253128

Bravo, Àlex, Piñero, J., Queralt-Rosinach, N., Rautschka, M., & Furlong, L. I. (2015). Extraction of relations between genes and diseases from text and large-scale data

analysis: implications for translational research. *BMC Bioinformatics*, *16*(1), 55. https://doi.org/10.1186/s12859-015-0472-9

Brigadir, I. (2018). *Default English stopword lists from many different sources: igorbrigadir/stopwords*. Python. Retrieved from https://github.com/igorbrigadir/stopwords (Original work published 2016)

Brown, A. S., & Patel, C. J. (2017). A standard database for drug repositioning. *Scientific Data*, *4*. https://doi.org/10.1038/sdata.2017.29

Bundschus, M., Dejori, M., Stetter, M., Tresp, V., & Kriegel, H.-P. (2008). Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*, *9*(1), 207. https://doi.org/10.1186/1471-2105-9-207

Campos, D., Matos, S., & Oliveira, J. L. (2013a). A modular framework for biomedical concept recognition. *BMC Bioinformatics*, *14*(1), 281. https://doi.org/10.1186/1471-2105-14-281

Campos, D., Matos, S., & Oliveira, J. L. (2013b). Gimli: open source and high-performance biomedical name recognition. *BMC Bioinformatics*, *14*(1), 54. https://doi.org/10.1186/1471-2105-14-54

Caporaso, J. G., Deshpande, N., Fink, J. L., Bourne, P. E., Cohen, K. B., & Hunter, L. (2008). Intrinsic evaluation of text mining tools may not predict performance on realistic tasks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 640–651.

Cejuela, J. M., Bojchevski, A., Uhlig, C., Bekmukhametov, R., Kumar Karn, S., Mahmuti, S., … Wren, J. (2017). nala: text mining natural language mutation mentions. *Bioinformatics*, *33*(12), 1852–1858. https://doi.org/10.1093/bioinformatics/btx083

Chan, P. P., Wasinger, V. C., & Leong, R. W. (2016). Current application of proteomics in biomarker discovery for inflammatory bowel disease. *World Journal of Gastrointestinal Pathophysiology*, *7*(1), 27–37. https://doi.org/10.4291/wjgp.v7.i1.27

Chang, J. T., Schutze, H., & Altman, R. B. (2002). Creating an online dictionary of abbreviations from MEDLINE. *Journal of the American Medical Informatics Association : JAMIA*, *9*(6), 612–620.

Chang, Y.-C., Chu, C.-H., Su, Y.-C., Chen, C. C., & Hsu, W.-L. (2016). PIPE: a protein–protein interaction passage extraction module for BioCreative challenge. *Database: The Journal of Biological Databases and Curation*, *2016*. https://doi.org/10.1093/database/baw101

Chen, E. S., Hripcsak, G., Xu, H., Markatou, M., & Friedman, C. (2008). Automated acquisition of disease drug knowledge from biomedical and clinical documents: an initial study. *Journal of the American Medical Informatics Association: JAMIA*, *15*(1), 87–98. https://doi.org/10.1197/jamia.M2401

Cheng, D., Knox, C., Young, N., Stothard, P., Damaraju, S., & Wishart, D. S. (2008). PolySearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Research*, *36*(Web Server issue), W399-405. https://doi.org/10.1093/nar/gkn296

Chiu, B., Crichton, G., Korhonen, A., & Pyysalo, S. (2016). How to Train good Word Embeddings for Biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing* (pp. 166–174). Berlin, Germany: Association for Computational Linguistics.

Chiu, B., Korhonen, A., & Pyysalo, S. (2016). Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP* (pp. 1–6). Berlin, Germany: Association for Computational Linguistics. Retrieved from http://anthology.aclweb.org/W16-2501

Chun, H.-W., Tsuruoka, Y., Kim, J.-D., Shiba, R., Nagata, N., Hishiki, T., & Tsujii, J. (2006). Extraction of gene-disease relations from Medline using domain dictionaries and

machine learning. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 4–15.

Comeau, D. C., Islamaj Doğan, R., Ciccarese, P., Cohen, K. B., Krallinger, M., Leitner, F., … Wilbur, W. J. (2013). BioC: a minimalist approach to interoperability for biomedical text processing. *Database*, *2013*. https://doi.org/10.1093/database/bat064

Corbett, P., & Copestake, A. (2008). Cascaded classifiers for confidence-based chemical named entity recognition. *BMC Bioinformatics*, *9*(Suppl 11), S4. https://doi.org/10.1186/1471-2105-9-S11-S4

Corney, D. P. A., Buxton, B. F., Langdon, W. B., & Jones, D. T. (2004). BioRAT: extracting biological information from full-length papers. *Bioinformatics (Oxford, England)*, *20*(17), 3206–3213. https://doi.org/10.1093/bioinformatics/bth386

Crichton, G., Pyysalo, S., Chiu, B., & Korhonen, A. (2017). A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinformatics*, *18*(1), 368. https://doi.org/10.1186/s12859-017-1776-8

Cruz Diaz, N. P., & Maña López, M. (2015). An Analysis of Biomedical Tokenization: Problems and Strategies. In *Proceedings of the Sixth International Workshop on Health Text Mining and Information Analysis* (pp. 40–49). Lisbon, Portugal: Association for Computational Linguistics. Retrieved from http://aclweb.org/anthology/W15-2605

Dang, T. H., Le, H.-Q., Nguyen, T. M., & Vu, S. T. (2018). D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information. *Bioinformatics*, *34*(20), 3539–3546. https://doi.org/10.1093/bioinformatics/bty356

de Matos, P., Alcántara, R., Dekker, A., Ennis, M., Hastings, J., Haug, K., … Steinbeck, C. (2010). Chemical Entities of Biological Interest: an update. *Nucleic Acids Research*, *38*(Database issue), D249-254. https://doi.org/10.1093/nar/gkp886

Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., … Ashburner, M. (2008). ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, *36*(Database issue), D344–D350. https://doi.org/10.1093/nar/gkm791

Dernoncourt, F., & Lee, J. Y. (n.d.). PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts, 6.

Ding, J., Berleant, D., Nettleton, D., & Wurtele, E. (2002). Mining MEDLINE: abstracts, sentences, or phrases? *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 326–337.

Djoumbou Feunang, Y., Eisner, R., Knox, C., Chepelev, L., Hastings, J., Owen, G., … Wishart, D. S. (2016). ClassyFire: automated chemical classification with a comprehensive, computable taxonomy. *Journal of Cheminformatics*, *8*, 61–61. https://doi.org/10.1186/s13321-016-0174-y

Doayan, R. I., Leaman, R., & Lu, Z. (2014). NCBI disease corpus: a resource for disease name recognition and concept normalization. *J Biomed Inform*, *47*, 1–10.

Doubova, S. V., Ferreira-Hermosillo, A., Perez-Cuevas, R., Barsoe, C., Gryzbowski-Gainza, E., & Valencia, J. E. (2018). Socio-demographic and clinical characteristics of type 1 diabetes patients associated with emergency room visits and hospitalizations in Mexico. *BMC Health Services Research*, *18*(1), 602. https://doi.org/10.1186/s12913-018-3412-3

Drakos, G. (2018, August 12). Support Vector Machine vs Logistic Regression. Retrieved January 10, 2019, from https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f

Elliott, J. H., Turner, T., Clavisi, O., Thomas, J., Higgins, J. P. T., Mavergames, C., & Gruen, R. L. (2014). Living Systematic Reviews: An Emerging Opportunity to Narrow the

Evidence-Practice Gap. *PLoS Medicine*, *11*(2). https://doi.org/10.1371/journal.pmed.1001603

Figueroa, R. L., Zeng-Treitler, Q., Kandula, S., & Ngo, L. H. (2012). Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, *12*, 8. https://doi.org/10.1186/1472-6947-12-8

Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 363–370). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.3115/1219840.1219885

Forbes, S. A., Beare, D., Boutselakis, H., Bamford, S., Bindal, N., Tate, J., … Campbell, P. J. (2017). COSMIC: somatic cancer genetics at high-resolution. *Nucleic Acids Research*, *45*(D1), D777–D783. https://doi.org/10.1093/nar/gkw1121

Friedrich, C. M., Revillion, T., Hofmann, M., & Fluck, J. (n.d.). Biomedical and Chemical Named Entity Recognition with Conditional Random Fields: The Advantage of Dictionary Features, 4.

Fundel, K., Küffner, R., & Zimmer, R. (2007). RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, *23*(3), 365–371. https://doi.org/10.1093/bioinformatics/btl616

Furlong, L. I., Dach, H., Hofmann-Apitius, M., & Sanz, F. (2008). OSIRISv1.2: a named entity recognition system for sequence variants of genes in biomedical literature. *BMC Bioinformatics*, *9*, 84. https://doi.org/10.1186/1471-2105-9-84

Gal, Y., & Ghahramani, Z. (2015). A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *ArXiv:1512.05287 [Stat]*. Retrieved from http://arxiv.org/abs/1512.05287

Galea, D. (2015). *Translational bioinformatics for bacterial identification using rapid evaporative ionization mass spectrometry*. Imperial College London, London.

Galea, D., Inglese, P., Cammack, L., Strittmatter, N., Rebec, M., Mirnezami, R., … Veselkov, K. A. (2017). Translational utility of a hierarchical classification strategy in biomolecular data analytics. *Scientific Reports*, *7*(1), 14981. https://doi.org/10.1038/s41598-017-14092-7

Galea, D., Laponogov, I., & Veselkov, K. (2018a). Data-driven visualizations in metabolic phenotyping. In *The Handbook of Metabolic Phenotyping* (pp. 309–326). Elsevier.

Galea, D., Laponogov, I., & Veselkov, K. (2018b). Exploiting and assessing multi-source data for supervised biomedical named entity recognition. *Bioinformatics*, *34*(14), 2474–2482. https://doi.org/10.1093/bioinformatics/bty152

Galea, D., Laponogov, I., & Veselkov, K. (2018c). Sub-word information in pre-trained biomedical word representations: evaluation and hyper-parameter optimization. In *Proceedings of the BioNLP 2018 workshop* (pp. 56–66). Melbourne, Australia: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/W18-2307

Ganesan, K., Kulandaisamy, A., Binny Priya, S., & Michael Gromiha, M. HuVarBase: A human variant database with comprehensive information at gene and protein levels. *PLoS One*. 14(*1*):e0210475. doi:10.1371/journal.pone/0210475

Gerner, M., Nenadic, G., & Bergman, C. M. (2009). LINNAEUS: A species name identification system for biomedical literature. In *BMC Bioinformatics*.

Gerner, M., Nenadic, G., & Bergman, C. M. (2010). An Exploration of Mining Gene Expression Mentions and Their Anatomical Locations from Biomedical Text. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing* (pp.

72–80). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1869961.1869970

Gonzalez Pigorini, G. (2018). *Machine-learning drug repositioning using large-scale propagated genomic data*. Imperial College London.

Gooch, P. (2018). *Python3 implementation of the Schwartz-Hearst algorithm for extracting abbreviation-definition pairs: philgooch/abbreviation-extraction*. Python. Retrieved from https://github.com/philgooch/abbreviation-extraction (Original work published 2017)

Gridach, M. (2017). Character-level neural network for biomedical named entity recognition. *Journal of Biomedical Informatics*, *70*, 85–91. https://doi.org/10.1016/j.jbi.2017.05.002

GuoDong, Z., & Jian, S. (2004). Exploring Deep Knowledge Resources in Biomedical Name Recognition. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications* (pp. 96–99). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1567594.1567616

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., & Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics (Oxford, England)*, *33*(14), i37–i48. https://doi.org/10.1093/bioinformatics/btx228

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th Python in Science Conference* (pp. 11–15). Pasadena, CA USA.

Hamosh, A., Scott, A. F., Amberger, J. S., Bocchini, C. A., & McKusick, V. A. (2005). Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic

disorders. *Nucleic Acids Research*, *33*(Database issue), D514-517. https://doi.org/10.1093/nar/gki033

Hashim Mohammed, B., & Omar, N. (2016). A back propagation neural network for identifying multi-word biomedical named entities. *International Review on Computers and Software*, *11*(8), 682–690. https://doi.org/10.15866/irecos.v11i8.9650

Haury, A.-C., Gestraud, P., & Vert, J.-P. (2011). The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures. *PLOS ONE*, *6*(12), e28210. https://doi.org/10.1371/journal.pone.0028210

He, J., & Chen, C. (2018). Predictive Effects of Novelty Measured by Temporal Embeddings on the Growth of Scientific Literature. *Frontiers in Research Metrics and Analytics*, *3*. https://doi.org/10.3389/frma.2018.00009

Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., & Declerck, T. (2013). The DDI corpus: an annotated corpus with pharmacological substances and drug-drug interactions. *Journal of Biomedical Informatics*, *46*(5), 914–920. https://doi.org/10.1016/j.jbi.2013.07.011

Hoffmann, R., & Valencia, A. (2004). A gene network for navigating the literature. *Nature Genetics*, *36*(7), 664. https://doi.org/10.1038/ng0704-664

Hsu, C.-N., Chang, Y.-M., Kuo, C.-J., Lin, Y.-S., Huang, H.-S., & Chung, I.-F. (2008). Integrating high dimensional bi-directional parsing models for gene mention tagging. *Bioinformatics (Oxford, England)*, *24*(13), i286-294. https://doi.org/10.1093/bioinformatics/btn183

Humphreys, B. L., Lindberg, D. A., Schoolman, H. M., Barnett, G. O. (1998). The unified medical language system: an information research collaboration, *J. Am. Med. Inform. Assoc*. 5 pp 1-13.

Huttlin, E. L., Ting, L., Bruckner, R. J., Gebreab, F., Gygi, M. P., Szpyt, J., … Gygi, S. P. (2015). The BioPlex Network: A Systematic Exploration of the Human Interactome. *Cell*, *162*(2), 425–440. https://doi.org/10.1016/j.cell.2015.06.043

Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *ICWSM*.

ICZN. (1999). In *International Code of Zoological Nomenclature* (4th ed., p. 306). London, UK: The International Trust for Zoological Nomenclature.

Jewison, T., Knox, C., Neveu, V., Djoumbou, Y., Guo, A. C., Lee, J., … Wishart, D. S. (2012). YMDB: the Yeast Metabolome Database. *Nucleic Acids Research*, *40*(Database issue), D815–D820. https://doi.org/10.1093/nar/gkr916

Jimeno Yepes, A. M. N., & Verspoor, K. (2013). Brat2BioC: conversion tool between brat and BioC. In *Proceedings of the BioCreative IV Workshop* (Vol. 1, pp. 46–53). Bethesda, MD. Retrieved from http://www.biocreative.org/resources/publications/biocreative-iv-proceedings/

Jimeno Yepes, A., & Verspoor, K. (2014). Mutation extraction tools can be combined for robust recognition of genetic variants in the literature. *F1000Research*. https://doi.org/10.12688/f1000research.3-18.v2

Kanehisa, M. & Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*. 28, 27-30. doi:https://doi.org/10.1093/nar/28.1.27

Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., & Morishima, K. (2017). KEGG: new perspective on genomes, pathways, diseases and drugs. *Nucleic Acids Res*. 45(D1). doi:https://doi.org/10.1093/nar/gkw1092

Keseler, I. M. et al. (2017). The EcoCyc database: reflecting new knowledge about Escherichia coli K-12. *Nucleic Acids Res*. 45, D543-D550. doi:https://doi.org/10.1093/nar/gkw1003

Kim, J.-D., Ohta, T., Tateisi, Y., & Tsujii, J. (2003). GENIA corpus--semantically annotated corpus for bio-textmining. *Bioinformatics (Oxford, England)*, *19 Suppl 1*, i180-182.

Kim, Jin-Dong, Ohta, T., Tsuruoka, Y., Tateisi, Y., & Collier, N. (2004). Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications - JNLPBA '04* (p. 70). Geneva, Switzerland: Association for Computational Linguistics. https://doi.org/10.3115/1567594.1567610

Kim, S., Yoon, J., & Yang, J. (2008). Kernel approaches for genic interaction extraction. *Bioinformatics*, *24*(1), 118–126. https://doi.org/10.1093/bioinformatics/btm544

Kim, S., Yoon, J., Yang, J., & Park, S. (2010). Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC Bioinformatics*, *11*(1), 107. https://doi.org/10.1186/1471-2105-11-107

Klinger, R., Kolářik, C., Fluck, J., Hofmann-Apitius, M., & Friedrich, C. M. (2008). Detection of IUPAC and IUPAC-like chemical names. *Bioinformatics*, *24*(13), i268–i276. https://doi.org/10.1093/bioinformatics/btn181

Koki, A. T., & Masferrer, J. L. (2002). Celecoxib: a specific COX-2 inhibitor with anticancer properties. *Cancer Control: Journal of the Moffitt Cancer Center*, *9*(2 Suppl), 28–35. https://doi.org/10.1177/107327480200902S04

Kosmopoulos, A., Androutsopoulos, I., & Paliouras, G. (n.d.). Biomedical Semantic Indexing using Dense Word Vectors in BioASQ, 20.

Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., & Valencia, A. (2015). CHEMDNER: The drugs and chemical names extraction challenge. *Journal of Cheminformatics*, *7*(1), S1. https://doi.org/10.1186/1758-2946-7-S1-S1

Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., … Valencia, A. (2015). The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, *7*(1), S2. https://doi.org/10.1186/1758-2946-7-S1-S2

Krishnan, V., & Ganapathy, V. (2005). *Named Entity Recognition*.

Kuhn, M., von Mering, C., Campillos, M., Jensen, L. J., & Bork, P. (2008). STITCH: interaction networks of chemicals and proteins. *Nucleic Acids Research*, *36*(Database issue), D684–D688. https://doi.org/10.1093/nar/gkm795

Kumar, A., Roberts, D., Wood, K. E., Light, B., Parrillo, J. E., Sharma, S., … Cheang, M. (2006). Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock. *Critical Care Medicine*, *34*(6), 1589–1596. https://doi.org/10.1097/01.CCM.0000217961.75225.E9

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 260–270). San Diego, California: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/N16-1030

Lamurias, A., Grego, T., & Couto, F. M. (2013). Chemical compound and drug name recognition using CRFs and semantic similarity based on ChEBI.

Landrum, M. J., Lee, J. M., Riley, G. R., Jang, W., Rubinstein, W. S., Church, D. M., & Maglott, D. R. (2014). ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Research*, *42*(Database issue), D980-985. https://doi.org/10.1093/nar/gkt1113

Lee, J. Y., & Dernoncourt, F. (2016). Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. *ArXiv:1603.03827 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1603.03827

Lee, K.-J., Hwang, Y.-S., Kim, S., & Rim, H.-C. (2004). Biomedical named entity recognition using two-phase model based on SVMs. *Journal of Biomedical Informatics*, *37*(6), 436–447. https://doi.org/10.1016/j.jbi.2004.08.012

Lee, S. Y. (2016). Temozolomide resistance in glioblastoma multiforme. *Genes & Diseases*, *3*(3), 198–210. https://doi.org/10.1016/j.gendis.2016.04.007

Lesk, M. (1986). Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation* (pp. 24–26). New York, NY, USA: ACM. https://doi.org/10.1145/318723.318728

Li, Haifeng, Jiang, T., & Zhang, K. (2006). Efficient and robust feature extraction by maximum margin criterion. *IEEE Transactions on Neural Networks*, *17*(1), 157–165. https://doi.org/10.1109/TNN.2005.860852

Li, Hao, & Lu, W. (2018). Learning with Structured Representations for Negation Scope Extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 533–539). Melbourne, Australia: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/P18-2085

Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C.-H., Leaman, R., … Lu, Z. (2016). BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database: The Journal of Biological Databases and Curation*, *2016*. https://doi.org/10.1093/database/baw068

Lim, E., Pon, A., Djoumbou, Y., Knox, C., Shrivastava, S., Guo, A. C., … Wishart, D. S. (2010). T3DB: a comprehensively annotated database of common toxins and their targets. *Nucleic Acids Research*, *38*(Database issue), D781–D786. https://doi.org/10.1093/nar/gkp934

Lin, D. (1998). An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 296–304). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=645527.657297

Liu, L., Shang, J., Xu, F. F., Ren, X., Gui, H., Peng, J., & Han, J. (2017). Empower Sequence Labeling with Task-Aware Neural Language Model. *CoRR*, *abs/1709.04109*. Retrieved from http://arxiv.org/abs/1709.04109

Liu, Y., Liang, Y., & Wishart, D. (2015). PolySearch2: a significantly improved text-mining system for discovering associations between human diseases, genes, drugs, metabolites, toxins and more. *Nucleic Acids Research*, *43*(W1), W535-542. https://doi.org/10.1093/nar/gkv383

Lou, Y., Zhang, Y., Qian, T., Li, F., Xiong, S., & Ji, D. (2017). A transition-based joint model for disease named entity recognition and normalization. *Bioinformatics*, *33*(15), 2363–2371. https://doi.org/10.1093/bioinformatics/btx172

Lowe, D. M., O'Boyle, N. M., & Sayle, R. A. (2016). Efficient chemical-disease identification and relationship extraction using Wikipedia to improve recall. *Database: The Journal of Biological Databases and Curation*, *2016*. https://doi.org/10.1093/database/baw039

Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., … Wren, J. (2018). An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics*, *34*(8), 1381–1388. https://doi.org/10.1093/bioinformatics/btx761

Ma, X., & Hovy, E. H. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. *CoRR*, *abs/1603.01354*. Retrieved from http://arxiv.org/abs/1603.01354

Maglott, D., Ostell, J., Pruitt, K. D., & Tatusova, T. (2005). Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, *33*(Database issue), D54–D58. https://doi.org/10.1093/nar/gki031

Marques, H., & Rinaldi, F. (2013). PyBioC: a python implementation of the BioC core. In *Fourth BioCreative Challenge Evaluation Workshop* (Vol. 1, pp. 2–4). Biocreative. Retrieved from https://doi.org/10.5167/uzh-91881

Material for MkDocs. (n.d.). Retrieved November 10, 2018, from https://squidfunk.github.io/mkdocs-material/

Matthews, H., Hanison, J., & Nirmalan, N. (2016). "Omics"-Informed Drug and Biomarker Discovery: Opportunities, Challenges and Future Perspectives. *Proteomes*, *4*, 28. https://doi.org/10.3390/proteomes4030028

McCallum, A. K. (2002). *MALLET: A Machine Learning for Language Toolkit*.

McShane, L. (2017). In Pursuit of Greater Reproducibility and Credibility of Early Clinical Biomarker Research. *Clinical and Translational Science*, *10*(2), 58–60. https://doi.org/10.1111/cts.12449

Medicines | European Medicines Agency. (n.d.). Retrieved November 10, 2018, from https://www.ema.europa.eu/en/medicines

MEDLINE, PubMed, and PMC (PubMed Central): How are they different? (n.d.). [FAQs, Help Files, Pocket Cards]. Retrieved December 19, 2018, from https://www.nlm.nih.gov/bsd/difference.html

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in Pre-Training Distributed Word Representations. *ArXiv:1712.09405 [Cs]*. Retrieved from http://arxiv.org/abs/1712.09405

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 3111–3119). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Miwa, M., Saetre, R., Miyao, Y., & Tsujii, J. (2009). A Rich Feature Vector for Protein-Protein Interaction Extraction from Multiple Corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (pp. 121–130). Singapore: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/D/D09/D09-1013

Miyao, Y., Sagae, K., Saetre, R., Matsuzaki, T., & Tsujii, J. (2009). Evaluating contributions of natural language parsers to protein-protein interaction extraction. *Bioinformatics (Oxford, England)*, *25*(3), 394–400. https://doi.org/10.1093/bioinformatics/btn631

MkDocs. (n.d.). Retrieved November 10, 2018, from https://www.mkdocs.org/

Müller, H.-M., Van Auken, K. M., Li, Y., & Sternberg, P. W. (2018). Textpresso Central: a customizable platform for searching, text mining, viewing, and curating biomedical literature. *BMC Bioinformatics*, *19*(1), 94. https://doi.org/10.1186/s12859-018-2103-8

Mulligen, E. M. van, Fourrier-Reglat, A., Gurwitz, D., Molokhia, M., Nieto, A., Trifiro, G., … Furlong, L. I. (2012). The EU-ADR corpus: Annotated drugs, diseases, targets, and their relationships. *Journal of Biomedical Informatics*, *45*(5), 879–884. https://doi.org/10.1016/j.jbi.2012.04.004

Mullin, R. (n.d.). Cost to Develop New Pharmaceutical Drug Now Exceeds $2.5B. Retrieved January 7, 2019, from https://www.scientificamerican.com/article/cost-to-develop-new-pharmaceutical-drug-now-exceeds-2-5b/

Neves, M., Damaschun, A., Kurtz, A., & Leser, U. (2012). Annotating and Evaluating Text for Stem Cell Research.

Newman-Griffis, D., Lai, A. M., & Fosler-Lussier, E. (2017). Insights into Analogy Completion from the Biomedical Domain. *ArXiv:1706.02241 [Cs]*. Retrieved from http://arxiv.org/abs/1706.02241

Ohta, T., Pyysalo, S., Tsujii, J., & Ananiadou, S. (2012). Open-domain Anatomical Entity Mention Detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse* (pp. 27–36). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=2391171.2391177

Pakhomov, S. V. S., Finley, G., McEwan, R., Wang, Y., & Melton, G. B. (2016). Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, *32*(23), 3635–3644. https://doi.org/10.1093/bioinformatics/btw529

Panyam, N. C., Verspoor, K., Cohn, T., & Ramamohanarao, K. (2018). Exploiting graph kernels for high performance biomedical relation extraction. *Journal of Biomedical Semantics*, *9*(1), 7. https://doi.org/10.1186/s13326-017-0168-3

Patwardhan, S. (2006). Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *In: Proceedings of the EACL* (pp. 1–8).

Pedersen, T., Pakhomov, S., McInnes, B., & Liu, Y. (n.d.). Measuring the Similarity and Relatedness of Concepts in the Medical Domain : IHI 2012 Tutorial, 210.

Peng, N., Poon, H., Quirk, C., Toutanova, K., & Yih, W. (2017). *Cross-Sentence N-ary Relation Extraction with Graph LSTMs*.

Peng, Y., Gupta, S., Wu, C., & Shanker, V. (2015). An extended dependency graph for relation extraction in biomedical texts. In *Proceedings of BioNLP 15* (pp. 21–30). Beijing,

China: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/W15-3803

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology/D14-1162

Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. *CoRR*, *abs/1705.00108*. Retrieved from http://arxiv.org/abs/1705.00108

Piñero, J., Bravo, À., Queralt-Rosinach, N., Gutiérrez-Sacristán, A., Deu-Pons, J., Centeno, E., … Furlong, L. I. (2017). DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, *45*(D1), D833–D839. https://doi.org/10.1093/nar/gkw943

Piñero, J., Queralt-Rosinach, N., Bravo, À., Deu-Pons, J., Bauer-Mehren, A., Baron, M., … Furlong, L. I. (2015). DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database: The Journal of Biological Databases and Curation*, *2015*, bav028. https://doi.org/10.1093/database/bav028

Pinter, Y., Guthrie, R., & Eisenstein, J. (2017). Mimicking Word Embeddings using Subword RNNs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 102–112). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D17-1010

Pletscher-Frankild, S., Pallejà, A., Tsafou, K., Binder, J. X., & Jensen, L. J. (2015). DISEASES: Text mining and data integration of disease–gene associations. *Methods*, *74*, 83–89. https://doi.org/10.1016/j.ymeth.2014.11.020

Ponomareva, N., Rosso, P., Pla, F., & Molina, A. (n.d.). *Conditional Random Fields vs. Hidden Markov Models in a biomedical Named Entity Recognition task*.

Poynter, L., Galea, D., Veselkov, K., Mirnezami, A., Kinross, J., Takats, Z., … Mirnezami, R. (2019). Network mapping of molecular biomarkers influencing radiation response in rectal cancer.

Pubchem. (n.d.). 3-Methyl-L-histidine. Retrieved December 7, 2018, from https://pubchem.ncbi.nlm.nih.gov/compound/64969

Pustejovsky, J., Castaño, J., Cochran, B., Kotecki, M., & Morrell, M. (2001). Automatic extraction of acronym-meaning pairs from MEDLINE databases. *Studies in Health Technology and Informatics*, *84*(Pt 1), 371–375.

Pustejovsky, James, Castano, J., Cochran, B., Kotecki, M., Morrell, M., & Rumshisky, A. (2004). *Extraction and Disambiguation of Acronym-Meaning Pairs in Medline*.

Python library implementing a trie data structure.: google/pygtrie. (2018). Python, Google. Retrieved from https://github.com/google/pygtrie (Original work published 2014)

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., & Ananiadou, S. (2013). Distributional Semantics Resources for Biomedical Text Processing. In *Proceedings of LBM 2013* (pp. 39–44). Retrieved from http://lbm2013.biopathway.org/lbm2013proceedings.pdf

Pyysalo, Sampo, & Ananiadou, S. (2014). Anatomical entity mention recognition at literature scale. *Bioinformatics (Oxford, England)*, *30*(6), 868–875. https://doi.org/10.1093/bioinformatics/btt580

Pyysalo, Sampo, Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., & Salakoski, T. (2007). BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, *8*, 50. https://doi.org/10.1186/1471-2105-8-50

Pyysalo, Sampo, Ohta, T., Miwa, M., Cho, H.-C., Tsujii, J., & Ananiadou, S. (2012). Event extraction across multiple levels of biological organization. *Bioinformatics*, *28*(18), i575–i581. https://doi.org/10.1093/bioinformatics/bts407

Pyysalo, Sampo, Ohta, T., Rak, R., Sullivan, D., Mao, C., Wang, C., … Ananiadou, S. (2012). Overview of the ID, EPI and REL tasks of BioNLP Shared Task 2011. *BMC Bioinformatics*, *13*(11), S2. https://doi.org/10.1186/1471-2105-13-S11-S2

Qiu, G., Lam, K. M., Kiya, H., Xue, X.-Y., Kuo, C.-C. J., & Lew, M. S. (Eds.). (2010). *Advances in Multimedia Information Processing -- PCM 2010, Part II: 11th Pacific Rim Conference on Multimedia, Shanghai, China, September 21-24, 2010 Proceedings*. Berlin Heidelberg: Springer-Verlag. Retrieved from //www.springer.com/gp/book/9783642156953

Radix tree - Swift Data Structure and Algorithms [Book]. (n.d.). Retrieved November 9, 2018, from https://www.oreilly.com/library/view/swift-data-structure/9781785884504/ch06s07.html

Raja, K., Subramani, S., & Natarajan, J. (2013). PPInterFinder—a mining tool for extracting causal relations on human proteins from literature. *Database: The Journal of Biological Databases and Curation*, *2013*. https://doi.org/10.1093/database/bas052

Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora (pp. 45–50). https://doi.org/10.13140/2.1.2393.1847

Rei, M., Crichton, G. K. O., & Pyysalo, S. (2016). Attending to Characters in Neural Sequence Labeling Models. *ArXiv:1611.04361 [Cs]*. Retrieved from http://arxiv.org/abs/1611.04361

Reimers, N., & Gurevych, I. (2017). Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. *ArXiv:1707.06799 [Cs]*. Retrieved from http://arxiv.org/abs/1707.06799

Rindflesch, T. C., Tanabe, L., Weinstein, J. N., & Hunter, L. EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature. *Pac Symp Biocomput*. 517-528.

Rios, A., Kavuluru, R., & Lu, Z. (2018). Generalizing biomedical relation classification with neural adversarial domain adaptation. *Bioinformatics*, *34*(17), 2973–2981. https://doi.org/10.1093/bioinformatics/bty190

Robert, V., Vu, D., Amor, A. B. H., van de Wiele, N., Brouwer, C., Jabas, B., … Crous, P. W. (2013). MycoBank gearing up for new horizons. *IMA Fungus*, *4*(2), 371–379. https://doi.org/10.5598/imafungus.2013.04.02.16

Rocktäschel, T., Weidlich, M., & Leser, U. (2012). ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics*, *28*(12), 1633–1640. https://doi.org/10.1093/bioinformatics/bts183

ROGERS, F. B. (1963). Medical subject headings. *Bulletin of the Medical Library Association*, *51*(1), 114–116.

Roskov, Y., Kunze, T., Paglinawan, L., Orrell, T., Nicolson, D., Culham, A., … Wever, A. D. (2013). *Species 2000 & ITIS Catalogue of Life, 2013 Annual Checklist* (Technical Report). Reading: University of Reading. Retrieved from http://centaur.reading.ac.uk/34322/

Sajed, T., Marcu, A., Ramirez, M., Pon, A., Guo, A. C., Knox, C., … Wishart, D. S. (2016). ECMDB 2.0: A richer resource for understanding the biochemistry of E. coli. *Nucleic Acids Research*, *44*(Database issue), D495–D501. https://doi.org/10.1093/nar/gkv1060

Saric, J., Jensen, L. J., Ouzounova, R., Rojas, I., & Bork, P. (2006). Extraction of regulatory gene/protein networks from Medline. *Bioinformatics (Oxford, England)*, *22*(6), 645–650. https://doi.org/10.1093/bioinformatics/bti597

Sayad, S. (2019). Logistic Regression. Retrieved January 11, 2019, from https://www.saedsayad.com/logistic_regression.htm

Scalbert, A., Andres-Lacueva, C., Arita, M., Kroon, P., Manach, C., Urpi-Sarda, M., & Wishart, D. (2011). Databases on Food Phytochemicals and Their Health-Promoting Effects. *Journal of Agricultural and Food Chemistry*, *59*(9), 4331–4348. https://doi.org/10.1021/jf200591d

Scherer, A. (2017). Reproducibility in biomarker research and clinical development: a global challenge. *Biomarkers in Medicine*, *11*(4), 309–312. https://doi.org/10.2217/bmm-2017-0024

Schriml, L. M., Arze, C., Nadendla, S., Chang, Y.-W. W., Mazaitis, M., Felix, V., … Kibbe, W. A. (2012). Disease Ontology: a backbone for disease semantic integration. *Nucleic Acids Research*, *40*(D1), D940–D946. https://doi.org/10.1093/nar/gkr972

Schwartz, A. S., & Hearst, M. A. (2003). A simple algorithm for identifying abbreviation definitions in biomedical text. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 451–462.

Segerdell, E., Bowes, J. B., Pollet, N., & Vize, P. D. (2008). An ontology for Xenopus anatomy and development. *BMC Developmental Biology*, *8*(1), 92. https://doi.org/10.1186/1471-213X-8-92

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA* (pp. 104–107).

Settles, B. (2005). ABNER: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, *21*(14), 3191–3192. https://doi.org/10.1093/bioinformatics/bti475

Shahab, E. (2017). A Short Survey of Biomedical Relation Extraction Techniques. *ArXiv:1707.05850 [Cs]*. Retrieved from http://arxiv.org/abs/1707.05850

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., … Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, *13*(11), 2498–2504. https://doi.org/10.1101/gr.1239303

SIMPLS: An alternative approach to partial least squares regression - ScienceDirect. (n.d.). Retrieved January 4, 2019, from https://www.sciencedirect.com/science/article/abs/pii/016974399385002X

Singh, V. (2017). Replace or Retrieve Keywords In Documents at Scale. *ArXiv E-Prints*.

Smalheiser, N. R. (2017). Rediscovering Don Swanson: The past, present and future of literature-based discovery. *JDIS,* 2(4), pp 43-64. doi:10.1515/jdis-2017-0019

Smalheiser, N. R., Torvik, V. I., & Zhou, W. (2009). Arrowsmith two-node search interface: A tutorial on finding meaningful links between two disparate sets of articles in MEDLINE. *Computer Methods and Programs in Biomedicine*, *94*(2), 190–197. https://doi.org/10.1016/j.cmpb.2008.12.006

Smith, L., Tanabe, L. K., Ando, R. J. nee, Kuo, C.-J., Chung, I.-F., Hsu, C.-N., … Wilbur, W. J. (2008). Overview of BioCreative II gene mention recognition. *Genome Biology*, *9*(2), S2. https://doi.org/10.1186/gb-2008-9-s2-s2

Song, B., Shu, Z.-B., Du, J., Ren, J.-C., & Feng, Y. (2017). Anti-cancer effect of low dose of celecoxib may be associated with lnc-SCD-1:13 and lnc-PTMS-1:3 but not COX-2 in

NCI-N87 cells. *Oncology Letters*, *14*(2), 1775–1779. https://doi.org/10.3892/ol.2017.6316

Song, H.-J., Jo, B.-C., Park, C.-Y., Kim, J.-D., & Kim, Y.-S. (2018). Comparison of named entity recognition methodologies in biomedical documents. *BioMedical Engineering OnLine*, *17*(Suppl 2). https://doi.org/10.1186/s12938-018-0573-6

Stark, C., Breitkreutz, B. J., Reguly, T., Boucher, L., Breitkreutz, A. & Tyers, M. (2006). BioGRID: a general repository for interaction datasets. *Nucleic Acids Res*. D535-539. doi:10.1093/nar/gkj109

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., … Robinson, G. E. (2015). Big Data: Astronomical or Genomical? *PLOS Biology*, *13*(7), e1002195. https://doi.org/10.1371/journal.pbio.1002195

Sutton, C., & McCallum, A. (2012). An Introduction to Conditional Random Fields. *Found. Trends Mach. Learn.*, *4*(4), 267–373. https://doi.org/10.1561/2200000013

Swanson, D. R. (1986). Undiscovered public knowledge. *The Library Quarterly: Information, Community, Policy.* 56(2), pp 103-118.

Szklarczyk, D., Morris, J. H., Cook, H., Kuhn, M., Wyder, S., Simonovic, M., … von Mering, C. (2017). The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Research*, *45*(Database issue), D362–D368. https://doi.org/10.1093/nar/gkw937

Thomas, P. E., Klinger, R., Furlong, L. I., Hofmann-Apitius, M., & Friedrich, C. M. (2011). Challenges in the association of human single nucleotide polymorphism mentions with unique database identifiers. *BMC Bioinformatics*, *12*(4), S4. https://doi.org/10.1186/1471-2105-12-S4-S4

Thompson, P., Iqbal, S. A., McNaught, J., & Ananiadou, S. (2009). Construction of an annotated corpus to support biomedical information extraction. *BMC Bioinformatics*, *10*, 349. https://doi.org/10.1186/1471-2105-10-349

Tikk, D., Thomas, P., Palaga, P., Hakenberg, J., & Leser, U. (2010). A Comprehensive Benchmark of Kernel Methods to Extract Protein–Protein Interactions from Literature. *PLOS Computational Biology*, *6*(7), e1000837. https://doi.org/10.1371/journal.pcbi.1000837

Tsai, R. T.-H., Sung, C.-L., Dai, H.-J., Hung, H.-C., Sung, T.-Y., & Hsu, W.-L. (2006). NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition. *BMC Bioinformatics*, *7 Suppl 5*, S11. https://doi.org/10.1186/1471-2105-7-S5-S11

Turland, N. J., Wiersema, J. H., Barrie, F. R., Greuter, W., Hawksworth, D. L., Herendeen, P. S., … Smith, G. F. (2018). *International Code of Nomenclature for algae, fungi, and plants (Shenzhen Code)*. Glashütten: Koeltz Botanical Books.

Understanding LSTM Networks -- colah's blog. (n.d.). Retrieved November 10, 2018, from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Ursu, O., Holmes, J., Knockel, J., Bologa, C. G., Yang, J. J., Mathias, S. L., … Oprea, T. I. (2017). DrugCentral: online drug compendium. *Nucleic Acids Research*, *45*(Database issue), D932–D939. https://doi.org/10.1093/nar/gkw993

Usié, A., Cruz, J., Comas, J., Solsona, F., & Alves, R. (2015). CheNER: a tool for the identification of chemical entities and their classes in biomedical literature. *Journal of Cheminformatics*, *7*(Suppl 1 Text mining for chemistry and the CHEMDNER track), S15. https://doi.org/10.1186/1758-2946-7-S1-S15

Veselkov, K., Gonzalez Pigorini, G., Aljifri, S., Galea, D., Mirnezami, R., Youssef, J., … Laponogov, I. (2018). HyperFoods: Machine intelligent mapping of cancer-beating molecules in foods.

Vincze, V., Szarvas, G., Farkas, R., Móra, G., & Csirik, J. (2008). The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC Bioinformatics*, *9*(Suppl 11), S9. https://doi.org/10.1186/1471-2105-9-S11-S9

Wei, C.-H., Phan, L., Feltz, J., Maiti, R., Hefferon, T., & Lu, Z. (2018). tmVar 2.0: integrating genomic variant information from literature with dbSNP and ClinVar for precision medicine. *Bioinformatics (Oxford, England)*, *34*(1), 80–87. https://doi.org/10.1093/bioinformatics/btx541

Winfield, L. L., & Payton-Stewart, F. (2012). Celecoxib and Bcl-2: emerging possibilities for anticancer drug design. *Future Medicinal Chemistry*, *4*(3), 361–383. https://doi.org/10.4155/fmc.11.177

Wishart, D. S., Feunang, Y. D., Marcu, A., Guo, A. C., Liang, K., Vázquez-Fresno, R., … Scalbert, A. (2018). HMDB 4.0: the human metabolome database for 2018. *Nucleic Acids Research*, *46*(D1), D608–D617. https://doi.org/10.1093/nar/gkx1089

Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., … Hassanali, M. (2008). DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Research*, *36*(Database issue), D901-906. https://doi.org/10.1093/nar/gkm958

World Health Organization. (1992). *The ICD-10 classification of mental and behavioural disorders : clinical descriptions and diagnostic guidelines*. Geneva: Geneva : World Health Organization. Retrieved from http://apps.who.int/iris/handle/10665/37958

Wu, Z., & Palmer, M. (1994). Verbs Semantics and Lexical Selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics* (pp. 133–138).

Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.3115/981732.981751

Xu, D., Zhang, M., Xie, Y., Wang, F., Chen, M., Zhu, K. Q., & Wei, J. (2016). DTMiner: identification of potential disease targets through biomedical literature mining. *Bioinformatics (Oxford, England)*, *32*(23), 3619–3626. https://doi.org/10.1093/bioinformatics/btw503

Yang, Z., Salakhutdinov, R., & Cohen, W. W. (2017). Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks. *CoRR*, *abs/1703.06345*. Retrieved from http://arxiv.org/abs/1703.06345

Yao, L., Liu, H., Liu, Y., Li, X., & Waqas Anwar, M. (2015). Biomedical Named Entity Recognition based on Deep Neutral Network. *International Journal of Hybrid Information Technology*, *8*, 279–288. https://doi.org/10.14257/ijhit.2015.8.8.29

Yates, B., Braschi, B., Gray, K., Seal, R., Tweedie, S., Bruford, E. (2017). Genenames.org: the HGNC and VGNC resources in 2017. *Nucleic Acids Res*. 45(D1): D619-625.

Ye, Z.-X., & Ling, Z.-H. (2018). Hybrid semi-Markov CRF for Neural Sequence Labeling. *CoRR*, *abs/1805.03838*. Retrieved from http://arxiv.org/abs/1805.03838

Yeh, A., Morgan, A., Colosimo, M., & Hirschman, L. (2005). BioCreAtIvE task 1A: gene mention finding evaluation. *BMC Bioinformatics*, *6 Suppl 1*, S2. https://doi.org/10.1186/1471-2105-6-S1-S2

Yetisgen-Yildiz, M., & Pratt, W. (2006). Using statistical and knowledge-based approaches for literature-based discovery. *Journal of Biomedical Informatics*, *39*(6), 600–611. https://doi.org/10.1016/j.jbi.2005.11.010

Younesi, E., Toldo, L., Müller, B., Friedrich, C. M., Novac, N., Scheer, A., … Fluck, J. (2012). Mining biomarker information in biomedical literature. *BMC Medical Informatics and Decision Making*, *12*, 148. https://doi.org/10.1186/1472-6947-12-148

Yu, Z., Wallace, B. C., Johnson, T., & Cohen, T. (2017). Retrofitting Concept Vector Representations of Medical Concepts to Improve Estimates of Semantic Similarity and Relatedness. *ArXiv:1709.07357 [Cs]*. Retrieved from http://arxiv.org/abs/1709.07357

Zeng, D., Sun, C., Lin, L., Liu, B., Zeng, D., Sun, C., … Liu, B. (2017). LSTM-CRF for Drug-Named Entity Recognition. *Entropy*, *19*(6), 283. https://doi.org/10.3390/e19060283

Zhang, Yaoyun, Xu, J., Chen, H., Wang, J., Wu, Y., Prakasam, M., & Xu, H. (2016). Chemical named entity recognition in patents by domain knowledge and unsupervised feature learning. *Database: The Journal of Biological Databases and Curation*, *2016*. https://doi.org/10.1093/database/baw049

Zhang, Yijia, Lin, H., Yang, Z., & Li, Y. (2011). Neighborhood hash graph kernel for protein–protein interaction extraction. *Journal of Biomedical Informatics*, *44*(6), 1086–1092. https://doi.org/10.1016/j.jbi.2011.08.011

Zhang, Yijia, Lin, H., Yang, Z., Wang, J., Zhang, S., Sun, Y., & Yang, L. (2018). A hybrid model based on neural networks for biomedical relation extraction. *Journal of Biomedical Informatics*, *81*, 83–92. https://doi.org/10.1016/j.jbi.2018.03.011

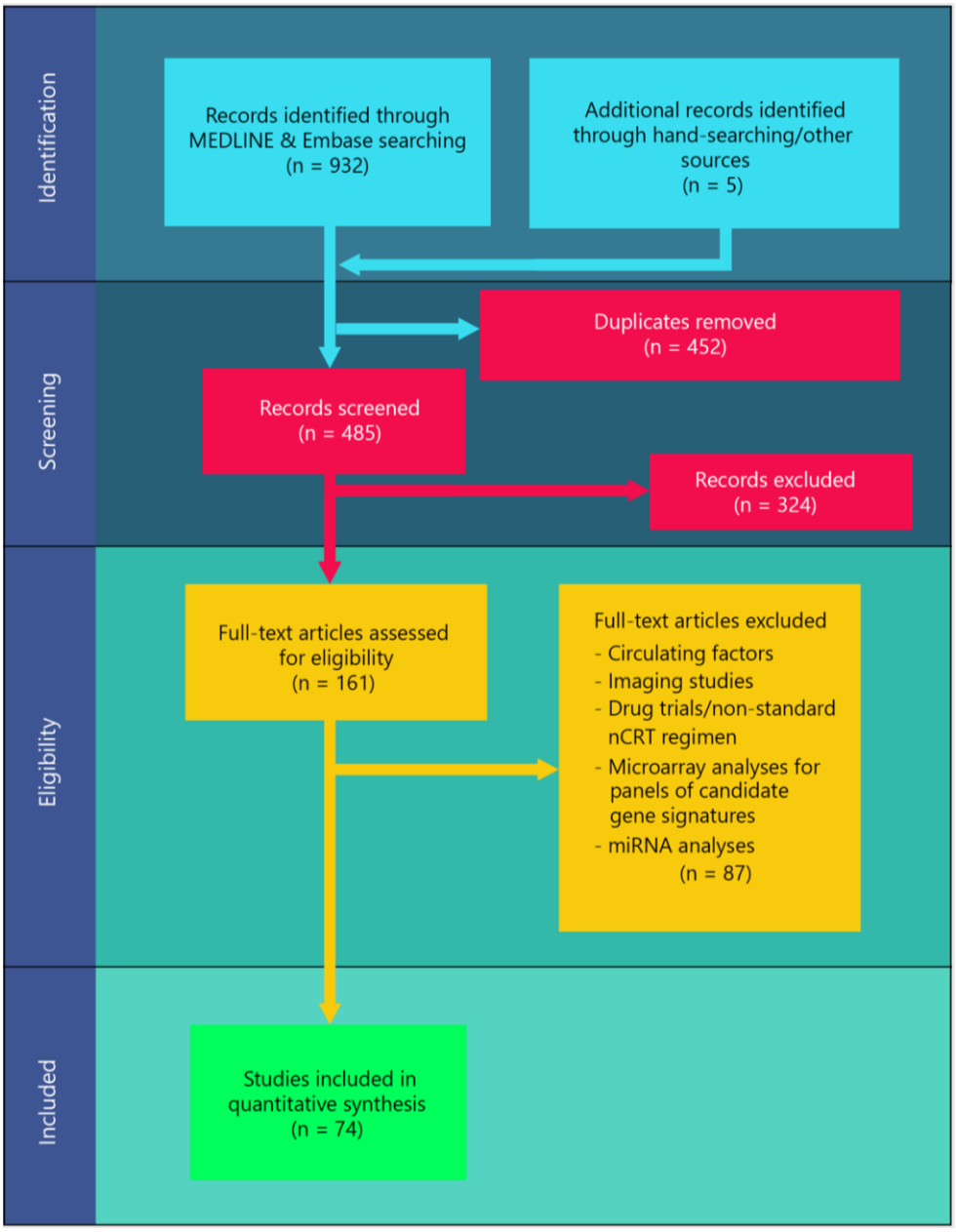
Zhao, Z., Yang, Z., Lin, H., Wang, J., & Gao, S. (2016). A protein-protein interaction extraction approach based on deep neural network. *International Journal of Data Mining and Bioinformatics*, *15*(2), 145–164. https://doi.org/10.1504/IJDMB.2016.076534

Zhou, G., & Su, J. (2002). Named Entity Recognition using an HMM-based Chunk Tagger. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics* (pp. 473–480). Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. https://doi.org/10.3115/1073083.1073163

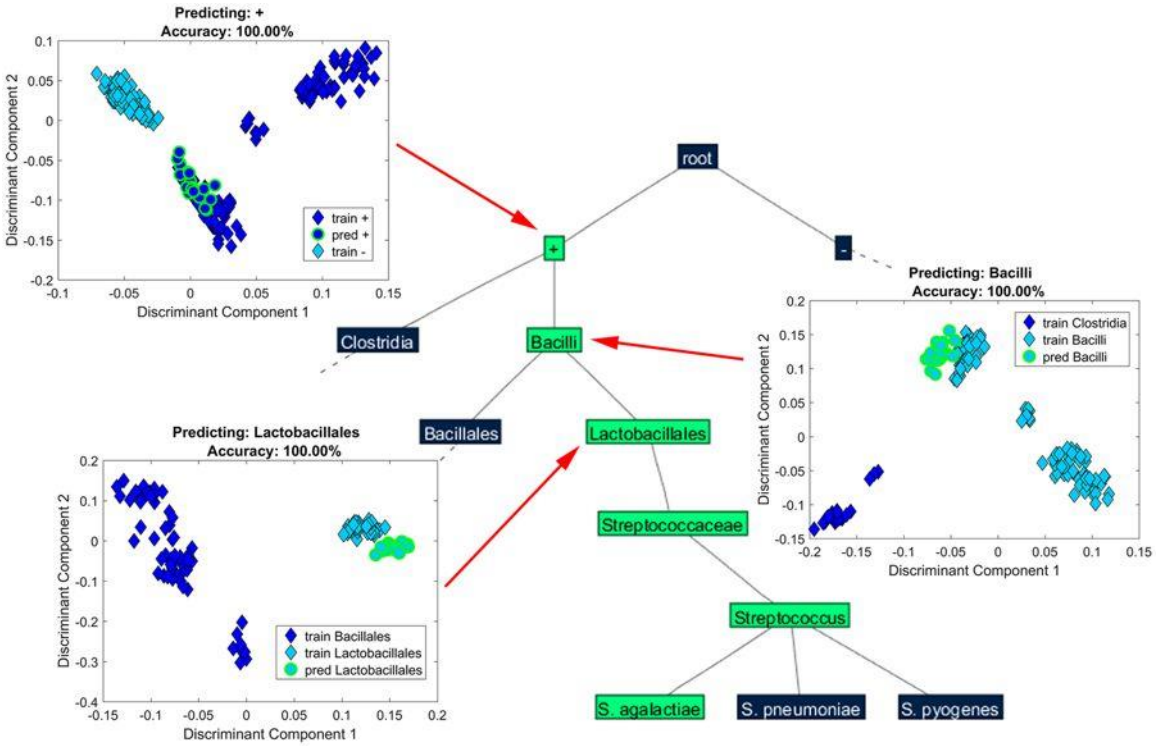
Zhou, G., Zhang, J., Su, J., Shen, D., & Tan, C. (2004). Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, *20*(7), 1178–1190. https://doi.org/10.1093/bioinformatics/bth060

Zhou, H., Deng, H., Chen, L., Yang, Y., Jia, C., & Huang, D. (2016). Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database*, *2016*. https://doi.org/10.1093/database/baw048

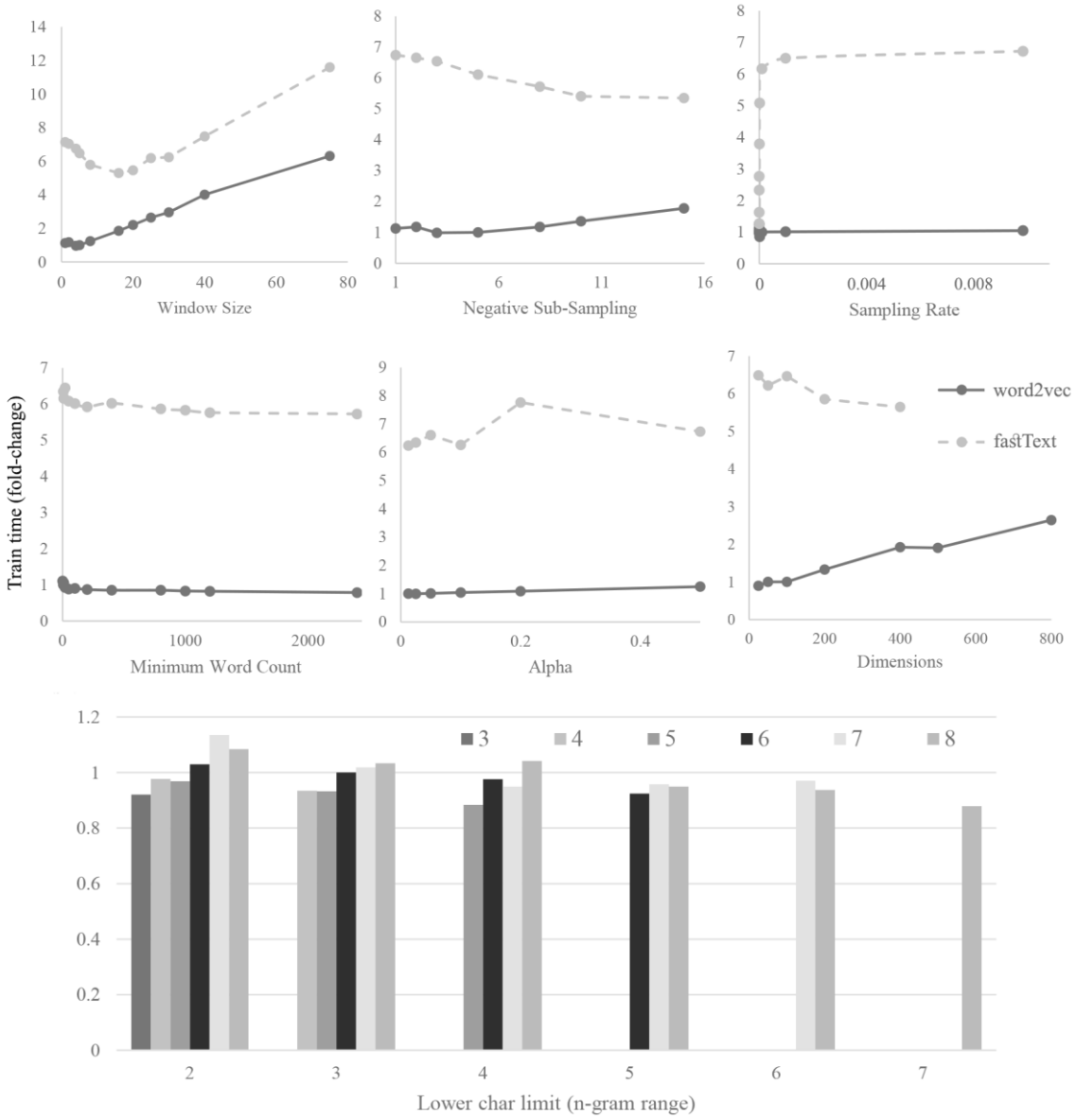
Supplementary Tables and Figures



Supplementary Figure 1. PRISMA flow diagram of the study filtering and selection process used to generate a corpus of studies that was in turn used to generate a systematic review of molecular biomarkers influencing radiation response in rectal cancer. Source: (Poynter et al., 2019)



Supplementary Figure 2. Output for the hierarchical leave-class-out prediction of bacterial species. Predicting hierarchical taxonomic classes (Gram positive, Bacilli, and Lactobacillales) for *Streptococcus agalactiae* using the algorithm developed in Galea et al. (2017). Source: (Galea et al., 2017).



Supplementary Figure 3. Word2vec and fastText training time benchmarks as a function of different values for the various hyperparameters: (i) window size; (ii) negative sub-sampling rate; (iii) sampling rate; (iv) word count; (v) alpha/learning rate; (vi) dimensions; and (vii) n-gram range. Y-axis units expressed in terms of fold-change to the default hyper-parameters. Source: (Galea et al., 2018c)

Supplementary Table 1. List of compiled biomedical corpora, the available formats and sources. Number of documents for each corpus may vary based on the source, and a document "unit" may be defined differently in different corpora (e.g. abstract, title, whole manuscript text). Source may be the original manuscript published, or if not available (or available in a different formation), other secondary sources hosting the resource. When a corpus is available in various formats and multiple sources, these are indicated. As links may become offline with time, a more dynamic and community-updated table is also hosted on https://github.com/dterg/biomedical_corpora. Source: Galea et al. (2018b).

Corpus	Year	Format	Documents	Original Paper	Downloaded From	Other URLs
Ab3P (Abbreviation Plus P-Precision)	2008	BioC	1250 PubMed Abstracts	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2576267/	http://bioc.sourceforge.net/	
AIMed	2005	BioC	~ 1000 MEDLINE abstracts (200 abstracts)	http://www.sciencedirect.com/science/article/pii/S0933365704001319	http://corpora.informatik.hu-berlin.de/	http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.3218&rep=rep1&type=pdf
AnatEM (Anatomical entity mention recognition)	2013	CONLL, standoff	1212 docs (500 docs from AnEM + 262 from MLEE + 450 others)	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3957068/	http://nactem.ac.uk/anatomytagger/#AnatEM	
AnEM	2012	BioC	500 docs (PubMed and PMC); abstracts and full text drawn randomly	http://www.nactem.ac.uk/anatomy/docs/ohta2012opendomain.pdf	http://corpora.informatik.hu-berlin.de/	
AZDC (Arizona Disease Corpus)	2009	IeXML, .txt	2856 PubMed abstracts (2775 sentences). Other source says 794 PubMed Abstracts	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2352871/	http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/IeXML/goldcorpus/azdc-1.xml	http://diego.asu.edu/downloads/AZDC 6-26-2009.txt
BEL (BioCreative V5 BEL Track)	2016	BioC		https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4995071/	https://wiki.openbel.org/display/BIOC/Datasets	
BioADI	2009	BioC	1201 PubMed abstracts	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2788358/	http://bioc.sourceforge.net/	
BioCause	2013	standoff	19 full-text documents	http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-2	http://www.nactem.ac.uk/biocause/download.php	
BioCreative-PPI		XML			https://www2.informatik.hu-berlin.de/~hakenberr/links/benchmarks.html	
BioGRID	2017	BioC	120 full text articles	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5225395/	http://bioc.sourceforge.net/BioC-BioGRID.html	
BioInfer	2007	BioC	1100 sentences from biomedical literature	http://www.biomedcentral.com/1471-2105/8/50	http://corpora.informatik.hu-berlin.de/	http://mars.cs.utu.fi/BioInfer

BioMedLat	2016	standoff	643 BioASQ questions/factoids	https://www.semanticscholar.org/paper/BioMedLAT-Corpus-Annotation-of-the-Lexical-Answer-Neves-Kraus/b0f09f94015771c31bd2483efdd8f0f86996384e	https://github.com/mariananeves/BioMedLAT	
BioText	2004	txt	100 titles and 40 abstracts	http://biotext.berkeley.edu/papers/acl04-relations.pdf	https://www2.informatik.hu-berlin.de/~hakenber/links/benchmarks.html	
CDR (BioCreative V)		BioC			http://bioc.sourceforge.net/	
CellFinder 1.0	2012	BioC	10 full documents from PMC from (Loser et al. 2009) on "Human Embryonic Stem Cell Lines and Their Use in International Research"	http://www.nactem.ac.uk/biotxtm2012/presentations/Neves-pres.pdf	http://corpora.informatik.hu-berlin.de/	http://cellfinder.de/about/annotation/
CG Cancer-Genetics (BioNLP-ST 2013)	2013	BioC, standoff		http://aclweb.org/anthology/W/W13/W13-2008.pdf	http://2013.bionlp-st.org/tasks/cancer-genetics	
CHEMDNER (BioCreative IV Track 2)	2013	BioC / standoff		http://www.biocreative.org/media/store/files/2013/bc4_v2_1.pdf	http://www.biocreative.org/tasks/biocreative-iv/chemdner/	
Chemical Patent Corpus	2014	standoff	200 patents	http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0107477	http://biosemantics.org/index.php/resources/chemical-patent-corpus	
CoMAGC	2013	XML	821 sentences on prostate, breast and ovarian cancer	http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-14-323	http://biopathway.org/CoMAGC/	
CRAFT	2012		97 full OA biomedical articles		http://bionlp-corpora.sourceforge.net/CRAFT/	
Craven (Wisconsin corpus)	1999	other	1,529,731 sentences (automated)	https://www.biostat.wisc.edu/~craven/ie/ReadMe	https://www.biostat.wisc.edu/~craven/ie/	
CTD (BioCreative IV Track 3)		BioC			http://www.biocreative.org/tasks/biocreative-iv/track-3-CTD/	
DDICorpus	2011 2013	BioC	792 texts from DrugBank and 233 Medline abstracts	https://www.ncbi.nlm.nih.gov/pubmed/23906817	http://bioc.sourceforge.net/ http://corpora.informatik.hu-berlin.de/	http://labda.inf.uc3m.es/ddicorpus
DIP-PPI (Database of Interaction Proteins)		other	Only proteins from yeast.		https://www2.informatik.hu-berlin.de/~hakenber/corpora/	

EBI:diseases	2008	other	856 sentences from 624 abstracts	http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-S3-S3	https://www2.informatik.hu-berlin.de/~hakenber/links/benchmarks.html	ftp://ftp.ebi.ac.uk/pub/software/textmining/corpora/diseases
eFIP	2012 2015	xlsx		https://www.ncbi.nlm.nih.gov/pubmed/23221174 https://www.ncbi.nlm.nih.gov/pubmed/25833953	http://research.bioinformatics.udel.edu/iprolink/corpora.php	
EMU (Extractor of Mutations)	2011	other		https://www.ncbi.nlm.nih.gov/pubmed/21138947	http://bioinf.umbc.edu/EMU/ftp/	
EU-ADR	2012	other	300 PubMed abstracts (drug-disoder, drug-target, gene-disorder, SNP-disorder)	http://www.sciencedirect.com/science/article/pii/S1532046412000573	http://biosemantics.org/index.php/resources/euadr-corpus	
Exhaustive PTM (BioNLP 2011)				http://dl.acm.org/citation.cfm?id=2002902.2002920	https://github.com/dterg/exhaustive-ptm	
FlySlip	2007	CONLL	82 abstracts, 5 full papers	https://www.ncbi.nlm.nih.gov/pubmed/17990496	http://compbio.ucdenver.edu/ccp/corpora/obtaining.shtml	http://www.wiki.cl.cam.ac.uk/rowiki/NaturalLanguage/FlySlip/Flyslip-resources
FSU-PRGE	2010	leXML	3236 MEDLINE abstracts (35,519 sentences)	http://aclweb.org/anthology/W/W10/W10-1838.pdf	http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/corpora.html	
GAD	2015	csv		http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0472-9	http://ibi.imim.es/research-lines/biomedical-text-mining/corpora/	
GeneReg	2010	BioC	314 Abstracts	http://www.lrec-conf.org/proceedings/lrec2010/pdf/407_Paper.pdf	http://corpora.informatik.hu-berlin.de/	http://www.julielab.de/Resources/GeneReg.html
GeneTag (BioCreative II Gene Mention)	2005	BioC	20,000 sentences MEDLINE	https://www.ncbi.nlm.nih.gov/pubmed/15960837	https://www2.informatik.hu-berlin.de/~hakenber/links/benchmarks.html http://bioc.sourceforge.net/	
GENIA (BioNLP Shared Task 2009)					http://www.nactem.ac.uk/tsujii/GENIA/SharedTask/detail.shtml#downloads	
GENIA (BioNLP Shared Task 2011) GE, EPI, ID, REL,		BioC, standoff			https://sites.google.com/site/bionlpst/home/epigenetics-and-post-translational-modifications http://2011.bionlp-st.org	http://corpora.informatik.hu-berlin.de/

REN, CO, BB, BI						
GENIA (term annotation)	2003	BioC, XML			http://corpora.informatik.hu-berlin.de/	http://www.nactem.ac.uk/aNT/genia.html
GETM	2010	BioC, standoff		http://dl.acm.org/citation.cfm?id=1869970	http://corpora.informatik.hu-berlin.de/	http://getm-project.sourceforge.net/
GREC (Gene Regulation Event Corpus)	2009	BioC, standoff, XML	240 MEDLINE (167 on E.coli and 73 on Human)	http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-10-349	http://corpora.informatik.hu-berlin.de/	http://www.nactem.ac.uk/GREC/
HIMERA	2016	standoff		http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0144717	http://www.nactem.ac.uk/himera/	
HPRD50 (Human Protein Reference Database)	2004	BioC	50 abstracts	https://www.ncbi.nlm.nih.gov/pubmed/14681466	http://corpora.informatik.hu-berlin.de/	http://www2.bio.ifi.lmu.de/publications/RelEx/
IEPA	2002	BioC	slightly over 300 MEDLINE abstracts	https://www.ncbi.nlm.nih.gov/pubmed/11928487	http://corpora.informatik.hu-berlin.de/	http://orbit.nlm.nih.gov/resource/iepa-corpus
iHOP	2004	other	~ 160 sentences	https://www.ncbi.nlm.nih.gov/pubmed/15226743	http://www.ihop-net.org/UniPub/iHOP/info/gene_index/manual/1.html	
iProLINK / RLIMS	2004	other, XML, BioC		https://www.ncbi.nlm.nih.gov/pubmed/15556482	http://research.bioinformatics.udel.edu/iprolink/corpora.php	
iSimp	2014	BioC	130 MEDLINE abstracts (1199 sentences)	https://www.ncbi.nlm.nih.gov/pubmed/24850848	http://research.bioinformatics.udel.edu/isimp/corpus.html	
Linnaeus	2010	standoff		http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-85	https://www2.informatik.hu-berlin.de/~hakenber/links/benchmarks.html	http://linnaeus.sourceforge.net/
LLL (Learning Language in Logic)	2005	BioC		https://www.cs.york.ac.uk/aig/lll/lll05/lll05-nedellec.pdf	http://corpora.informatik.hu-berlin.de/	http://genome.jouy.inra.fr/texte/LLLchallenge/
MEDSTRACT		BioC	199 PubMed citations	https://www.ncbi.nlm.nih.gov/pubmed/11604766	http://bioc.sourceforge.net/	
MedTag	2005	other		https://www.researchgate.net/publication/234785358_MedTag_a_collection_of_biomedical_annotations	ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedTag/medtag.tar.gz https://sourceforge.net/projects/medtag/	
Metabolite and Enzyme	2011	BioC, XML	296 abstracts	http://link.springer.com/article/10.1007%2Fs11306-010-0251-6	http://www.nactem.ac.uk/metabolite-corpus/	http://argo.nactem.ac.uk/bioc/
miRTex	2015	BioC, standoff	350 abstracts (200 development, 150 test)	http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004391	http://research.bioinformatics.udel.edu/iprolink/corpora.php	

MLEE	2012	CONLL, standoff	262 PubMed abstracts on molecular mechanisms of cancer (specifically relating to angiogenesis)	https://academic.oup.com/bioinformatics/article/28/18/i575/249872/Event-extraction-across-multiple-levels-of	http://nactem.ac.uk/MLEE/	
mTOR pathway event corpus (BioNLP 2011)	2011	standoff		http://dl.acm.org/citation.cfm?id=2002919	https://github.com/dterg/mtor-pathway/tree/master/original-data	
Mutation Finder	2007	other	305 abstract (development data set), 508 abstract test set	https://www.ncbi.nlm.nih.gov/pubmed/17495998	http://mutationfinder.sourceforge.net/	https://github.com/rockt/SETH
Nagel		XML, standoff			http://sourceforge.net/projects/bionlp-corpora/files/ProteinResidue/	
NCBI Disease	2012	other	6881 sentences in 793 PubMed abstracts	https://www.ncbi.nlm.nih.gov/pubmed/24393765	http://www.ncbi.nlm.nih.gov/CBBresearch/Fellows/Dogan/disease.html	
OMM (Open Mutation Miner)	2012	other	40 full texts	https://www.ncbi.nlm.nih.gov/pubmed/22759648	http://www.semanticsoftware.info/open-mutation-miner	
OSIRIS	2008	BioC, XML, standoff	105 articles	https://www.ncbi.nlm.nih.gov/pubmed/18251998	http://corpora.informatik.hu-berlin.de/	https://sites.google.com/site/laurafurlongweb/databases-and-tools/corpora
PC (Pathway Curation) (BioNLP-ST 2013)	2013	BioC			http://argo.nactem.ac.uk/bioc/	http://2013.bionlp-st.org/tasks/pathway-curation
PennBioIE-oncology	2004	leXML	1414 PubMed abstracts on cancer	http://www.aclweb.org/anthology/W04-3111	http://www.ebi.ac.uk/Rebholz-srv/CALBC/corpora/corpora.html	
pGenN (Plant-GN)	2015	BioC	104 MEDLINE abstracts	http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0135305	http://research.bioinformatics.udel.edu/iprolink/corpora.php	
PICAD	2011	XML	1037 sentences from PubMed	http://dl.acm.org/citation.cfm?doid=2147805.2147853	http://ani.stat.fsu.edu/~jinfeng/resources/PICAD.txt	http://corpora.informatik.hu-berlin.de/
PolySearch (includes v1. and v2.)		other		https://www.ncbi.nlm.nih.gov/pubmed/25925572	http://polysearch.cs.ualberta.ca/downloads	
ProteinResidue		other			http://bionlp-corpora.sourceforge.net/	
SCAI_Klinger	2008	CONLL		https://academic.oup.com/bioinformatics/article/24/13/i268/235854/Det	https://www.scai.fraunhofer.de/en/business-research-	

				ection-of-IUPAC-and-IUPAC-like-chemical-names	areas/bioinformatics/downloads/corpora-for-chemical-entity-recognition.html	
SCAI_Kolarik	2008	CONLL		http://www.lrec-conf.org/proceedings/lrec2008/workshops/W4_Proceedings.pdf#page=55	https://www.scai.fraunhofer.de/en/business-research-areas/bioinformatics/downloads/corpora-for-chemical-entity-recognition.html	
SETH	2016	standoff	630 publications from The American Journal of Human Genetics and Human Mutation	https://www.ncbi.nlm.nih.gov/pubmed/?term=27256315	https://github.com/rockt/SETH/tree/master/resources/SETH-corpus	
SH (Schwartz and Hearst)	2003	BioC	1000 PubMed Abstracts	https://www.ncbi.nlm.nih.gov/pubmed/12603049	http://bioc.sourceforge.net/	
SNPCorpus	2011	BioC	296 MEDLINE abstracts	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3194196/	http://corpora.informatik.hu-berlin.de/	http://www.scai.fraunhofer.de/snp-normalization-corpus.html
Species	2013	standoff	800 PubMed abstracts	http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0065390	http://species.jensenlab.org/	http://species.jensenlab.org/
T4SS (Type 4 Secretion System)	2011	CONLL		http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0014780	http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0014780	
T4SS Event Extraction (BioNLP 2010)	2010	other		http://dl.acm.org/citation.cfm?id=1869961.1869980	https://github.com/dterg/t4ss-event	
tmVar	2013	BioC	500 PubMed abstracts	https://www.ncbi.nlm.nih.gov/pubmed/23564842	https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/#tmVar	https://github.com/rockt/SETH
VariomeCorpus (hvp)	2013	BioC		https://www.ncbi.nlm.nih.gov/pubmed/23584833	http://corpora.informatik.hu-berlin.de/	http://www.opennicta.com/home/health/variome
Yapex	2002	other	99 training, 101 test MEDLINE abstracts	https://www.ncbi.nlm.nih.gov/pubmed/12460631	http://www.rostlab.org/~nlprot/yapex.txt	https://www2.informatik.hu-berlin.de/~hakenber/links/benchmarks.html

Supplementary Methods

Translational utility of a hierarchical classification strategy in biomolecular data analytics

Dieter Galea, Paolo Inglese, Lidia Cammack, Nicole Strittmatter, Monica Rebec, Reza Mirnezami, Ivan Laponogov, James Kinross, Jeremy Nicholson, Zoltan Takats & Kirill A. Veselkov

Rights and Permissions - Open Access: *This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/.*

Data acquisition and pre-processing

Mass spectral profiles for 15 different clinically-isolated bacterial species (15–20 samples per species; cultured on Columbia horse blood agar in anaerobic conditions for *Clostridium difficile* and aerobic conditions for all other species) were acquired by rapid evaporative ionization mass spectrometry (REIMS) using an Exactive instrument (Thermo Fisher Scientific, San Jose, California)9. Spectral preprocessing was performed using an in-house MATLAB workflow. Spectral data were converted from RAW to mzXML format and imported at 0.001 Da resolution within the mass range of 150–2000 m/z. Noise was removed by an optimum threshold adaptively calculated using a histogram-based method19. Five mass spectra per sample were selected by total ion intensity (TIC) after optimum thresholding, and peak detection was performed by a 3rd order derivative of a Savitzy-Golay polynomial filter20. Peak-matching was performed following determination of a common m/z feature vector estimated using a kernel density estimation approach21, where a peak with the highest peak counts was considered as a common m/z value for all the spectra. A mean mass spectrum for each bacterial species sample was derived and subsequently median-fold change normalization was applied to adjust for relative ion intensities between spectra. We have previously demonstrated the robustness of this data processing strategy22.

Mass spectrometric data shows heteroscedastic variation of ionic intensities, where technical variance increases as a function of signal intensity, resulting in additive or multiplicative signal. To account for the unmet assumption of multivariate statistical techniques (that noise structure is consistent throughout the whole intensity range) log variance stabilizing transformation was performed on the normalized spectra23.

Hierarchy structure was defined by the taxonomy of bacterial species, where: Gram staining type, class, order, family, genus and species were considered. Taxonomic information was automatically retrieved from the National Microbial Pathogen Data Resource (http://www.nmpdr.org/), and List of Prokaryotic Names with Standing in Nomenclature (http://www.bacterio.net/) online data repositories.

For the cancer genomic data, raw RNA sequencing data (RNASeqV2) for 9 cancer types (breast adenocarcinoma (BRCA), glioblastoma multiforme (GBM), kidney renal clear cell carcinoma (KIRC), kidney renal papillary carcinoma (KIRP), acute myeloid leukemia (LAML), lower grade glioma (LGG), lung adenocarcinoma (LUAD), prostate adenocarcinoma (PRAD), and bladder urothelial carcinoma (BLCA)), profiled using Illumina HiSeq, were compiled from The Cancer Genome Atlas data repository (https://tcga-data.nci.nih.gov/tcga/). Data preprocessing and filtering was performed using an in-house RNA sequencing workflow. This involved median fold change normalization followed by variance stabilizing log-2 transformation. Dubiously annotated genes were removed as well as normal tissue samples. To avoid biases, in circumstances where multiple samples were available for the same patient, a single sample was selected at random.

Cancer gene expression data was organized in a pre-defined two-level hierarchy: (i) cancer type; and subsequently (ii) cancer sub-type. Subtype assignment was based on previous literature with published gene expression unsupervised cluster assignment of mutual samples used here24,25,26,27,28,29,30,31,32,33,34,35. Subtypes with less than 15 samples were removed. A total of 1960 mutual samples/patients with pre-assigned subtypes were retrieved. Clusters correlated in the literature with other molecular subtypes or clinical outcomes, such as overall survival, were assigned this information label to indicate potential implications, interpretations and/or applications of the findings reported here (Information Note 1).

Classification algorithm

The developed algorithm is based on training a discrimination model for each node of a hierarchical tree, stratifying the data at each parent node. Starting from the upper-most node, a selective dimensionality reduction method approach36 is implemented using 4 different methods: (i) alternative partial least squares regression (SIMPLS)37, (ii) recursive linear discriminant analysis using maximum margin criterion (MMC-LDA)38, (iii) support vector machine (SVM) using LIBSVM version 3.20 (https://www.csie.ntu.edu.tw/~cjlin/libsvm/), and (iv) linear discriminant analysis using the Fisherfaces approach (PCA-LDA)39. Table S1 summarizes the eigenvalue decompositions used to derive the components in each of these.

Table S1 Principles of the dimensionality reduction techniques used in the classification and prediction algorithms.

Method	Method Abbrev.	Components derivation
Principal Component Analysis	PCA	Maximizes overall dataset variance without considering between-class variance
Partial Least Squares	PLS	Maximizes between-class variance without considering within-class variance

Method	Method Abbrev.	Components derivation
Maximum Margin Criterion	MMC	Maximizes between-class variance, while minimizing within-class variance
Linear Discriminant Analysis	LDA	Maximizes ratio of between- and within-class variation while the number of samples is greater than the number of variables
Support Vector Machines	SVM	Maximizes the margin of separation between the classes

Data are initially split into 5-fold cross-validated main test and training sets, and in turn the training set is then split into nested 5-fold cross-validated test and training sets (Figure S1). All 4 methods outlined above were trained on the nested training set and then applied to the nested test set. The classifier achieving the highest classification accuracies throughout all the 5-fold cross-validations is saved, therefore the most generalizable and best-performing method across the data strata is selected (Figure S1). In cases where different classifiers give equal classification accuracy, the method with the shortest computational time is chosen. This ensures that maximum performance is achieved by choosing the 'best classifier', whilst simultaneously ensuring computational efficiency. Via this approach, a 'method map' is derived which is subsequently applied to the main outer test sets, giving rise to the classification accuracies that have been reported here. In each outer cross-validation, the training and test set are fixed for the subsequent nested cross-validation in order to ensure unbiased comparison of method performance.

The user is prompted with a dialog to adjust the cross-validation rounds according to the minimum class size, ensuring robustness of the selected model. Where a parent node has only 1 down-stream node, a model is built to discriminate between the single down-stream node and other parent-related lower order nodes. For example, if a genus (level 5) has only 1 species (level 6), and the genus is related to other genera through the family (level 4), a model is built between the 'offspring' nodes of these related species and the single species. Samples of this single class/species which are not classified into the same species are considered as 'non-classified'.

Probabilistic classification is performed throughout by logistic regression, where samples are assigned to the class with the highest probability. In a multi-class classification problem, classifiers are applied in a one-against-all manner40. In addition to nested cross-validation, classifications are repeated to obtain an average classification accuracy for each node, at each level, assessing model discriminatory performance and stability. The choice of the above linear models is determined by their scalability to high-throughput datasets and the transparency of derived discriminatory molecular signatures.

Graphic visualization is provided while the algorithm is operating to indicate workflow progress, as well as accuracies during the training phase. When complete, a set of confusion matrices are generated to summarize average classification accuracy for each node, at each taxonomic level. In addition, we present an alternative semi-quantitative visualization technique that can intuitively summarize the performance of hierarchical classification methods, simplifying the presentation and identification of class misclassifications.

Class-prediction algorithm

Based on the HC approach, a class prediction algorithm was developed and tested using a leave-class out cross-validation strategy where each class/down-stream-node from the lower-most hierarchical level was excluded completely from the training phase, one at a time. The most efficient cross-validated method map was again determined, now based on the new cross-validated data 'subset'. The left-out class is then applied to the method map and predicted. Starting from the root node at the upper-most hierarchical level, each sample of the left-out class is assigned to a down-stream class based on probability estimates. A probability difference between the highest probability class and the second is set as a threshold, below which samples are considered as 'non-classified'. Percentage prediction accuracy is determined for each species after repeated predictions to assess predictive robustness.

Flowcharts illustrating the classification workflow and leave-one-out class prediction algorithms are presented in Figure S1. Classification algorithms described here were developed in MATLAB 2014a.

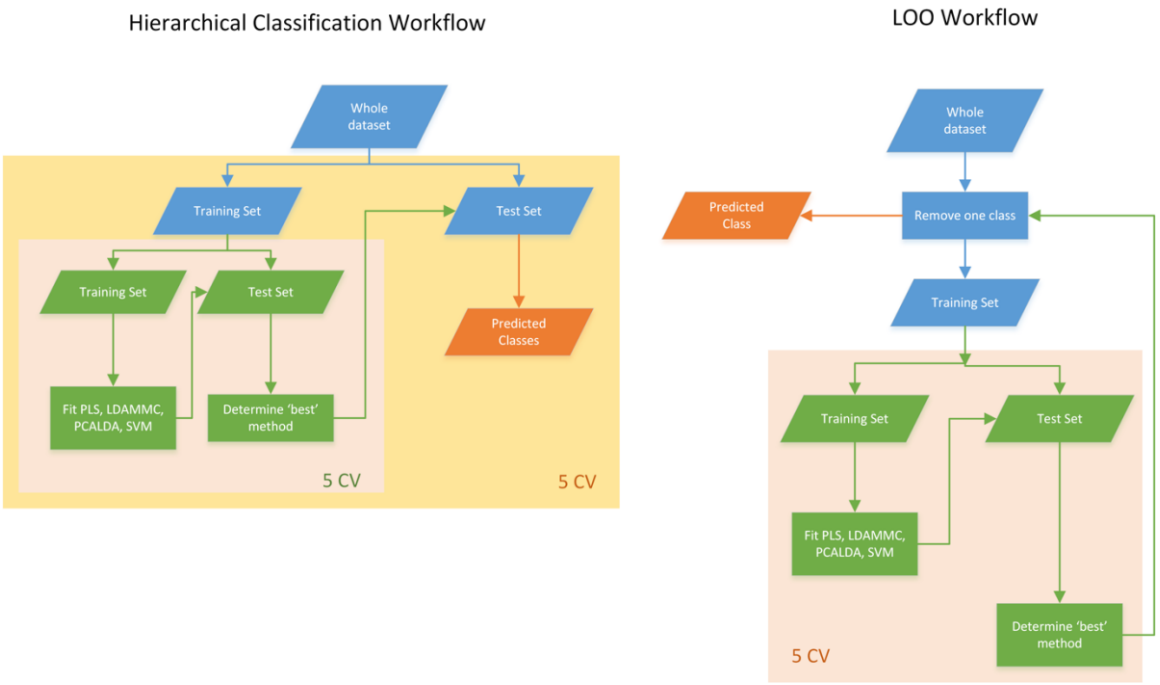


Figure S1. Flow charts for the classification and class prediction algorithms. A) Nested cross-validated selective classifier approach algorithm indicating splitting and fixing of training/test set, dimensionality reduction using MMCLDA, SVM, PCALDA and PLS, classification and prediction by logistic regression and the most accurate method is applied to the test set for classification of the test set. B) Modification to the probabilistic classification model for leave-one-out (LOO) prediction where the best model is used to classify a class that is not part of the model training phase. Predictions are given for each hierarchical level.

Code/data availability

The source code for the developed algorithms and data are available at: https://bitbucket.org/iAnalytica/hierarchical-classification-publication/overview.

Information Note 1 – Cancer subtype information retrieved from literature

Kidney Renal Clear Cell Carcinoma – KIRC

KIRC mRNA clusters were reported[24] to relate to the clear cell type A (ccA) and clear cell type B (ccB) expression subtypes[28], with cluster 1 found to be correlated to ccA, cluster 2 and 3 correlated with ccB, and cluster 4, while not correlated with neither ccA nor ccB, is suggested to account for ~15% tumors previously unclassified in ccA/ccB classes.

Kidney Renal Papillary Cell Carcinoma – KIRP

KIRP mRNA clusters 1 and 3 were reported[25] to be dominated by papillary renal cell carcinoma (pRCC) Type I, Stage I-II while cluster 2 was dominated by pRCC Type II, Stage III-IV.

Acute Myeloid Leukemia – LAML

LAML mRNA clusters were reported[30] to correlate with the French-American-British (FAB) subtype classification of acute leukemias[39], with cluster 3 represented FAB subtype M3 (acute promyelocytic leukemia), cluster 4 associated with FAB subtype M1 (AML with minimal maturation), cluster 5 with FAB subtype M4 (acute myelomonocytic leukemia), and cluster 7 with FAB subtype M5 (acute monoblastic or monocytic leukemia). Clusters 1, 2, and 6 were not correlated with FAB subtypes and thus survival information was retrieved from the Kaplan Meier overall survival curve provided in the original study.

Lower Grade Glioma – LGG

The Cancer Genome Atlas Research Network[29] report 4 mRNA clusters for LGG. Cluster 2 was enriched for IDHwt (Isocitrate dehydrogenase wild-type) tumor samples with the worst overall survival compared to the other clusters, cluster 1 was correlated with DNA methylation subtype M5 and DNA methylation subtype M3 identified in the same study, cluster 3 was found to be composed entirely of IDH-mut-codel (IDH-mutation 1p/19q codeletion) gliomas, while cluster 4 was found not to be associated with any specific subtype.

Prostate Adenocarcinoma – PRAD

3 mRNA clusters were found to be highly correlated with ETS (E26 transformation-specific transcription factors) fusion status by The Cancer Genome Atlas Research Network[31], with cluster 1 consisted primarily of ETS-negative tumors while clusters 2 and 3 reported to contain ETS fusion-positive tumors.

Glioblastoma Multiforme – GBM

Proneural, neural, classical and mesenchymal molecular subclasses of GBM are transcriptomic clinically-relevant subtypes reported in literature[40]. Cluster assignment for specific samples were retrieved from The Cancer Genome Atlas Research Network[32].

Breast Adenocarcinoma – BRCA

The Cancer Genome Atlas Research Network[27] reported 3 breast tumor types based on mRNA sequencing data; reactive-like, proliferative, and immune-related. These were identified to have several genomic differences at the mRNA and protein/phosphoprotein level but not somatic mutations or DNA copy-number alterations.

Bladder Urothelial Carcinoma – BLCA

4 main mRNA subtypes were identified for BLCA in the study by The Cancer Genome Atlas Research Network[26]. Cluster 1 samples were enriched with FGFR3 alterations, FGFR3 expression and reduced FGFR3-related miRNA expression. Cluster 3 samples were dominated by squamous features, mRNA subtype 4 showed low expression of FGFR3 mRNA, while cluster 2 showed lower KRT and EGFR expression compared to the other subtypes.

Lung Adenocarcinoma – LUAD

The Cancer Genome Atlas Research Network[28] propose 3 molecular subtypes for LUAD, based on the clustering of mRNA. Namely: Terminal Respiratory Unit (TRU), proximal-inflammatory and proximal-proliferative transcriptional subtypes. Proximal-proliferative was found to be enriched for KRAS mutations, inactivation of STK11 tumor suppressor gene, and reduced gene expression. Proximal-inflammatory was associated with the co-mutation of NF1 and TP53, while TRU cluster samples were highly EGFR-mutated and expressing kinase fusion.

Supplementary References

9. Strittmatter, N. *et al*. Characterization and identification of clinically relevant microorganisms using rapid evaporative ionization mass spectrometry. *Anal. Chem.* 86, 6555–6562, https://doi.org/10.1021/ac501075f (2014).

19. Otsu, N. A threshold selection method from Gray-level histograms. *IEEE Trans. Syst., Man, Cybern., Syst.* 9, 62–66 (1975).

20. Savitzky, A. G. M. J. E. Smoothing and differentiation of data by simplified least squares procedures. *Anal. Chem.* 36, 1627–1639, https://doi.org/10.1021/ac60214a047(1964).

21. Fushiki, T., Fujisawa, H. & Eguchi, S. Identification of biomarkers from mass spectrometry data using a "common" peak approach. *BMC Bioinform.* 7, 1–9, https://doi.org/10.1186/1471-2105-7-358(2006).

22. Veselkov, K. A. *et al*. Optimized preprocessing of ultra-performance liquid chromatogrpahy/mass spectrometry urinary metabolic profiles for improved information recovery. *Anal. Chem.* 83, 5864–5872, https://doi.org/10.1021/ac201065j (2011).

23. Veselkov, K. A. *et al*. Chemo-informatic strategy for imaging mass spectrometry-based hyperspectral profiling of lipid signatures in colorectal cancer. *Proc. Natl. Acad. Sci. USA* 111, 1216–1221, https://doi.org/10.1073/pnas.1310524111(2014).

24. Network, T. C. G. A. R. Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature* 499, 43–49, https://doi.org/10.1038/nature12222(2013).

25. Network, T. C. G. A. R. Comprehensive molecular characterization of papillary renal-cell carcinoma. *N. Eng. J. Med.* 374, 135–145, https://doi.org/10.1056/NEJMoa1505917(2016).

26. Network, T. C. G. A. R. Comprehensive molecular characterization of urothelial bladder carcinoma. *Nature* 507, 315–322, https://doi.org/10.1038/nature12965(2014).

27. Network, T. C. G. A. R. Comprehensive molecular portraits of invasive lobular breast cancer. *Cell* 163, 506–519, https://doi.org/10.1038/nature12965(2015).

28. Network, T. C. G. A. R. Comprehensive molecular profiling of lung adenocarcinoma. *Nature* 511, 543–550, https://doi.org/10.1038/nature13385(2014).

29. Network, T. C. G. A. R. Comprehensive, integrative genomic analysis of diffuse lower-grade gliomas. *N. Engl. J. Med.* 372, 2481–2498, https://doi.org/10.1056/NEJMoa1402121(2015).

30. Network, T. C. G. A. R. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *N. Engl. J. Med.*368, 2059–2074, https://doi.org/10.1056/NEJMoa1301689(2013).

31. Network, T. C. G. A. R. The molecular taxonomy of primary prostate cancer. *Cell*163, 1011–1025, https://doi.org/10.1016/j.cell.2015.10.025(2015).

32. Network, T. C. G. A. R. The somatic genomic landscape of glioblastoma. *Cell*155, 462–477, https://doi.org/10.1016/j.cell.2013.09.034(2013).

33. Brannon, A. R. *et al*. Molecular stratification of clear cell renal cell carcinoma by consensus clustering reveals distinct subtypes and survival patterns. *Genes Cancer* 1, 152–163, https://doi.org/10.1177/1947601909359929(2010).

34. Bennett, J. M. *et al*. Proposals for the classification of the acute leukaemias French-American-British (FAB) co-operative group. *Br. J. Haematol.* 33, 451–458, https://doi.org/10.1111/j.1365-2141.1976.tb03563.x (1976).

35. Verhaak, R. G. *et al*. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell* 17, 98–110, https://doi.org/10.1016/j.ccr.2009.12.020(2010).

36. Secker, A. D. *et al*. An experimental comparison of classification algorithms for the hierarchical prediction of protein function. *Expert Update* 9, 17–22 (2007).

37. De Jong, S. SIMPLS: An alternative approach to partial least squares regression. *Chemometr. Intell. Lab* 18, 251–263, https://doi.org/10.1016/0169-7439(93)85002-X (1993).

38. Li, H., Jiang, T. & Zhang, K. Efficient and robust feature extraction by maximum margin criterion. *IEEE Trans. Neural Netw.* 17, 157–165, https://doi.org/10.1109/TNN.2005.860852(2004).

39. Belhumeur, P. N., Hespanha, J. P. & Kriegman, D. J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* 19, 711–720, https://doi.org/10.1109/34.598228 (1997).

40. Lorena, A. C., Carvalhom A. C. P. L. F. & Gama, J. M. A review on the combination of binary classifiers in multiclass problems. *Artif. Intell. Rev*. 30, https://doi.org/10.1007/s10462-009-9114-9 (2008).