Imperial College London
Department of Computing

Generating Argumentation Frameworks from text

Oana Cocarascu

April 2019

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London
and the Diploma of Imperial College London

Declaration

I declare that this thesis was composed by myself and that the work it presents is my own, except where otherwise stated. I certify that all material in this thesis which is not my own work has been properly acknowledged.

Oana Cocarascu

Abstract

Artificial Intelligence targets the design and creation of intelligent machines that behave like humans. One fundamental aspect of human intelligence is the ability to argue. In the past years, the field of Computational Argumentation has seen a remarkable expansion within the Natural Language Processing (NLP) community with the potential to have an impact on real world applications ranging from legal and healthcare to finance and (social) media. Our contributions to this field are threefold: we develop models for Argument Mining (AM), methods for mining Argumentation Frameworks (AFs) from text, and applications of the two in real world settings. We focus on one area in AM, namely Relation-based AM (RbAM), that deals with determining the argumentative relation of attack or support between texts. We bring together existing datasets available for RbAM, covering various domains and topics, and propose neural models for the cross-domain RbAM classification task. We show that our models perform homogeneously over all existing datasets for relation prediction in AM overcoming the issue of AM models that are hardly portable from one application domain to another. We also present methods for extracting AFs from text, using the developed AM models in combination with NLP techniques, and Abstract Argumentation (AA)-driven Case-Based Reasoning (CBR), an existing reasoning formalism. We apply AM in detecting deceptive reviews and in extracting arguments from a movie review website, Rotten Tomatoes, one of the most trusted websites for movies, whilst providing explanations extracted from the AFs mined from reviews. We apply AA-CBR in the Sentiment Analysis task to determine the sentiment polarity of short texts and show empirically that reasoning with AFs leads to an improvement in results compared to standard machine learning models whilst also illustrating the portability of explanations to binary classification problems.

Copyright Declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Acknowledgements

Above all, I would like to thank Francesca Toni for giving me the opportunity to pursue a PhD, supporting me throughout, introducing me to various research areas, projects, and colleagues in the field, allowing me to gain teaching experience, and to develop career-wise. She has supported me in my academic endeavors and I am grateful to her for caring about my academic development and for giving me the freedom to pursue any interests I had during my studies.

My group, Computational Logic and Argumentation, is the next one I would like to thank for the extremely valuable discussions throughout my studies, in particular with Antonio and Kris. Antonio especially deserves my thanks for being a great collaborator and a great friend.

Also, I would like to thank Elena Cabrio and Serena Villata for giving me the opportunity to work with them in France. I would like to thank Christos Christodoulopoulos, Arpit Mittal, James Thorne, and Andreas Vlachos for allowing me to join them in the organisation of the FEVER workshops.

Zero contributions in terms of research I have brought to Philippa Gardner's research group, Verified Trustworthy Software Specification, but that nonetheless accepted me as being part of the group and included me in all celebrations. Thanks all for providing a great atmosphere and research environment. In particular, I would like to thank Teresa and Petar for being great friends, always there to help.

In addition, I would like to thank Dalal Alrajeh, Robert Chatley, and Duncan Gillies for indirectly contributing to my decision to pursue a PhD. I would like to thank Amani El-Kholy for looking after students and solving all the admin issues. I would also like to thank my examiners, Alessandra Russo and Chris Reed, for the great discussion and valuable feedback during a very enjoyable viva.

Next, special thanks to the supporting cast (in order of appearance): Paulina, Michal, Nuri, Antonio, Shale, Jose, Gian, Petar, Teresa & Petros, Giulia.

Grateful for having the best parents one can dream of. I cannot put into words how lucky I am. Merci pentru absolut tot.

Contents

1. Introduction **15**

1.1. Thesis Contributions 19

1.1.1. Portability of RbAM 19

1.1.2. Portability of RbAM: from downstream task to mining BAFs for applications 20

1.1.3. Portability of method that empowers explanations: from reasoning to mining AAFs 21

1.2. Thesis Structure 22

1.3. Publications 22

2. Background and Literature review **24**

2.1. Core Natural Language Processing and Deep Learning background 24

2.1.1. Natural Language Processing - Preprocessing 24

2.1.1.1. Tokenization 25

2.1.1.2. Normalization 25

2.1.1.3. POS tagging 26

2.1.2. Word vectors 26

2.1.2.1. Standard word vectors 26

2.1.2.2. Neural embeddings 27

2.1.3. Deep learning classifiers 29

2.2. Argument Mining 32

2.2.1. Argument Mining component detection 32

2.2.2. Relation-based Argument Mining 34

2.2.2.1. Datasets for Relation-based Argument Mining . 34

2.2.2.2. Methods for Relation-based Argument Mining . 36

2.3. Argumentation 38

2.3.1. Abstract Argumentation Frameworks 39

2.3.1.1. AA-CBR 40

2.3.2. (Quantitative) Bipolar Argumentation Frameworks 44

2.3.3. Other approaches . . . 49
2.4. Explainable AI . . . 49
2.5. Summary and Discussion . . . 52

3. Deep Learning for 2 class Relation-based Argument Mining 53
3.1. Motivation . . . 53
3.2. Datasets for 2 class Relation-based Argument Mining . . . 55
3.3. Neural models . . . 62
3.3.1. Features . . . 62
3.3.2. Concatenation model (C) . . . 63
3.3.3. Mix model (M) . . . 64
3.3.4. Autoencoder model (AE) . . . 65
3.3.5. Attention model (A) . . . 66
3.4. Experimental results . . . 67
3.5. Summary and Discussion . . . 72

4. Deep Learning for 3 class Relation-based Argument Mining 73
4.1. Motivation . . . 73
4.2. Deep Learning for 3 class Relation-based Argument Mining . . . 74
4.2.1. Datasets for 3 class Relation-based Argument Mining . . . 75
4.2.2. Neural models . . . 78
4.2.3. Experimental results . . . 78
4.3. Comparing mined BAFs with annotators . . . 82
4.3.1. Mining BAFs from a factual text . . . 83
4.3.1.1. Merged arguments . . . 84
4.3.1.2. Unmerged arguments . . . 85
4.3.2. Mining BAFs from a dialogue . . . 88
4.3.2.1. Annotation task . . . 89
4.3.2.2. Results . . . 92
4.4. Summary and Discussion . . . 94

5. Using mined Bipolar Argumentation Frameworks in applications 96
5.1. Motivation . . . 96
5.2. Mining Bipolar Argumentation Frameworks to determine movie scores . . . 99
5.2.1. Feature-based characterisation of movies . . . 99
5.2.1.1. Extracting Review Aggregations . . . 101

5.2.1.2. Mining votes using Sentiment Analysis 104
5.2.1.3. Mining votes using Relation-based Argument Mining 105
5.2.1.4. Formulating QBAFs from Review Aggregations 105
5.2.2. Determining Rotten Tomatoes scores: experimental results 106
5.2.2.1. Experimental Results 107
5.2.2.2. Qualitative Assessment 109
5.2.3. Explanations from mined Bipolar Argumentation Frameworks: an illustration 111
5.3. Mining Bipolar Argumentation Frameworks from reviews for deception detection 119
5.3.1. Mining Bipolar Argumentation Frameworks methodology 120
5.3.1.1. Building a topic-dependent BAF 121
5.3.1.2. Identifying arguments 121
5.3.1.3. Clustering arguments by topics 122
5.3.1.4. Determining argumentative relations 123
5.3.1.5. BAFs from reviews 124
5.3.1.6. From BAFs to argumentative features 126
5.3.2. Detecting deceptive reviews: experimental results 127
5.3.3. Explanations from mined Bipolar Argumentation Frameworks: an illustration 131
5.4. Summary and Discussion 138

6. Mining Abstract Argumentation Frameworks by AA-CBR 142
6.1. Motivation 142
6.2. Pipeline for predicting with AA-CBR 143
6.2.1. Stage 1: Feature selector 144
6.2.2. Stage 2: AA-CBR miner 145
6.2.3. Stage 3: AA-CBR predictor 145
6.2.4. Stage 4: AA-CBR explainer 146
6.3. Predicting and explaining from categorical data with AA-CBR . . 148
6.3.1. Stage 1: Feature selection 149
6.3.2. Stages 2 and 3: Empirical evaluation 150
6.3.3. Stage 4: Illustration of AA-CBR explanations 152

6.4. Predicting and explaining from textual data with AA-CBR 154
6.4.1. Stage 1: Feature selection 155
6.4.1.1. Clustering related terms 155
6.4.1.2. Random Forests for Feature selection 157
6.4.2. Upgraded AA-CBR: Changing the attack definition 157
6.4.3. Stages 2 and 3: Empirical evaluation 158
6.4.4. Stage 4: Illustration of AA-CBR explanations 159
6.5. Summary and Discussion 163

7. Conclusion **166**
7.1. Summary of Contributions 166
7.2. Future work 168
7.2.1. Portability 169
7.2.2. Explainability 170

Bibliography **172**

Appendices **192**

A. 2 class Relation-based Argument Mining **193**
A.1. 100 words for sequence length 193
A.1.1. Main training dataset: *aif* 193
A.1.2. Main training dataset: *essay* 194
A.1.3. Main training dataset: *ibm* 200
A.1.4. Main training dataset: *web* 202

B. 3 class Relation-based Argument Mining **204**

List of Tables

3.1. Summary of datasets. 62
3.2. Results using *syntactic* features on the datasets with attack (A) and support (S) relations. RF stands for Random Forests, A for autoencoder model, C for concatenation model, M for mix model, G for GloVE embeddings, and FT for FastText embeddings. The blanks represent the training datasets. The Average (Avg) does not include the results of the dataset(s) used for training. 69
3.3. Results using *all* features on the datasets with attack (A) and support (S) relations. RF stands for Random Forests, A for autoencoder model, C for concatenation model, M for mix model, G for GloVE embeddings, and FT for FastText embeddings. The blanks represent the training datasets. The Average (Avg) does not include the results of the dataset(s) used for training. 70

4.1. Example of instances in the web dataset [37]. 75
4.2. Example of instances in the nk dataset [128]. 76
4.3. Example of instances in the com dataset [21]. 77
4.4. Example of instances in the ukp dataset [180]. 77
4.5. The split of the selected datasets into the three types of relations. . 77
4.6. Overview of features used in determining relations between pairs of sentences. 80
4.7. Baseline results using 5 fold CV. 81
4.8. Results using 5x5 fold CV. 81
4.9. The relations (*a* for *attack* and *s* for *support*) and the number of times they have been indicated by the annotators for the "merged" version. 84
4.10. The relations (*a* for *attack* and *s* for *support*) and the number of times they have been indicated by the annotators for the "unmerged" version. 86

4.11. The relations (*a* for *attack* and *s* for *support*) and the number of times they have been indicated by the annotators. 91

5.1. Errors between strength and Rotten Tomatoes score for SA and RbAM. 108
5.2. Classification report using SA (682 Fresh and 306 Rotten movies) and RbAM (620 Fresh and 267 Rotten movies). The difference in the number of movies considered is attributed to filtering, using different methods, of 'neutral' relations in SA and RbAM respectively, which results in a movie being discarded if it had not received votes from at least 33% critics. 109
5.3. Examples of movies' extracted themes. 110
5.4. Reviews for the movie *Inception*. 112
5.5. Votes (and their polarity) extracted from the reviews in Table 5.4. 114
5.6. Features and the associated category. 128
5.7. Classifiers' performances on the hotel dataset using 5 fold cross-validation. 129
5.8. Classifiers' performances on the restaurant dataset using 5 fold cross-validation. 130
5.9. Classifier performance on the hotel dataset using 5 fold cross-validation on subsets of the dataset. 131
5.10. Reviews for a hotel in Chicago. 133
5.11. Arguments extracted from the reviews in Table 5.10. 135

6.1. 5-fold cross-validation results 150
6.2. Average of 5 runs of training on a reduced dataset and testing on the remaining examples. 152
6.3. Examples of word relatedness. 156
6.4. Experimental results with various parameters: **#F**(eatures), **S**(ynsets), **D**(efault) chosen, the use of **C**(lusters), **U**(pgraded) version of AA-CBR, obtained using 5 fold CV. 159
6.5. Experimental results with various parameters: **#F**(eatures), **S**(ynsets), **D**(efault) chosen, the use of **C**(lusters), **U**(pgraded) version of AA-CBR, obtained using 5 fold CV. 159

B.1. Results using 5x5 fold CV. 204

List of Figures

1.1. Example AAF represented as a graph (edges labelled as $-$ represent disagreement/attack). 16
1.2. Example BAF represented as a graph (edges labelled as $+$ represent agreement/support). 16

2.1. CBOW and Skip-Gram models [131]. 28
2.2. An overview of how RNN is unrolled. 29
2.3. LSTM overview. 30
2.4. GRU overview. 31
2.5. AAF extracted from texts. 39
2.6. $(Args_N, \mathcal{L}_N^-)$ for Example 2.3.2. 42
2.7. $(Args_N, \mathcal{L}_N^-)$ for Example 2.3.2 but with $(N, ?) = (\{\mathsf{a}, \mathsf{c}\}, ?)$. . . 44
2.8. BAF extracted from texts. 45
2.9. Example BAF. 45

3.1. General Deep Learning (DL) architecture. P_{attack} stands for the probability of *attack* class and $P_{support}$ stands for the probability of *support* class. The output is given by the class with the highest probability. 54
3.2. An example of microtext and the associated argumentation graph. 56
3.3. Concatenation instance of DL architecture. 64
3.4. Mix instance of DL architecture. 65
3.5. Autoencoder instance of DL architecture. 66
3.6. Attention instance of DL architecture. 67

4.1. DL architecture for 3 class RbAM. P_{attack} stands for the probability of *attack* class, $P_{neither}$ stands for the probability of *neither support nor attack* class, and $P_{support}$ stands for the probability of *support* class. 74
4.2. The LSTM-based architecture. 79

4.3. The LSTM-based architecture with attention. 79
4.4. Our manually extracted BAF from the flat example. 83
4.5. Argumentative relations and the number of times these have been annotated for the "merged" version. 85
4.6. Recall of ML classifiers we used to determine argumentative relations in the "merged" version of the factual text. Threshold *t* represents the number of times the relations have been identified by at least *t* annotators. 85
4.7. Argumentative relations and the number of times these have been annotated for the "unmerged" version. 87
4.8. Recall of ML classifiers we used to determine argumentative relations in the "unmerged" version of the factual text. Threshold *t* represents the number of times the relations have been identified by at least *t* annotators. 88
4.9. Our manually extracted BAF from the excerpt dialogue. 90
4.10. Argumentative relations and the number of times these have been annotated. 93
4.11. Recall of ML classifiers we used to determine argumentative relations. Threshold *t* represents the number of times the relations have been identified by at least *t* annotators. 93

5.1. An overview of ADA for extracting aggregated scores from (snippets from) reviews, compared with the original RT method of computing a RT score (the Tomato Score - TS in the figure) as an aggregation method. 99
5.2. An overview of ADA for extracting aggregated scores with dialogical explanations from (snippets from) reviews. TS (Tomato Score) in the figure represents the RT score. We add Dialogical Explanations to Figure 5.1. 111
5.3. QBAF for the movie *Inception* with base scores (normal font) and DF-QuAD strengths (bold font). 115
5.4. Functions $r_a^+, r_a^-, r_b^+, r_b^-$, for any $\gamma \in Args$ 115
5.5. QBAF for the movie *Phantom Thread* with base scores (normal font) and DF-QuAD strengths (bold font). 117
5.6. QBAF for the movie *The Post* with base scores (normal font) and DF-QuAD strengths (bold font). 118

5.7. Example BAF. Here, b supports a (indicated as a + edge) and c attacks b (indicated as a - edge). 119
5.8. Overview of how RbAM and argumentative (arg) reasoning and features are used for detecting deceptive reviews. 120
5.9. Topic-dependent BAF obtained from $\mathbf{R_1}$, $\mathbf{R_2}$ in Section 5.3.1. . . 126
5.10. BAF obtained from removing (arguments from) $\mathbf{R_1}$. 127
5.11. BAF obtained from the reviews in Table 5.10. 136

6.1. An overview of how AA-CBR is used for classification. The AAF is used to predict the oucome of (any) M via its trimmed version N. 144
6.2. Autoencoder architecture: $|\mathbb{F}_{\mathcal{L}}|$ binary features are used to train the autoencoder to obtain a **code** (hidden layer **h** of size $H < |\mathbb{F}_{\mathcal{L}}|$) that best captures the input $|\mathbb{F}_{\mathcal{L}}|$ features. 149
6.3. Example of AAF for categorical data. 153
6.4. First example of AAF for Sentiment Analysis. 160
6.5. Second example of AAF for Sentiment Analysis. 162

1. Introduction

"Argumentation is the action or process of reasoning systematically in support of an idea, action, or theory" as defined in the Oxford English dictionary. Put simply, it is the process by which we try to prove or impose our opinions and attempt to convince others to accept them. We, as humans, do this on a regular basis: we justify a statement, trying to convince others that we are right by giving (logical) justification for the statement we want to support or combat. The opposite is true as well, argumentation occurring when *somebody else* (a person, an advertisement, a review) tries to convince us of the truth, the utility, the beauty of an idea, product or decision. Finally, argumentation also occurs when, while expressing an opinion, we are asked the question "*Why* (is that the case)?".

Argumentation as overviewed in [95] is a well established research area within Artificial Intelligence (AI) concerned with developing formal, logic-based accounts of arguments [55, 104, 135]. Argumentation has been shown to be able to deal with incomplete and conflicting information (see [7] for a recent survey). Within AI, Argumentation encompasses the definition of arguments and of relations between these arguments within Argumentation Frameworks (AFs) that can be analysed and evaluated using various semantics. In this thesis we focus on two types of AFs: Abstract Argumentation Frameworks (AAFs) [55] and Bipolar Argumentation Frameworks (BAFs) [40]. In AAFs, each argument is represented as an entity and we have a single type of relation, *attack*, between arguments. Consider the following dialogue:

a: We should go see this movie.

b: All reviews for this movie are bad.

c: All our friends said it was good.

Figure 1.1 shows the AAF obtained from these opinions in the dialogue in which argument a is attacked by argument b (intuitively, this means that b disagrees with a) and argument b is attacked by argument c. The conflicts that arise in these AAFs can be resolved using qualitative semantics based on membership criteria [55]. One such semantics amounts to *conflict-freeness*. In our example, the set

$\{a, c\}$ is conflict-free whereas $\{a, b\}$ is not, since b attacks a. Other, more complex, semantics are defined based on the notion of *defence* which states that an argument defends another argument by attacking all its attackers. In our example, c defends a because it attacks b (the only attacker of a). Conflict-freeness can be used in applications where conflicts lead to determining the outcome, such as detecting fake/deceptive texts, whereas semantics based on the notion of *defence* can be used in debates.

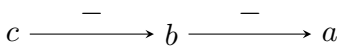


Figure 1.1.: Example AAF represented as a graph (edges labelled as $-$ represent disagreement/attack).

In BAFs, each argument is again represented as an entity but, in addition to the *attack* relation from AAFs, we also have a *support* relation between arguments. Consider again the dialogue that led to the AAF in Figure 1.1 along with the following additional opinion:

d: The movie was nominated for many awards.

Figure 1.2 shows the BAF obtained from these texts in which argument a is supported by argument d (intuitively, this means d is an argument agreeing with argument a). Different semantics have been defined to accommodate the support relation [20, 40, 39], in the same spirit as the semantics for AAFs.

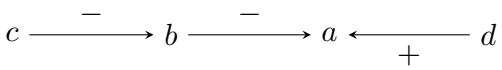


Figure 1.2.: Example BAF represented as a graph (edges labelled as $+$ represent agreement/support).

Alternatively, for both AAFs and BAFs, quantitative gradual semantics have been defined in which arguments are assigned a score, starting from a given base score within a $[0, 1]$ range [12, 162]. Assuming all arguments have a base score of 0.5, then argument a from Figure 1.1 has, using the semantics defined in [162], a score of 0.375 whereas argument a from Figure 1.2 has a score of 0.625 (the increased strength is due to the supporting argument).

Argumentation has proven successful in several domains, including multi-agent systems [29], and decision support in medicine [69] and engineering [12]. These

applications focus on semantics and require engineering of the AFs rather than obtaining the AFs from data.

With the rise of online media and with the availability of large amounts of information, these AFs could be extracted from text. The Web and social media represented by online newspapers, debate platforms, and social networks are abundant in text that can be analysed, in particular arguments that can be extracted from these texts. The availability of such textual data, together with advances in Natural Language Processing (NLP) and Machine Learning (ML), especially Deep Learning (DL), has recently led to the rise of a new research area in AI called Argument Mining (AM). AM can be seen as being at the intersection of NLP by means of ML techniques and Argumentation. From the seminal works in this field [141, 185] and the very first events dedicated to the study of computational models of argument in 2014, The First Workshop on Argumentation Mining[1] and Frontiers and Connections between Argumentation Theory and Natural Language Processing[2], AM is now a prominent research area at AI and NLP conferences. The successes emerging from this field such as ARG-Tech's analysis of BBC's Moral Maze[3] and IBM's Project Debater[4] have been widely covered and promoted in the news[5,6].

AM is "the general task of analyzing discourse on the pragmatics level and applying a certain argumentation theory to model and automatically analyze the data at hand" [83]. AM is a complex task due to the lack of a clear argumentative structure in free natural language text. From an application point of view, AM can be seen as an advancement of Sentiment Analysis and Opinion Mining [84, 117]. Whilst the goal of Opinion Mining is to identify *what* users think about X, the goal of Argument Mining is to understand the reason *why* users think about X the way they think, not limiting to opinions and their sentiment polarity. For example, "I like the movie" is not an argument whereas "I like the movie, the plot was great" is an argument as it gives a reason as to why the speaker liked the movie.

Two tasks are crucial in AM [31, 118, 148]: *1)* argument component detection aiming at the identification of arguments within the input natural language text (i.e., claim, premises, and their textual boundaries); and *2)* relation prediction aiming at the prediction of the relations between the argumentative components

[1] http://acl2014.org/acl2014/W14-21/index.html
[2] http://www-sop.inria.fr/members/Serena.Villata/BiCi2014/frontiersARG-NLP.html
[3] www.arg-tech.org/index.php/argument-mining-deployed-in-bbc-app/
[4] www.research.ibm.com/artificial-intelligence/project-debater/
[5] www.bbc.co.uk/news/technology-41010848
[6] www.wired.com/story/now-the-computer-can-argue-with-you/

identified in the first stage. For example, from the text *We should not go to the park as the weather forecast announced rain* we extract the claim *we should not go to the park* and the premise *weather forecast announced rain* where the premise *supports* the claim.

One challenge that arises in AM is that even humans have difficulties from distinguishing (isolated) argumentative phrases from simple statements [37, 171]. Rather than using a two-stage pipeline (i.e. identifying components and then identifying relations), we can instead determine relations independently of the type of components. Relation-based Argument Mining (RbAM) [37] addresses this problem by identifying an argumentative relation of attack or support between two texts, considering both texts to be argumentative, irrespectively of their stand-alone argumentativeness. This way, we can identify phrases that may be considered statements when analysed in isolation but that are in an argumentative relation with another phrase, hence get the status of argument when read in context. For example, in RbAM we directly identify a *support* relation between the following two sentences without previously identifying them as arguments: *Think of how many other things you can buy with the money you spend on this smart watch* and *No plan to buy, price too high*. It is important to note that RbAM does not rely on or assume any specific argument model or internal structure of arguments.

Despite the amount of approaches [31] tackling the relation prediction task in AM with satisfying results, a problem arises: these solutions heavily rely on the peculiar features of the dataset taken into account for the experimental setting and are hardly **portable** from one application domain to another. This is because the relation prediction task in AM can be seen as a stand-alone classification task that is addressed by ML techniques [23, 38], and, similarly to ML tasks, the models developed for RbAM are trained and tested on data from the same, or similar, domains. Various existing domains have been used for predicting argumentative relations: education (in particular student essays [178] and short argumentative texts [149]), legal [145, 184], politics [128], and web-based content such as social media and news articles [21, 23, 37, 38]. The solutions proposed are domain-dependent, with results being reported either on topics selected from a dataset consisting of several topics [9, 33, 180] or on datasets similar to the ones the models have been trained on [178]. The difficulty in portability across domains gives rise to another issue: whether AM can successfully be applied in real world applications. As there are no available datasets that have AM as a component, we rely on other tasks with labelled data to test the portability of AM.

1.1. Thesis Contributions

Our research has arisen from the following problems and questions in the field: *1)* whether we can overcome the current problem of RbAM solutions that heavily rely on datasets which makes them difficult to transfer to other domains as well as to compare amongst proposed techniques, *2)* whether AFs can be mined from text using RbAM or other methods, and *3)* whether these mined AFs can be efficiently used in real world settings from which users would benefit.

To address these questions, we make three substantial contributions.

1.1.1. Portability of RbAM

Our focus is on RbAM so that, given a pair of texts, we are interested in determining the relation between the first, called *child* argument, and the second, called *parent* argument, by means of a neural network. We identify two realisations of RbAM, each with its own advantages and applications.

The first realisation is treating RbAM as a 2 class problem (determining a class amongst *attack* and *support*) to tackle the issue of portability in RbAM by proposing a set of strong neural baselines for the RbAM community to compare more effectively how well a method performs on this task. As the majority of the datasets containing argumentative relations target only two types of relations, attack and support, we define neural models to address the 2 class RbAM problem, analysing all available datasets for this task. Each of the two texts is represented using embeddings as well as other features. We propose four neural network architectures for the classification task, two concerned with the way child and parent arguments are passed through the network (*concatenation* and *mix* models), an *autoencoder*, and an *attention*-based model. Our baselines are shown to perform homogeneously over all the datasets proposed in the literature for the relation prediction task in AM, oppositely to what is achieved by the single methods proposed in the literature. The contribution of our proposal is to bestow the AM community with a set of strong cross-dataset baselines to compare with in order to demonstrate how well a relation prediction method performs.

The second realisation is treating RbAM as a 3 class problem (determining a class amongst *attack*, *support*, and *neither attack nor support*) thus developing a model that determines the argumentative relation, if any, between two texts that can be used in real world applications. We consider three classes as, deployed in the wild, the model must first discard the neither support nor attack relations. We apply

our model on two linguistic sources, a factual text and an excerpt dialogue, and show it is comparable to human annotators. Having a model which is comparable to annotators can lead to the automatic creation of (AM) corpora, highly needed in experimentation.

1.1.2. Portability of RbAM: from downstream task to mining BAFs for applications

In addition to proposing efficient models that address the 3 class RbAM task, we consider the application of RbAM in the online setting in which RbAM can be seen as a prerequisite step for constructing BAFs. We narrow the online setting as users are increasingly turning to aggregation sites that give an overview of reviews such as *Rotten Tomatoes*[7] (RT), an aggregation website of trusted critic reviews for movies, *Amazon*[8], an aggregation website of user reviews for products, and *TripAdvisor*[9], an aggregation website of user reviews for hotels and restaurants.

We propose the use of BAFs extracted from textual data by means of RbAM with quantitative gradual semantics to provide the backbone for explanations. As the aggregation of online reviews is fast becoming the chosen method of quality control for users in various domains, from retail to entertainment, fair, thorough and explainable aggregation of reviews is increasingly sought-after. The explanations drawn from mined BAFs, which can be seen as summaries of textual data, overcome the simplicity that comes with the aggregation of reviews which, in the movie domain for example, it has been said to be "hacking off critical nuance"[10].

Within the movie review domain, and in particular RT's aggregation method that amounts to the percentage of critics giving the movie a positive review, we show how NLP and quantitative gradual semantics together can be used as a novel approach to remedy RT's lack of reasoning and explainability. We define an argumentative dialogical agent (ADA) that extracts a BAF for any chosen movie and provides a dialectical strength which is comparable with the RT score while at the same time being able to provide dialogical explanations of why a movie obtained its strength via interactions between the user and ADA.

Within the hotel/restaurant review domain, we show that our method for extracting BAFs contributes a novel argumentative feature that, when used in combina-

[7] www.rottentomatoes.com
[8] www.amazon.co.uk/
[9] www.tripadvisor.co.uk/
[10] www.nytimes.com/2017/09/07/business/media/rotten-tomatoes-box-office.html

tion with other features in standard supervised classifiers, outperforms the latter in the task of detecting whether hotel/restaurant reviews are deceptive. Our topic-dependent BAF relies on using topics to identify pairs of chunks of text to be classified using 3 class RbAM, along a temporal dimension whereby more recent chunks of text may either support or attack less recent ones, but not vice versa. This choice is well suited in settings such as online reviews, where information is provided incrementally, over time, and contributors have full vision of existing data. The BAFs extracted are used in explaining the reasoning within the reviews as well as providing strengths of aspects identified in reviews.

1.1.3. Portability of method that empowers explanations: from reasoning to mining AAFs

Our last contribution makes use of AA-CBR [48, 49], an Abstract Argumentation (AA)-driven Case-Based Reasoning (CBR) model, originally designed for the task of deciding and explaining outcomes of new cases based on past cases, to mine AAFs from data and to use them for classification while also providing explanations. Explaining ML algorithms is a hot topic nowadays. Solutions have been proposed to explain ML classifications using model-agnostic methods [166, 167], to explain black box models [82], to highlight the words that contribute the most to the NLP classification task [6, 106] to name a few. Argumentation has also been used to generate explanations in AI, with a focus on explaining decisions [3], classifications [4], recommendations [28], as well as (non-)acceptance of arguments [63, 64, 74].

We port the explanations of AA-CBR to explanations of classification of categorical data, and then to classification of text sentiment by proposing an upgraded version of AA-CBR to accommodate the richness of text. Our focus is on binary sentiment classification of textual data. We cluster semantically related terms from the input texts to reduce the feature space while keeping the semantics unchanged. We rank the feature importance in classification with an ensemble technique and select a subset of features with the highest scores. We then customise the approach of [48] as well as a variant thereof to map the training examples (represented through the selected features) and a default argument (for a chosen default classification) onto AAFs. We illustrate two types of explanations that can be extracted from AAFs, counterfactual and dialectical, and show that our method is competitive in predicting when compared to Decision Trees, which are also ex-

plainable (but by means of rules).

1.2. Thesis Structure

This thesis is structured as follows. Chapter 2 gives the necessary background in NLP and an overview of the literature for RbAM and types of AFs. In Chapter 3, we propose different neural architectures to address the 2 class RbAM problem that the (Rb)AM community can then use to test whether models are portable across domains and that the models do not perform well only on the datasets they had been trained on. In Chapter 4, we propose a neural architecture to address the 3 class RbAM problem and use it to compare its performance to annotators on two linguistic sources. In Chapter 5, we deploy the 3 class RbAM model "in the wild" to benefit applications and to aid mining BAFs that provide the backbone of explanations in various settings. In Chapter 6, we describe a completely different approach obtaining AAFs rather than BAFs and using qualitative rather than gradual semantics. This approach ports AA-CBR to NLP binary classification tasks. Finally, in Chapter 7, we conclude with a discussion of the contributions of the thesis as well as a look ahead to future work.

1.3. Publications

The work presented in this thesis has resulted in several publications.

Chapter 4 builds upon the following:

- *Identifying attack and support argumentative relations using deep learning*
  Oana Cocarascu, Francesca Toni
  Conference on Empirical Methods in Natural Language Processing (EMNLP) 2017
- *Mining argumentative relations "in the wild"*
  Oana Cocarascu, Francesca Toni
  The 9th Conference of the International Society for the Study of Argumentation (ISSA) 2018

Chapter 5 builds upon the following:

- *Detecting deceptive reviews using Argumentation*
  Oana Cocarascu, Francesca Toni

The International Workshop on AI for Privacy and Security, PrAISe@ECAI 2016

- *A System for Supporting the Detection of Deceptive Reviews Using Argument Mining*
  Oana Cocarascu, Francesca Toni
  The 6th International Conference on Computational Models of Argument (COMMA) 2016
- *Mining Bipolar Argumentation Frameworks from natural language text*
  Oana Cocarascu, Francesca Toni
  Computational Models of Natural Argument, The 17th Workshop on Computational Models of Natural Argument (CMNA@ICAIL) 2017
- *Combining deep learning and argumentative reasoning for the analysis of social media textual content using small datasets*
  Oana Cocarascu, Francesca Toni
  Computational Linguistics 2018
- *Extracting Dialogical Explanations for Review Aggregations with Argumentative Dialogical Agents*
  Oana Cocarascu, Antonio Rago, Francesca Toni
  The International Conference on Autonomous Agents and Multiagent Systems (AAMAS) 2019

Chapter 6 builds upon the following:

- *Explanatory Predictions with Artificial Neural Networks and Argumentation*
  Oana Cocarascu, Kristijonas Cyras, Francesca Toni
  Workshop on Explainable Artificial Intelligence (XAI@IJCAI- ECAI) 2018

2. Background and Literature review

This chapter gives the background and a review of the literature most relevant to this thesis. In Section 2.1, we give the core Natural Language Processing (NLP) background required in text processing and an overview of the Deep Learning (DL) classifiers used to obtain models for various classification tasks in the rest of the thesis. We present background and related work for Argument Mining (AM) in general and Relation-based Argument Mining (RbAM) in particular in Section 2.2. We cover the different Argumentation Frameworks (AFs) used throughout this thesis in Section 2.3. Section 2.4 gives an overview of methods that address Explainable AI, a task that we also touch upon in this thesis. Section 2.5 provides a summary of this chapter with a focus on the following chapters.

2.1. Core Natural Language Processing and Deep Learning background

In this section we give the relevant background in NLP and an overview of DL models used in the thesis.

2.1.1. Natural Language Processing - Preprocessing

Preprocessing is a critical step in any NLP task. Preprocessing represents the task of preparing the raw text corpus for analysis. It consists of a number of steps needed to transfer the text from human language to machine-readable format for further processing. Each of these steps may or may not be applied depending on the task, but generally fall under the following broad categories:

- tokenization (including stop words removal and lowercase transformation)
- normalization (lemmatization and stemming)
- Part-of-Speech (POS) tagging

2.1.1.1. Tokenization

Tokenization represents the process of dividing a text into a sequence of tokens such as words, sentences, or any other meaningful elements. Generally, tokens roughly correspond to words as in English words are separated by white spaces. In this step, stop words and punctuation marks alongside tabs and line breaks are also removed.

Stop words are commonly used terms such as pronouns, prepositions, conjunctions, quantifiers, determiners, conjugations of the verbs "to be" and "to do", that do not contribute to the context but are just connecting parts of a sentence. Usually, these stop words are removed as this process reduces the textual data and improves the system performance which is useful for some tasks such as determining the topics found in documents.

There are several challenges in tokenization. One challenge concerns the abbreviations/acronyms and text chunking which extracts phrases from unstructured text. For example, there might be several ways of writing the same phrase: *state-of-the-art* and *state of the art* that we would like to match. If we consider *New York*, a tokenizer which operates on white spaces only would identify two tokens, *New* and *York*. An occurrence of *NY* in another text would lead to the identification of the token *NY*. In this case, we would like *New York* and *NY* to correspond to the same token and hence text chunking is required. Another challenge is related to the division of text into words and sentences. Sentences end with punctuation, however some punctuation marks are ambiguous: the *semicolon* can either be a separator of a list of elements or a sentence separator and the *period* can be either a full-stop or an abbreviation (e.g. *m.p.h.*). Some of these tokens can be identified using regular expressions grammars.

Another preprocessing step is related to case folding that is reducing all letters to lower case. This way, the tokens "hello", "Hello", "HELLO" will be treated the same. However, this is not without problems as the distinction between *US* (United States) and the pronoun *us* needs to be accommodated.

2.1.1.2. Normalization

Different forms of the same word can be used in text such as "seeing", "sees", "saw", "seen". In text analysis, it is useful to reduce the inflected forms of a word to a common base form. There are two main approaches to achieve this, namely stemming and lemmatization.

Stemming is the process by which words are reduced to a root by removing inflection. There are several stemming models such as Porter [155] and Snowball [156]. Stemming can be seen as cutting suffixes (e.g. "automate", "automatic", "automation" are all reduced to "automat") with original words not necessarily related ("universe" and "university" are both reduced to "univers").

To overcome this issues, an alternative method to removing inflection can be used. Lemmatisation reduces words to their basic forms by using POS. Lemmatisation would reduce "seeing", "sees", "saw", "seen" to "see" which stemming would not be able to perform.

2.1.1.3. POS tagging

The meaning of sentence can be determined by analysing the POS of words which is useful in word sense disambiguation. Basic POS include tags for most common parts of speech (noun, verb, adjective, etc.). For better performance, the POS tags also include detail about the singular vs plural form of nouns, verb conjugations, tenses, voice, etc. The English taggers are based on the Penn Treebank tag set [126]. For example, in the following phrase "the cat sat on the mat", "the" is a determiner, "cat" and "mat" are nouns, "sat" is a verb, and "on" is a preposition.

These preprocessing steps help reduce the indexing size by removing stop words and performing normalization.

2.1.2. Word vectors

In order to process natural language we need a method for representing text. The typical representation is through word vectors where words from the vocabulary are mapped to vectors of real numbers. The text can be vectorized in various ways where the coefficient of each token can be i) binary (indicating the presence of the vocabulary word in the analysed text), ii) the index of the token in the dictionary, iii) based on word count, iv) based on term frequency-inverse document frequency.

We focus on two types of word vectors: standard word vectors and neural embeddings.

2.1.2.1. Standard word vectors

The standard word vector representations are: Bag-of-Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) and distributional vectors.

BoW is a vector that highlights the presence of words (or n-grams) in text. Every instance has the same number of features, one for each word in the vocabulary of the training dataset. The vector marks the value at index i with 1 if the occurence of the word at index i in vocabulary appears in the text, and with a 0 otherwise.

One drawback of this representation is that it does not encode any information with respect to the meaning of the word. Another downside is that this approach yields sparse vectors as the majority of words (or n-grams) will be set to 0. In BoW, features are weighted evenly, independently of how frequently they appear in text.

TF-IDF improves on BoW representation by measuring how important a word (or n-gram) is to a document relative to the corpus. It assigns a weight to each word based on how often they occur in a document, but offsets the weight depending on the number of documents from the corpus that contain that word. Formally, TF-IDF for term *t* (where *d* is a document) is:

$$\text{TF-IDF}_t = \text{TF} \cdot \text{IDF} \tag{2.1}$$

where

$$\text{TF} = \frac{\text{Number of times } t \text{ appears in } d}{\text{Total number of terms in } d} \tag{2.2}$$

$$\text{IDF} = log\frac{\text{Total number of documents}}{\text{Number of documents that contain } t} \tag{2.3}$$

This adjusts to the fact that some words appear more frequently than others that are not necessarily relevant for text analysis.

Distributional vectors [10] incorporate information about the context of a word. These vectors can be represented as a global co-occurrence frequency within a window. This is based on the idea that words that occur in similar contexts have similar meaning. In a distributional embedding matrix, each row encodes distributional context based on the number of words it co-occurs with. Distributional embeddings precede word embeddings.

2.1.2.2. Neural embeddings

Pre-trained word representations have become an essential part in many NLP and Machine Learning tasks. Used to initialize the first layer of a neural network, a word embedding is a dense feature in a low dimensional space. The neural

embeddings that are the most commonly used and that we focus on are Word2Vec, Global Vectors for Word Representation (GloVE) and FastText.

Word2Vec [130] is a predictive embedding model. There are two architectures: the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. The architectures can be seen in Figure 2.1. CBOW predicts target words ($W(t)$) from context words ($\{W(t-2), W(t-1), W(t+1), W(t+2)\}$) whereas Skip-Gram predicts the context words ($\{W(t-2), W(t-1), W(t+1), W(t+2)\}$) from a given word ($W(t)$). In Continuous Skip-Gram, nearby context words have a higher weight than more distant context words.

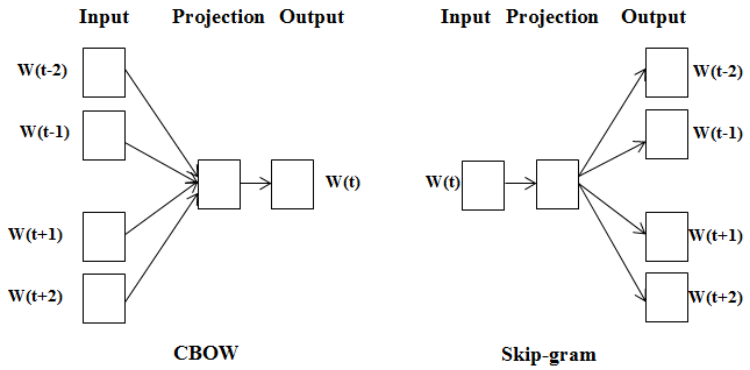


Figure 2.1.: CBOW and Skip-Gram models [131].

GloVE embeddings [151] make use of the global context unlike the two models of Word2Vec that take only local context into account. The two Word2Vec models are window-based models that do not consider the co-occurrence statistics of the corpus. With GloVE, a co-occurrence matrix is constructed using a fixed window size (i.e. words co-occur if they appear together in the window size). Then it makes use of the aggregated global co-occurrence matrix and represents word vectors by decomposing this matrix using neural models. The word vectors created capture meaning in vector space (i.e “king” is to “queen” what “man” is to “woman”).

FastText [98, 129] improves Word2Vec by learning vector representations for words and the n-grams (substrings of characters) found within each word. The overall embedding is hence the sum of the n-grams. The main advantage over the other types of embeddings is that FastText is able to provide an embedding for rare and out-of-corpus words by analysing the chunks the word can be broken down into.

The advantage of these word embeddings is that we can use them directly for

the task we are interested in (i.e. the downstream task) without needing the model itself that was used to train these vectors.

2.1.3. Deep learning classifiers

In this section we describe the most widely used models in Deep Learning for NLP: Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) models and Gated Recurrent Unit (GRU) models.

These models are able to achieve state-of-the-art performance in various (textual-based) machine learning tasks (e.g. machine translation, sentiment analysis, etc.). Their ability is to process sequential data, which is the case of text. In order to understand a sentence, we process each word along with what we learnt from the previous words. RNNs use previously learnt information to predict the next output word.

RNNs [60, 132] are a type of neural networks in which some hidden layer is connected to itself so that the previous hidden state can be used along with the input at the current step. Consider the diagram from Figure 2.2. A part of the neural network, A, takes an input x_t and outputs h_t. The loop allows information to be passed from one step of the network to the next. Thus, RNNs use their internal state to process sequences of inputs (i.e. words) making use of context to determine the output based on input.

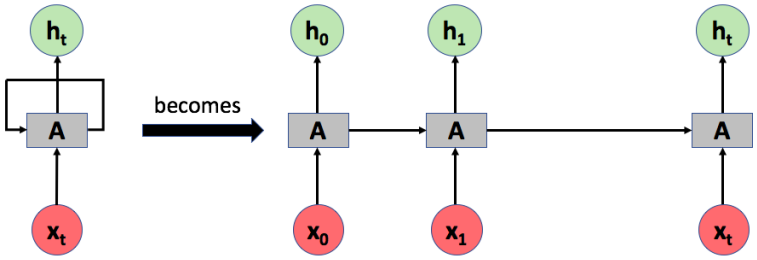


Figure 2.2.: An overview of how RNN is unrolled.

However RNNs tend to suffer from the vanishing gradients problem [14] while trying to capture long-term dependencies. Long Short-Term Memory (LSTM) models [91] and Gated Recurrent Unit (GRU) [44] models address this problem through gating mechanisms.

LSTM models [91] address this problem by introducing memory cells and gates into networks. LSTM models are a type of RNNs which use memory cells to store contextual information and three types of gates (input i_t, forget f_t, and output o_t

gates) that determine which information needs to be added or removed in order to learn long term dependencies within a sequence. The LSTM is shown in Figure 2.3 and the equations can be seen below where σ represents sigmoid activation, $*$ elementwise multiplication, $\tilde{C}_t$ the "candidate" hidden state, C_t the internal memory, and W and U parameter matrices:

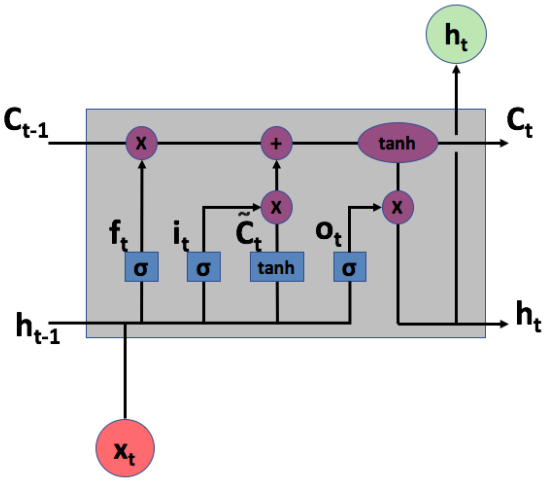


Figure 2.3.: LSTM overview.

$$
\begin{aligned}
i_t &= \sigma\left(x_t U^i + h_{t-1} W^i\right) \\
f_t &= \sigma\left(x_t U^f + h_{t-1} W^f\right) \\
o_t &= \sigma\left(x_t U^o + h_{t-1} W^o\right) \\
\tilde{C}_t &= \tanh\left(x_t U^c + h_{t-1} W^c\right) \\
C_t &= \sigma\left(f_t * C_{t-1} + i_t * \tilde{C}_t\right) \\
h_t &= \tanh(C_t) * o_t
\end{aligned}
\tag{2.4}
$$

$\tilde{C}_t$ is computed based on the current input and the previous hidden state. C_t represents the combination of the previous memory multiplied by the forget gate and the new hidden state multiplied by the input gate. h_t is computed by multiplying the memory with the output gate.

One problem with RNNs/LSTM models in NLP is that they do not make use of the information of future words. BiRNNs/BiLSTMs [174] solve this problem by using both previous and future words. This neural model processes the input sequence with two RNNs: one in the forward and one in the backward direction, resulting in two vectors for each input.

The **GRU** model [44] is a simpler variant of the LSTM model. It does not have a cell state C_t and has two gates, a reset gate r_t and an update gate z_t, compared to three gates as found in LSTMs. The forget and input gates from LSTM are combined into a single update gate. Compared to LSTMs, GRUs have fewer parameters and are easier to train. The GRU is showed in Figure 2.4 and the equations can be seen below:

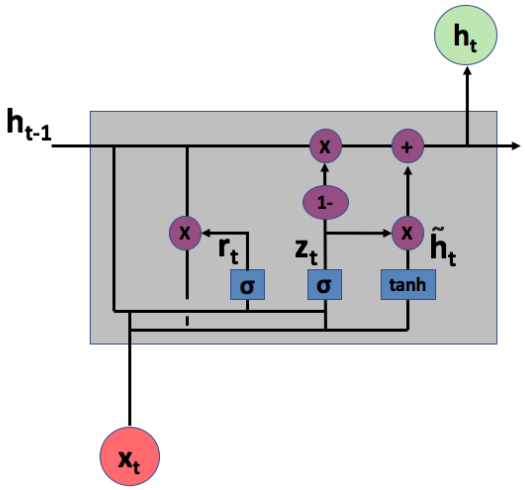


Figure 2.4.: GRU overview.

$$
\begin{aligned}
z_t &= \sigma\big(x_t U^z + h_{t-1} W^z\big) \\
r_t &= \sigma\big(x_t U^r + h_{t-1} W^r\big) \\
\tilde{h}_t &= \tanh\big(x_t U^h + (r_t * h_{t-1}) W^h\big) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
\tag{2.5}
$$

The update gate represents how much of the previous memory to use whereas the reset gate represents a combination of the new input with the previous memory.

In contrast to RNNs, LSTMs and GRUs keep the state from previous activations rather than replacing them entirely. This way, LSTMs and GRUs are able to remember words for a longer time, addressing the vanishing gradient problem. Using the gating mechanism, the two types of networks control the amount of previous state used in generating the next (output) word.

2.2. Argument Mining

Argument Mining (AM) [181] consists of mainly two tasks [31, 118, 148]: argument component detection aiming at the identification of arguments within the input natural language text (i.e., claim, premises, and their textual boundaries) and relation prediction aiming at the prediction of the relations (i.e. support and attack) between the argumentative components identified in the first stage. At a more detailed level, there are several stages in AM: identifying argumentative sentences, detecting component boundaries, determining argument components (e.g. claim and premise), and determining relations between arguments. With the recent success of Deep Learning (DL) in NLP tasks, neural end-to-end approaches have also been proposed for AM. Identifying argumentative components as well as the relations between these, has been done, for example, using LSTMs [59, 138, 157].

Most studies identify arguments at sentence-level. There are several issues with this approach given that in free text it might be the case that: a part of the sentence represents an argumentative element, two or more argumentative elements can be contained within the same sentence, an argumentative element can span across multiple sentences. However, most works assume only one of these cases.

In this thesis, we focus on Relation-based AM (RbAM) [37], that is identifying argumentative relations irrespectively of the argumentativeness of the texts tested. Our arguments are identified at sentence level, thus one sentence can be composed of one or several arguments. While we do not identify explicit argument components (i.e. claims and premises), we give here an overview of existing work that tackle this problem, focusing on tasks and methods.

2.2.1. Argument Mining component detection

The first task in argument mining component detection is determining argumentative sentences (i.e. building a supervised classifier to distinguish between argumentative text and non-argumentative text), followed by determining the argument components (e.g. claims and premises). In natural language however these elements can be implicit (i.e. enthymemes). For example, *We cannot trust this man, for he has perjured himself in the past* is an enthymeme as the premise *Those who perjure themselves cannot be trusted* is omitted.

The first types of texts analysed using AM were legal texts. Various classifiers were used on the European Court of Human Rights (ECHR) corpus, consisting of legal documents, to distinguish between claims and premises: Naïve Bayes,

SVM and Maximum Entropy [141]. Different argumentative structures have been identified: *Multiple Argumentation* where alternative arguments are given for another argument and *Compound Argumentation* where arguments reinforce each other and can be defined as *Subordinatively Compound Argumentation* where one argument supports another and *Coordinatively Compound Argumentation* where multiple arguments combined support another argument.

Detecting claims was the topic of study in [113, 119]. Whilst Levy et al. [113] identified claims depending on context, Lippi and Torroni [119] adopted a context-independent claim detection approach. Levy et al. [113] used Logistic Regression on a corpus consisting of Wikipedia articles to determine claims, whereas Lippi and Torroni [119] used SVM and partial tree kernel which can capture the rhetorical structure of a sentence and which was used to measure similarity between sentences. On a dataset of persuasive essays corpus, Stab et al. [179] focused on identifying argument components, claims and premises, comparing various classifiers, SVM, Naïve Bayes, Decision Trees and Random Forests. A two-step argument extraction was performed in [78] identifying sentences containing arguments (Logistic Regression, Random Forest, SVM, Naïve Bayes were compared) and using Conditional Random Fields to identify the fragments corresponding to claims and premises.

Identifying the argumentative sections in a text was studied in [199]. From a corpus about buying a camera, passages from the text were extracted based on predefined rules (premise and conclusion) which were analysed by an expert in order to construct an Argumentation Framework from these passages. This is similar in spirit to our work as we also mine Argumentation Frameworks from texts. Whilst in [199] the authors use only attacks, we use both attacks and supports between arguments.

We have presented existing work that target the component detection in AM and will present in the next Section the works that target the argumentative relations. Before this, we briefly present the neural end-to-end approaches proposed for AM [59, 138, 157] which aim to learn both components and relations between arguments. All have experimented with the persuasive essays dataset [178]. Eger et al. [59] presented a neural end-to-end model for AM framed as a token-based dependency parsing and as a token-based sequence tagging problem, Niculae et al. [138] used a factor graph parametrized with LSTM and structured SVM, and Potash et al. [157] proposed a pointer network of LSTM with attention.

2.2.2. Relation-based Argument Mining

Various types of relations between arguments have been studied in the literature from identifying whether a sentence supports or does not support a claim [179] or whether it is a justification of a claim [141], to a three-class classification problem (i.e. attack/support/neither) [37] and a more fine-grained classification problem (i.e. explicitly/implicitly attacking or supporting relation or neither) [21].

In this thesis we focus on RbAM by directly identifying argumentative relations between texts. We consider two forms of RbAM: 2 class RbAM, with the aim to identify support and attack relations, and 3 class RbAM, with the aim to identify support, attack and neither support nor attack relations. We will see the advantages of each instantiation later in the thesis. We provide here an overview of RbAM from the point of view of datasets used for this task and methods employed.

2.2.2.1. Datasets for Relation-based Argument Mining

The majority of datasets for RbAM consist of only two types of relations, namely support and attack. The datasets cover various domains: education and in particular student essays [178] and short argumentative texts [149], legal [145, 184], politics [128], and web-based content (e.g. social media, news articles) [21, 23, 37, 38]. Some datasets are balanced [9, 37, 180] with thousands of pairs of arguments. Some datasets consist only of hundreds of pairs of arguments [21, 33, 128, 149].

Determining relations between any two texts (called *child* and *parent*) can also be viewed as a three-class problem, where the labels are: *support*, *attack*, *neither support nor attack*. Carstens and Toni [37] introduced a dataset covering topics such as UKIP, opinions about movies, technology, and politics, consisting of 1301 attack relations, 1329 support relations, and 1973 neither attack nor support relations. This is the only balanced dataset of reasonable size (i.e. more than a thousand instances per class) that has pairs of arguments (rather than an argument and a topic) with the relation labelled explicitly as neither support nor attack. Other datasets have a "non-argumentative" type of relation [21, 128, 180]. Several of these RbAM datasets have been used in an experimental setting in this thesis, hence a more detailed analysis of the datasets (examples and statistics) will be given later.

There are other datasets that do no focus on attack and support relations but have similarities to the two main argumentative relations. The Argument Interchange

Format (AIF) [16, 43, 95, 164] provides a model where relations between arguments are illustrated as a directed graph using an ontology of constraints and rules. AIF provides two types of nodes: *information nodes* (I-nodes) representing claims and *scheme (application) nodes* (S-nodes) representing application of *schemes*. Scheme nodes can be divided into three categories: *rule of inference application (RA) nodes* representing an application of an inference schemes, *preference application (PA) nodes* representing an application of a preference scheme, and *conflict application (CA) nodes* representing an application of a conflict scheme. There are two types of edges: *data edges* which can be between an I-node and an S-node ('information/data' supplying edge) and *scheme edges* which can be between an S-node and an S-node ('warrant edge') or an I-node ('conclusion' edge). One of the shortcomings of AIF is that is can only handle monologic argumentation. AIF+ was introduced in [164] to overcome this problem. In AIF+ locutions are a subclass of I-nodes and are represented through L-nodes. Further argumentation schemes were introduced: *transition application (TA) nodes*, a subclass of RA-nodes, representing the inference between two L-nodes [164] and *illocutionary application (YA) nodes* which make explicit the relation between locutions and underlying arguments expressed by the locution and which can be between an L-node and an I-node [165]. *MA nodes* represent default rephrase and occur when a text simply uses different words. AIFdb is a corpus of argument maps which follows the structure defined by AIF [109].

The Stanford Natural Language Inference (SNLI) corpus contains 570k sentence pairs labelled as *entailment*, *contradiction*, or *neutral* [26]. These relations may seem to have some similarity with the relations of interest to RbAM, namely *support*, *attack*, or *neither (support nor attack)*, respectively. However, the type of sentence pairs found in this corpus is different from the types of texts we are interested in analysing in RbAM. To illustrate, an example of entailment pair in the SNLI corpus is as follows: *"A soccer game with multiple males playing"* and *"Some men are playing a sport"*. We are instead interested in dialectical relations, for example of support as between the following two texts: *"I believe that what UKIP is doing is vital for this country"* and *"It is because of UKIP that we are finally discussing the European question and about immigration and thank goodness for that"* [37].

2.2.2.2. Methods for Relation-based Argument Mining

Whilst most studies that focus on determining links between (components of) arguments assume that the arguments have already been identified, Carstens and Toni [37] take a different approach and focus on identifying directly the relations between any two pieces of text, irrespective of their stand-alone argumentativeness, being able to identify texts that may appear non-argumentative when analysed in isolation but become argumentative when read in context.

RbAM can be treated as a ML classification task with, generally, two classes: *support* and *attack*, or three classes: *support*, *attack*, and *neither support nor attack*. Several ML techniques have been deployed in the literature for RbAM: in [38], Random Forests [27] and SVMs [24] were used to determine the type of relations between texts extracted from news, whereas in [178], SVMs were used to classify support and attack relations in persuasive essays. SVM with radial kernel is the most commonly used algorithm in the works that target the relation prediction task [9, 21, 128, 138]. Different features have been used in different studies. Stab and Gurevych [178] used structural features such as number of preceding and following tokens in the covering sentence, number of components in paragraph, number of preceding and following components in paragraph, relative position of the argument component in paragraph, Stab et al. [180] used topic similarity features (as the *parent* argument is a topic), whereas Menini et al. [128] used the position of the topic and similarity with other related/unrelated pair from the dataset, keyword embeddings of topics from the dataset.

Three methods for AM were covered in [108]. Firstly, the relations between propositions were identified using discourse indicators. Based on the assumption that the arguments follow the conclusion (i.e. the conclusion is presented first), the argumentative structure was constructed by identifying changes in topic. Finally, Naïve Bayes was used to determine premises and conclusions. AM was also viewed as a five-class classification problem: A (explicitly attacking an argument), a (vaguely/implicitly attacking an argument), N (no use of argument), s (vaguely/implicitly supporting an argument), S (explicitly supporting an argument), and the classification task was addressed with SVMs [21].

RbAM has been applied in various domains. Menini et al. [128] identified *attack* and *support* relations in political speeches from the 1960 presidential campaign using SVMs. Further, [58] focused on the task of mining arguments from Twitter, distinguishing between opinions and factual arguments and identifying the source

of these arguments using Logistic Regression and Random Forests. Hua and Wang [92] focused on identifying supporting arguments in relevant documents given a claim about a controversial issue.

Other methods, different from ML techniques, have also been used to address the RbAM task. Determining whether a sentence represents support or attack of a given claim on a corpus focusing on the discourse structure of persuasive essays was studied in [179]. On the legal ECHR corpus, Palau and Moens [141] predicted the relations between argument components using context free grammars.

Bosc et al. [23] used a corpus consisting of tweets to determine *attack* and *support* relations between tweets. Using an encoder-decoder architecture with two LSTMs where the second LSTM is initialised with the last hidden state of the first LSTM, they obtain negative results (0.2 F_1 score for *support* and 0.16 F_1 score for *attack*). Gema et al. [77] used LSTM Siamese Network with cosine similarity and attention mechanism to determine support relation between two sentences.

Taking a new direction in this process, Cabrio and Villata [30, 32] used textual entailment to address the RbAM task. Textual entailment represents the task of identifying inference between two text fragments. It is a directional relation as one text fragment can be entailed from another but the reverse may not hold. Textual entailment was used to identify support and attack relations [30] on the Debatepedia dataset. Four attacks were introduced: supported, secondary, mediated and extended. The affirmative form of the title of the debate was set as the main argument and each opinion was extracted as an argument and was then associated with either the starting argument or the most recent argument the current argument referred to. Finally, the pairs were assigned an attack or support relation. Textual entailment was combined with Dung's framework in [32] to identify arguments. Firstly, the arguments were identified in the text and the relations between these arguments were determined using textual entailment. A directed graph was constructed to represent the argumentative discussion where the probability of entailment between a pair of arguments is inversely proportional to the distance between the constituent arguments. Some of the decisions taken in constructing the graphs are similar to the ones we adopt in this thesis (i.e. the temporal approach). Recognising textual entailment between two sentences was also addressed in [168], using LSTMs and word-by-word neural attention mechanism on the SNLI dataset.

Several types of DL architectures have been used in tasks where sentence pairs need to be classified, tasks different from the RbAM task, but that still share some similarities (i.e. either by considering pairs of texts or experimentation with

datasets similar to RbAM datasets). These include LSTMs [25, 26, 123], attentional LSTMs [102, 123, 168], which use a soft attention mechanism so that the representation of one piece of text depends on the representation of the other piece of text, and (attention-based) CNNs [86, 201]. Determining the relations in an SNLI sentence pair was addressed in [26], using stacked LSTMs with the bottom layer taking as input the concatenation of the premise and of the hypothesis, and in [25], using TreeLSTM models with shared parameters between the premise and the hypothesis. Habernal and Gurevych [86] experimented with both Bidirectional LSTMs (BiLSTM) and BiLSTM extended with an attention mechanism and a convolution layer over the input to determine the class which explains why a certain argument is more convincing than the other in the pair. Whilst they focus on determining convincingness, we focus on identifying *attack*, *support*, or *neither* relations between arguments.

Ferilli et al. [66] proposed a general summarisation framework based on Abstract Argumentation used to select sentences from text. This summarisation can be seen as extracting the most relevant information (arguments) from reviews. Poddar et al. [153] focused on identifying supporting opinions in reviews using a probabilistic graphical model. Both works are similar to our work. In this thesis, we mine Argumentation Frameworks from texts (e.g. reviews) and we focus on support and attack relations, Argumentation Frameworks that can be seen as summaries of reviews.

There are few works on AM "in the wild". For example, whilst Cerutti et al. [41] manually extracted a BAF from a online debate, Salah et al. [172] used Sentiment Analysis to extract graphs from parliamentary debates where nodes are labelled with the sentiment of the speaker and edges between nodes of same/different polarity are labelled as supporting/opposing.

2.3. Argumentation

From philosophers to linguists [187], Argumentation has been at the centre of much research, and, for the past 20 years, it has become a core field of study in AI [7, 13, 55, 135]. Within philosphy, the study of Argumentation is concerned with how assertions are proposed, discussed, and resolved when conflicting opinions are put forward. In day to day life, the arguments are often based on some background assumption shared by the parties involved, and thus are incomplete or vague.

Within AI, there are broadly two approaches in Argumentation: abstract and

structured. The former considers arguments as abstract entities, that can be instantiated as needed, and is concerned with the development of semantics used to evaluate various Argumentation Frameworks such as Abstract [55] (with one type of relation: attack) and Bipolar [40] (with two types of relations: attack and support). In addition, arguments can be equipped with base scores amounting to an intrinsic strength which can then be evaluated, using the strength of attacking and supporting arguments, to give the final dialectical strength. Structured argumentation considers arguments to have some sort of internal structure based on an underlying structured knowledge [15].

In this thesis, we focus on evaluating abstract arguments (that are seen as placeholders for arguments mined from text) using quantitative gradual semantics, and thus present in the following the types of frameworks of interest.

2.3.1. Abstract Argumentation Frameworks

Abstract Argumentation Frameworks (AAFs), introduced by Dung [55], are pairs consisting of a set of arguments, $Args$, and a binary relation, $\mathcal{L}^-$, between arguments $Args$, representing attacks. For $(\mathsf{a}, \mathsf{b}) \in \mathcal{L}^-$, then we say that a *attacks* b and that a is an *attacker of* b. All results defined in this section are from [55].

For example, consider the following three texts taken from the dataset introduced in [37]:

t: *"We should grant politicians immunity from prosecution"*
t': *"The ability to prosecute politicians is the ultimate protection against the abuse of power"*
t'': *"Prosecuting politicians is not what the country needs now"*

Here t' *attacks* t and t'' attacks t'. The AAF is $\langle Args, \mathcal{L}^- \rangle$ where $Args = \{t, t', t''\}$, $\mathcal{L}^- = \{(t', t), (t'', t')\}$. The AAF is given in Figure 2.5 as a graph with *attack* represented as $-$.

$$t'' \xrightarrow{-} t' \xrightarrow{-} t$$

Figure 2.5.: AAF extracted from texts.

A set of arguments S is said to attack an argument a iff $\exists \mathsf{b} \in S$ attacks a. A

set of arguments S_1 attacks a set of arguments S_2 iff S_1 attacks some $\mathsf{a} \in S_2$. A set of arguments S is said to defend an argument a iff S attacks all arguments attacking a. A set of arguments S_1 defends a set of arguments S_2 iff S_1 defends every argument in S_2.

Several semantics have been defined for AAFs, all selecting sets of arguments as "accepted". The following semantics are relevant for this thesis:

- A set of arguments S is *conflict-free* iff S does not attack itself.
- A set of arguments S is an *admissible extension* iff S is conflict-free and defends itself.
- A set of arguments S is a *complete extension* iff S is an admissible extension and contains all arguments it defends.
- A set of arguments S is a *grounded extension* iff S is a $\subseteq$-minimal complete extension. The grounded extension $\mathbb{G}$ always exists and is unique. If there are unattacked arguments in the AAF, then $\mathbb{G}$ is non-empty.

Using the AAF from Figure 2.5, $\{t\}$, $\{t'\}$, and $\{t, t''\}$ are conflict-free, $\{t, t''\}$ is an admissible extension, and $\{t, t''\}$ is a complete extension as well as the grounded extension. Given that t'' is unattacked, $\mathbb{G}$ is non-empty.

2.3.1.1. AA-CBR

AA-CBR [48, 49] is an Abstract Argumentation (AA)-driven Case-Based Reasoning (CBR)-inspired model for reasoning with (past) cases (where a case is represented as a set of features together with an outcome) to predict the outcomes of new cases. AA-CBR resolves conflicts amongst most similar past cases with conflicting outcomes and justifies outcomes.

Consider a fixed set $\mathbb{F}$ of *features*, and a set $\mathbb{O} = \{\delta, \overline{\delta}\}$ of two (distinct) *outcomes*, with δ called the *default outcome* and $\overline{\delta}$ called the *non-default outcome*. Then:

- a *case* is a pair (X, o) with $X \subseteq \mathbb{F}$ and $o \in \mathbb{O}$;
- a *case base* is a finite set $CB \subseteq \wp(\mathbb{F}) \times \mathbb{O}$ that is *coherent*, i.e. for $(X, o_X), (Y, o_Y) \in CB$, if $X = Y$ then $o_X = o_Y$;
- a *new case* is a pair $(N, ?)$ with $N \subseteq \mathbb{F}$ and ? indicating that the outcome is yet unknown.

To illustrate, in this section we use the following example.

Example 2.3.1. Let $\mathbb{F} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$, $CB = \{(\{\mathsf{a}\}, \overline{\delta}), (\{\mathsf{b}\}, \overline{\delta}), (\{\mathsf{a}, \mathsf{c}\}, \delta), (\{\mathsf{b}, \mathsf{d}\}, \overline{\delta})\}$ and $(N, ?) = (\{\mathsf{a}, \mathsf{d}\}, ?)$. Then CB is coherent.

AA-CBR maps the problem of determining the outcome for a new case into a membership problem within the grounded extension of an AAF obtained from the case base CB, the new case $(N, ?)$ and the default outcome δ.

AA-CBR encompasses the following steps:

1. extracting an AAF from a coherent case base and a default outcome,
2. determining the outcome of a new case from the grounded extension of the AAF augmented to take into account the new case,
3. providing explanations for the outcome of a new case by means of dispute trees.

Step 1 gives $(Args, \mathcal{L}^-)$ with:

- $Args = CB \cup \{(\{\}, \delta)\}$, where the *default argument* $(\{\}, \delta)$ represents the outcome obtained in the absence of any information (i.e. features);
- for $(X, o_X), (Y, o_Y) \in CB \cup \{(\{\}, \delta)\}$, $((X, o_X), (Y, o_Y)) \in \mathcal{L}^-$ iff
  1. $o_X \neq o_Y$, and (different outcomes)
  2. $Y \subsetneq X$, and (specificity)
  3. $\nexists (Z, o_X) \in CB$ with $Y \subsetneq Z \subsetneq X$. (concision)

The construction of $(Args, \mathcal{L}^-)$ singles out conflicts between cases. In particular, a conflict arises if factors of one case form a subset of the factors of another case (specificity) and the two cases have different outcomes. AA-CBR models this with an attack from the latter to the former, provided that the latter is the most concise such case. Note that $(Args, \mathcal{L}^-)$ is finite, as CB is.

Step 2 gives the *AA-CBR outcome* of $(N, ?)$ defined as follows, for $(Args, \mathcal{L}^-)$ obtained from step 1. In the remainder we refer to the AA-CBR outcome of $(N, ?)$ given $(Args, \mathcal{L}^-)$ as the outcome of $((Args, \mathcal{L}^-), N)$.

i. let $(Args_N, \mathcal{L}^-_N)$ be the AAF with
   - $Args_N = Args \cup \{(N, ?)\}$;
   - $\mathcal{L}^-_N = \mathcal{L}^- \cup \{((N, ?), (Y, o_Y)) \,|\, (Y, o_Y) \in Args, Y \not\subseteq N\}$,
     i.e. $(Args_N, \mathcal{L}^-_N)$ extends $(Args, \mathcal{L}^-)$ with the new case argument $(N, ?)$ that attacks all arguments with 'irrelevant' features, i.e. features not in N;

ii. let $\mathbb{G}$ be the grounded extension of $(Args_N, \mathcal{L}^-_N)$; $\mathbb{G}$ is non-empty, as $(N, ?)$ is unattacked;

iii. then the *AA-CBR outcome* of $(N, ?)$ is
   - δ, if $(\{\}, \delta) \in \mathbb{G}$;
   - $\overline{\delta}$, otherwise, if $(\{\}, \delta) \notin \mathbb{G}$.

Example 2.3.2. Given CB, δ and $(N, ?)$ from Example 2.3.1, then step 2 gives $\overline{\delta}$. Indeed $(Args, \mathcal{L}^-)$ after step 1 is the AAF depicted in Figure 2.6 when the polygon node and all arrows from it are ignored, and $(Args_N, \mathcal{L}^-_N)$ from (i) in step 2 is the full AAF in the figure[1].

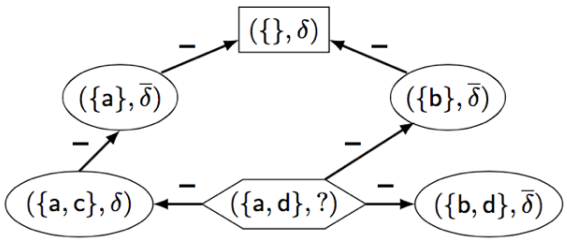


Figure 2.6.: $(Args_N, \mathcal{L}^-_N)$ for Example 2.3.2.

The grounded extension of $(Args_N, \mathcal{L}^-_N)$ is $\mathbb{G} = \{(\{\mathsf{a}, \mathsf{d}\}, ?), (\{\mathsf{a}\}, \overline{\delta})\}$, and $(\{\}, \delta) \notin \mathbb{G}$. Note that, in $(Args_N, \mathcal{L}^-_N)$, $((\{\mathsf{b}, \mathsf{d}\}, \overline{\delta}), (\{\}, \delta)) \notin \mathcal{L}^-$ because of the concision requirement: $(\{\mathsf{b}\}, \overline{\delta})$ is a more concise attacker of $(\{\}, \delta)$. Note also that $(N, ?)$ attacks all cases that have features not present in N.

Step 3 gives explanations for why an outcome for a new case is (not) computed by means of dialogical exchanges between a *proponent* P, in favour of the *default outcome*, and an *opponent* O, against the *default outcome*. The explanations are defined in terms of *dispute trees* [54, 56] where a *dispute tree* for $\mathsf{a} \in Args$ (in $(Args, \mathcal{L}^-)$) is a tree T such that:

1. every node of T is of the form $[L : \mathsf{x}]$, with $L \in \{P, O\}$ and $\mathsf{x} \in Args$ (i.e. the node is labelled by argument x and assigned the status of either P or O);
2. the root of T is a P node labelled by a;
3. for every P node n labelled by some $\mathsf{b} \in Args$, and for every $\mathsf{c} \in Args$ such that $(\mathsf{c}, \mathsf{b}) \in \mathcal{L}^-$, there exists a child of n, which is an O node labelled by c;
4. for every O node n, labelled by some $\mathsf{b} \in Args$, there exists at most one child of n which is a P node labelled by some $\mathsf{c} \in Args$ such that $(\mathsf{c}, \mathsf{b}) \in \mathcal{L}^-$;
5. there are no other nodes in T except those given by 1-4.

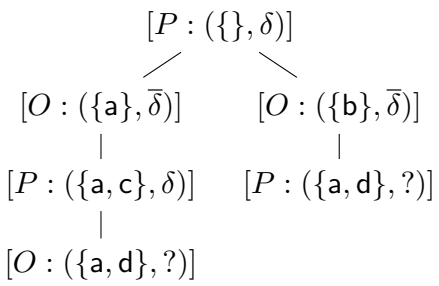
[1] We use different shapes for nodes for readability: the default argument $(\{\}, \delta)$ and the new case $(\{\mathsf{a}, \mathsf{d}\}, ?)$ argument are enclosed in rectangle and polygon respectively, while other arguments are enclosed in ellipses.

A dispute tree T for some $\mathsf{a} \in \mathit{Args}$ is a *maximal dispute tree* iff for all opponent nodes $[O : \mathsf{x}]$ which are leaves in T there is no $\mathsf{y} \in \mathit{Args}$ such that $(\mathsf{y}, \mathsf{x}) \in \mathcal{L}^-$. Thus, in a maximal dispute tree, no opponent leaf node is "attackable". An explanation for why the outcome for N is $\overline{\delta}$ is any *maximal dispute tree* for $(\{\}, \delta)$.

A dispute tree T is an *admissible dispute* tree iff (i) every O node in T has a child, and (ii) no argument in T labels both P an O nodes. If the outcome of the new case N is the default outcome δ, then there is a T which acts as an explanation for why the outcome of N is δ (i.e. T is an *admissible dispute tree* for $(\{\}, \delta)$).

The explanation from the tree T can be interpreted as a dialectical exchange between P and O where P seeks to establish the default outcome and O seeks to establish the non-default. O argues for the case in question by putting forward arguments with the non-default outcome $\overline{\delta}$ and P tries to defend against those. In the end, if P cannot argue against an argument put forward by O, the outcome of the case is deemed to be the non-default. Otherwise, if O has no other arguments to put forward and all of O's arguments are attacked by P's arguments, then P 'wins' and the outcome of the case is deemed to be the default.

The $(\mathit{Args}_N, \mathcal{L}_N^-)$ represented in Figure 2.6 yields outcome $\overline{\delta}$. We obtain a *maximal dispute tree* T as follows:



Note that this tree is not admissible as argument $(\{\mathsf{a}, \mathsf{d}\}, ?)$ labels both an opponent and a proponent node and P cannot argue against O. Thus, the outcome is the non-default $\overline{\delta}$.

Consider Example 2.3.1 but with $(N, ?) = (\{\mathsf{a}, \mathsf{c}\}, ?)$. The resulting $(\mathit{Args}_N, \mathcal{L}_N^-)$ is as in Figure 2.7. As the grounded extension of $(\mathit{Args}_N, \mathcal{L}_N^-)$ is $\mathbb{G} = \{(\{\}, \delta), (\{\mathsf{a}, \mathsf{c}\}, \delta), (\{\mathsf{a}, \mathsf{c}\}, ?)\}$, and $(\{\}, \delta) \in \mathbb{G}$, the outcome is δ.

We obtain an *admissible dispute tree* T as follows:

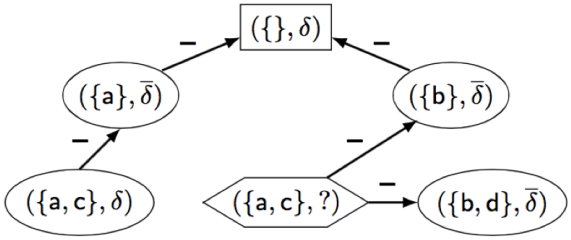
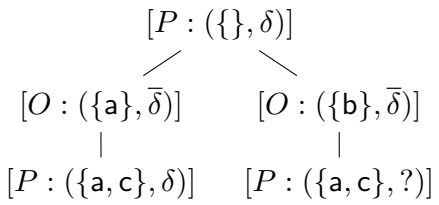


Figure 2.7.: $(Args_N, \mathcal{L}_N^-)$ for Example 2.3.2 but with $(N, ?) = (\{\mathsf{a}, \mathsf{c}\}, ?)$.



Thus, as O has no other arguments to put forward and all its arguments are attacked by P, the outcome is the default δ. Indeed, the new case is matching exactly an existing case whose outcome is δ.

In Chapter 6, we will use AA-CBR to explain predictions for binary classifications of textual data. We will show the dialectical exchange between P and O that argue for each of the two possible outcomes in the binary classification task.

2.3.2. (Quantitative) Bipolar Argumentation Frameworks

Bipolar Argumentation Frameworks (BAFs) extend AAFs by considering two independent binary relations between arguments: attack and support [40]. Formally, a BAF is any $\langle Args, \mathcal{L}^-, \mathcal{L}^+ \rangle$ consisting of a set $Args$ of arguments, a binary (*attack*) relation $\mathcal{L}^-$ on $Args$ and a binary (*support*) relation $\mathcal{L}^+$ on $Args$. For $\alpha, \beta \in Args$, if $(\alpha, \beta) \in \mathcal{L}^-$, then we say α attacks β, if $(\alpha, \beta) \in \mathcal{L}^+$, then we say α supports β. For $\alpha \in Args$: $\mathcal{L}^-(\alpha) = \{\beta \in Args | (\beta, \alpha) \in \mathcal{L}^-\}$ is the set of *attackers* of α and $\mathcal{L}^+(\alpha) = \{\beta \in Args | (\beta, \alpha) \in \mathcal{L}^+\}$ is the set of *supporters* of α.

For example, consider the following three texts taken from the dataset introduced in [37]:

t: *"We should grant politicians immunity from prosecution"*
t': *"The ability to prosecute politicians is the ultimate protection against the abuse of power"*
t'': *"Giving politicians immunity from prosecution allows them to focus on performing their duties"*

Here t' *attacks* t, t'' *supports* t, and t'' and t' *neither attack nor support each other*. The BAF is given in Figure 2.8 as a graph with two types of edges: *attack* represented as $-$ and *support* represented as $+$.

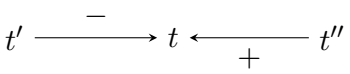


Figure 2.8.: BAF extracted from texts.

Semantics of BAFs have been defined in terms of extensions (similarly to AAFs) [40]. In this thesis, we use *gradual semantics* instead, defined for Quantitative Bipolar Argumentation Frameworks (QBAFs).

As in [11], a QBAF is a quadruple $\langle Args, \mathcal{L}^-, \mathcal{L}^+, \tau \rangle$ where $\langle Args, \mathcal{L}^-, \mathcal{L}^+ \rangle$ is a BAF and $\tau : Args \to \mathbb{I}$, a total function where $\mathbb{I} = [0, 1]$. For any $\alpha \in Args$, $\tau(\alpha)$ is the *base score* of α and the *strength* of α is given by $\sigma(\alpha)$, where $\sigma : Args \to \mathbb{I}$ is a total function (also referred to as a *gradual semantics*). For any set of arguments $S \subseteq Args$ we denote $\sigma(S)$ the set of all elements of the set $\{\sigma(\alpha) | \alpha \in S\}$ (thus $\sigma(S) \in \mathbb{I}^* \triangleq \cup_{i \geq 0} \mathbb{I}^i$).

We consider three gradual semantics for a QBAF: Restricted Euler-based (REB) [2], QuAD semantics [12], and DF-QuAD semantics [162] which we review below and illustrate on the QBAF with the BAF in Figure 2.9 and $\tau(\alpha) = 0.5$ for each argument α.

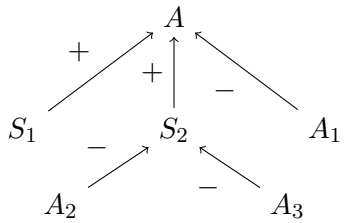


Figure 2.9.: Example BAF.

The **REB** semantics is such that for $\alpha \in Args$, $\sigma(\alpha) = 1 - \frac{1-\tau(\alpha)^2}{1+\tau(\alpha)\cdot e^{E_\alpha}}$ where $E_\alpha = \sum_{\beta \in \mathcal{L}^+(\alpha)} \sigma(\beta) - \sum_{\gamma \in \mathcal{L}^-(\alpha)} \sigma(\gamma)$.

Considering the BAF from Figure 2.9, we have:

$$\sigma(A_2) = 1 - \frac{1-\tau(A_2)^2}{1+\tau(A_2)\cdot e^{E_{A_2}}} = 1 - \frac{1-0.5^2}{1+0.5\cdot e^0} = 0.5$$

as $E_{A_2} = 0$ since A_2 has no attackers nor supporters. Thus, all the other arguments α that have no attackers and supporters will have $\sigma(\alpha) = 0.5$ (i.e. A_2, A_3, S_1, A_1). To compute the strength of S_2 we have:

$$E_{S_2} = -(\sigma(A_2) + \sigma(A_3)) = -(0.5 + 0.5) = -1 \Rightarrow$$
$$\sigma(S_2) = 1 - \frac{1-\tau(S_2)^2}{1+\tau(S_2)\cdot e^{E_{S_2}}} = 1 - \frac{1-0.5^2}{1+0.5\cdot e^{-1}} = 0.366$$

Having calculated the strength of S_2 (A's only argument that is in turn attacked) we can now calculate the strength of A.

$$E_A = \sigma(S_1) + \sigma(S_2) - \sigma(A_1) = 0.5 + 0.366 - 0.5 = 0.366 \Rightarrow$$
$$\sigma(A) = 1 - \frac{1-\tau(A)^2}{1+\tau(A)\cdot e^{E_A}} = 1 - \frac{1-0.5^2}{1+0.5\cdot e^{0.366}} = 0.564$$

The **QuAD** semantics is such that, for $\alpha \in Args$ with $\tau(\alpha) = v_0$, $\sigma(\alpha) = g(v_0, \mathcal{F}_a(v_0, \sigma(\mathcal{L}^-(\alpha))), \mathcal{F}_s(v_0, \sigma(\mathcal{L}^+(\alpha))))$ where if $(\alpha_1, \ldots, \alpha_n)$ is an arbitrary permutation of the $(n \geq 0)$ attackers in $\mathcal{L}^-(\alpha)$, $\sigma(\mathcal{L}^-(\alpha)) = (\sigma(\alpha_1), \ldots, \sigma(\alpha_n))$ (similarly for supporters), and $g, \mathcal{F}_a, \mathcal{F}_s$ are defined as follows. The operator $g : \mathbb{I} \times \mathbb{I} \cup \{nil\} \times \mathbb{I} \cup \{nil\} \to \mathbb{I}$ is defined, for $v_0, v_a, v_s \in \mathbb{I}$, as:
$g(v_0, v_a, v_s) = v_a$ if $v_s = nil$ and $v_a \neq nil$;
$g(v_0, v_a, v_s) = v_s$ if $v_a = nil$ and $v_s \neq nil$;
$g(v_0, v_a, v_s) = v_0$ if $v_a = v_s = nil$;
$g(v_0, v_a, v_s) = \frac{v_a + v_s}{2}$ otherwise.

Letting $\times$ stand for either a or s, the operator $\mathcal{F}_\times$ is defined as $\mathcal{F}_\times : \mathbb{I}^* \to \mathbb{I}$, where for $S = (v_1, \ldots, v_m) \in \mathbb{I}^*$, $(w_1, \ldots, w_n)$ is an arbitrary permutation of the

[2]This formulation is a modification of the original formulation of $\mathcal{F}_\times$, in which $(w_1, \ldots, w_n)$ was not used.

non-zero elements in S^2:
if $n = 0$: $\mathcal{F}_\times(v_0, S) = nil$;
if $n = 1$: $\mathcal{F}_\times(v_0, S) = f_\times(v_0, w)$;
if $n > 1$: $\mathcal{F}_\times(v_0, (w_1, \ldots, w_n)) = f_\times(\mathcal{F}_\times(v_0, (w_1, \ldots, w_{n-1})), w_n)$;
with the *base expressions* $f_\times : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ defined, for $v_0, v \in \mathbb{I}$, as:
$f_a(v_0, v) = v_0 - v_0 \cdot v = v_0 \cdot (1 - v)$ and
$f_s(v_0, v) = v_0 + (1 - v_0) \cdot v = v_0 + v - v_0 \cdot v$.

Considering the BAF from Figure 2.9, we have:

$$\sigma(S_2) = g(\tau(S_2), \mathcal{F}_a(\tau(S_2), \sigma(\mathcal{L}^-(S_2))), \mathcal{F}_s(\tau(S_2), \sigma(\mathcal{L}^+(S_2))))$$

where S_2 is attacked by A_2 and A_3, and has no supporters. Then

$$\sigma(S_2) = g(0.5, \mathcal{F}_a(0.5, \sigma(A_2), \sigma(A_3)), \mathcal{F}_s(0.5, 0))$$

For an argument α with no attackers and no supporters $\sigma(\alpha) = \tau(\alpha)$ (i.e. A_2, A_3, S_1, A_1). To compute the strength of S_2 we have:

$$\begin{aligned}\sigma(S_2) &= g(0.5, \mathcal{F}_a(0.5, \tau(A_2), \tau(A_3)), nil) = \mathcal{F}_a(0.5, 0.5, 0.5) \\ &= f_a(\mathcal{F}_a(0.5, 0.5), 0.5) = f_a(f_a(0.5, 0.5), 0.5) = f_a(0.25, 0.5) = 0.125\end{aligned}$$

We can now calculate the strength of A, as we have the strengths of all its attackers and supporters.

$$\begin{aligned}\sigma(A) &= g(\tau(A), \mathcal{F}_a(\tau(A), \sigma(\mathcal{L}^-(A))), \mathcal{F}_s(\tau(A), \sigma(\mathcal{L}^+(A)))) \\ &= g(0.5, \mathcal{F}_a(0.5, \sigma(A_1)), \mathcal{F}_s(0.5, \sigma(S_1), \sigma(S_2))) \\ &= \frac{1}{2} \cdot (\mathcal{F}_a(0.5, 0.5) + \mathcal{F}_s(0.5, 0.5, 0.125)) \\ &= \frac{1}{2} \cdot (0.25 + f_s(\mathcal{F}_s(0.5, 0.5), 0.125)) \\ &= \frac{1}{2} \cdot (0.25 + f_s(f_s(0.5, 0.5), 0.125)) \\ &= \frac{1}{2} \cdot (0.25 + f_s(0.5 + 0.5 - 0.5 \cdot 0.5, 0.125)) \\ &= \frac{1}{2} \cdot (0.25 + f_s(0.75, 0.125)) = 0.515\end{aligned}$$

The **DF-QuAD** semantics is such that, for any $\alpha \in Args$ with $\tau(\alpha) = v_0$ and n attackers with strengths $v_1, \ldots, v_n$ and m supporters with strengths $v'_1, \ldots, v'_m$, $\sigma(\alpha) = \mathcal{C}(v_0, \mathcal{F}(v_1, ..., v_n), \mathcal{F}(v'_1, ..., v'_m))$. The combination function $\mathcal{C}$, for an argument with base score v_0, attackers with strengths $v_1, ..., v_n$ (for $n \geq 0$, $n = 0$ amounts to the argument having no attackers and supporters with strengths $v'_1, ..., v'_m$ (for $m \geq 0$, $m = 0$ amounts to the argument having no supporters) is defined as follows, for $v_a = \mathcal{F}(v_1, ..., v_n)$ and $v_s = \mathcal{F}(v'_1, ..., v'_m)$:
if $v_a = v_s$: $\mathcal{C}(v_0, v_a, v_s) = v_0$;
if $v_a > v_s$: $\mathcal{C}(v_0, v_a, v_s) = v_0 - (v_0 \cdot |v_s - v_a|)$;
if $v_a < v_s$: $\mathcal{C}(v_0, v_a, v_s) = v_0 + ((1 - v_0) \cdot |v_s - v_a|)$.
Given n arguments with strengths $v_1, ..., v_n$, if $n = 0$ then $\mathcal{F}(v_1, ..., v_n) = 0$, otherwise $\mathcal{F}(v_1, ..., v_n) = 1 - \prod_{i=1}^{n}(|1 - v_i|)$.

Considering the BAF from Figure 2.9, we have:

$$\sigma(S_2) = \mathcal{C}(\tau(S_2), \mathcal{F}(\tau(A_2), \tau(A_3)), 0)$$

As the strength of an argument with no attackers and no supporters is given by its base score, we have:

$$\begin{aligned}
\sigma(S_2) &= \mathcal{C}(0.5, \mathcal{F}(0.5, 0.5), 0) \\
&= \mathcal{C}(0.5, 1 - |1 - 0.5| \cdot |1 - 0.5|, 0) \\
&= \mathcal{C}(0.5, 0.75, 0) \\
&= 0.5 - (0.5 \cdot |0 - 0.75|) = 0.125
\end{aligned}$$

We can now calculate the strength of A.

$$\begin{aligned}
\sigma(A) &= \mathcal{C}(\tau(A), \mathcal{F}(\tau(A_1)), \mathcal{F}(\tau(S_1), \tau(S_2))) \\
&= \mathcal{C}(0.5, \mathcal{F}(0.5), \mathcal{F}(0.5, 0.125)) \\
&= \mathcal{C}(0.5, 0.5, 0.5625) \\
&= 0.5 + ((1 - 0.5) \cdot |0.5625 - 0.5|) = 0.531
\end{aligned}$$

In this thesis, we will use quantitative gradual semantics to derive dialectical strength of arguments in BAFs mined from texts.

2.3.3. Other approaches

There are other approaches in Argumentation that do not play a role in this thesis but that we mention in this section to highlight some key differences.

Walton et al. [194, 195] proposed argumentation schemes which represent structures or templates for forms of arguments. Argumentation schemes enable one to identify and evaluate common types of argumentation in everyday discourse. The schemes can be analysed into the roles that propositions play in each scheme and the structure of the associated propositions. The schemes can serve as templates for analysing and creating arguments and can be seen as building blocks for constructing an argumentation graph. Knowing a text falls into a specific scheme, we could employ AM techniques to fill in the graph [65, 107]. However, in this thesis, we are interested in developing models that can be applied on various natural language texts where a certain structure is not known upfront.

The frameworks we have presented in Section 2.3 are abstract. In Abstract and (Quantitative) Bipolar Argumentation, the internal structure of an argument has no relevance to the task and thus is not imposed. In contrast, other types of Argumentation models have been proposed that take structure into consideration. Structured Argumentation [15] assumes some sort of structure imposed upon arguments. In Assumption-Based Argumentation (ABA) [57, 186] arguments are identified as deductions, supported by assumptions and rules, and the attacks are identified in terms of contraries of assumptions, where assumptions are premises that form the support in arguments for conclusions. In this thesis, we focus on Abstract and (Quantitative) Bipolar Argumentation as a means to evaluate the previously mined arguments from text. Hence, the notion of structure in our case is implicitly considered during the mining phase.

2.4. Explainable AI

In this thesis, we are are interested in how Argumentation can be used to provide explanations. Below, we provide a brief overview of approaches in Explainable AI.

Explaining ML algorithms has attracted a lot of interest recently (see [82] for a survey of black box models that have been addressed). In this thesis, 'explainability' and 'interpretability' are used interchangeably. Interpretability comes in two forms: global (when the entire model is interpretable as well as the reasoning

behind each outcome) and local (when only the reasons behind outcomes can be understood - i.e. the single prediction can be explained). Some works focus on explaining models [166, 167] whereas others focus on explaining the decision of ML/DL algorithms for forms of data common in human daily life such as images and texts [6, 106, 112]. Some works have employed Argumentation for explanations [4, 74, 160]. Graphs similar to BAFs have also been extracted from texts [80, 191] that are able to provide insights into debates or opinions by providing a (visual) summary which can be used to provide explanations. We will give more details about these works in the following.

In terms of annotated datasets, Camburu et al. [35] extended the SNLI corpus [26] with explanations of the entailment relations that can be used to learn justifications of a model's decisions. ANNs have been targeted as models that are needed to become more interpretable. In [8, 101, 208] Decision Trees and ANNs are combined to give Neural Decision Trees. Such approaches can potentially be interpretable as trees are easier to analyse than ANNs.

In terms of explainability of models, Ribeiro et al, [166] proposed a model-agnostic system that explains ML models with rules called anchors. These have been used in various tasks: from sentiment prediction (an anchor explanation being that the words 'not' and 'bad' occurring together give a positive sentiment), to POS tagging (an anchor explanation being that the word 'play' is a verb if it is preceded by a pronoun, and is a noun if it is preceded by a determiner), and images (by highlighting the areas in the image associated to the object predicted). In the same spirit, Ribeiro et al. [167] proposed Local Interpretable Model-agnostic Explanations (LIME).

Attention networks have also been successful at providing explanations by highlighting, in text or images, the most important aspects for classification. Hudson and Manning [93] proposed a recurrent Memory, Attention, and Composition network for visual question answering. In text, identifying words that are most important in classifying new instances can be seen as explanations. Lai and Tan [106] provided users the words that SVMs use to determine whether a review is true or deceptive to test human accuracy in this task with varying levels of machine assistance (i.e. heatmap of words, classification with accuracy, classification without accuracy). Highlighting rationales from texts was done in [112] that extracted words from the input text that suffice for predicting the sentiment. Arras et al. [6] aimed at explaining RNNs predictions for sentiment using Layer-wise Relevance Propagation (LRP) thus showing the words that contribute to classify new

instances. Pörner et al. [154] conducted a comprehensive evaluation of explanation methods for NLP and in particular testing RNNs and Convolutional Neural Networks where explanations are based on LRP by computing the contributions of neurons in topic classification and morphological prediction (number: singular or plural). However, these approaches require large datasets and are bespoke. Additionally, it has been argued that attention models, while improving performance, do not in fact provide meaningful explanations for classification tasks as the learnt attention weights are uncorrelated with feature importance and that different attention distributions give similar predictions [96].

Argumentation has also been used to generate explanations in AI. For example, Argumentation has been used for explaining decisions [3], classifications [4], recommendations [28, 160], as well as (non-)acceptance of arguments [63, 64, 74]. In [4], argumentation and concept learning are used to reason about classifications. Arguments are examples and hypotheses. Preferences over arguments are employed to resolve conflicts from an inconsistent training set, using the grounded extension to capture the version space model, while also supporting generation of argumentative explanations for classifications. Dispute trees have been used in explanations [74, 206]. García et al. [74] formalised dialectical explanations for argument-based reasoning. Within the legal setting, argumentation has been used to explain multi-criteria decision making obtained from dispute trees [206]. Briguez et al. [28] relied on a set of predefined postulates describing the conditions in which a movie should be recommended to a given user and which can be translated into DeLP [42, 73, 169] rules. Explanations are extracted from the dialectical tree supporting a recommendation. In [160], an argumentation-driven recommender system was defined to provide explanations for recommendations extracted from AFs.

Grosse et al. [80] proposed a framework to represent Twitter comments as "opinion trees" with conflicting elements identified being similar to dialectical trees. Wachsmuth et al. [191] proposed an argumentation model for classification and "sentiment flow patterns" that generalise well. The "opinion trees" and the "sentiment flow patterns" can be considered as being similar to BAFs that we extract from text in this thesis and that we then use to provide explanations.

2.5. Summary and Discussion

In this chapter we provided the background required for NLP and DL classifiers used throughout this thesis. We gave an overview of AFs that we employed in this thesis. We presented related work for AM and RbAM in particular, and gave a brief overview of existing approaches for Explainable AI.

The NLP and DL classifiers background provides the basis for understanding the DL architectures developed in Chapters 3 and 4 for 2 class and 3 class RbAM, respectively. We use the 3 class RbAM model to mine BAFs that benefit applications and show how explanations can be derived from the mined BAFs in Chapter 5. This literature provides the necessary background for understanding AAFs. In Chapter 6, we use AA-CBR to mine AAFs from (textual) data related to binary classification and provide explanations for classifications.

3. Deep Learning for 2 class Relation-based Argument Mining

In this chapter we propose different neural architectures to address the 2 class RbAM problem (i.e. distinguishing between attack and support relations). We give the motivation for this work in Section 3.1. We describe and give examples from the datasets concerned with 2 class RbAM in Section 3.2. We present four types of architectures that address this problem in Section 3.3 and, in Section 3.4, conduct extensive experimentation to test their portability across the 2 class RbAM datasets. We summarise the research in this chapter in Section 3.5.

3.1. Motivation

The majority of the datasets that focus on argumentative relations contain only two types of relations: attack and support (see [31] for a survey of datasets and the domains of their applicability). Despite the numerous approaches proposed to tackle the 2 class RbAM problem, the models are hardly portable from one domain to another due to features used.

Most results are reported using cross-validation within a single dataset, thus the testing and training datasets are similar in stucture and wording. Few attempts have been made at testing on different datasets from the ones used for training. Some works have tested their proposed models on different topics selected from the dataset available [9, 33, 180] or on datasets similar to the ones models have been trained on [178]. Some studies that attempt testing on a different dataset from the training one do not target only the relations but also other forms of AM such as determining the argument components [138].

We define neural models to address the two class problem, analysing all (to the best of our knowledge) available relational datasets for this task, and show that our models obtain homogeneous results on all the datasets proposed in the literature for RbAM. This way, the (Rb)AM community can use these baselines to test whether

models are portable across domains/datasets and that the models do not perform well only on the datasets they had been trained on. Note that these baselines are not to be considered state-of-the-art and often obtaining better results on the respective dataset can be achieved by adding features related to the dataset tested. Our focus here however is on portability.

All our neural models are instances of the general DL architecture in Figure 3.1. There are two input texts and we are interested in determining the argumentative relation between the first, called *child*, and the second, called *parent*, by means of a neural network. Each of the two texts is represented using embeddings as well as other (standard) features. We propose four neural models for the 2 class RbAM task, two concerned with the way child and parent are passed through the network (*concatenation* and *mix* models), an autoencoder model, and an attention-based model.

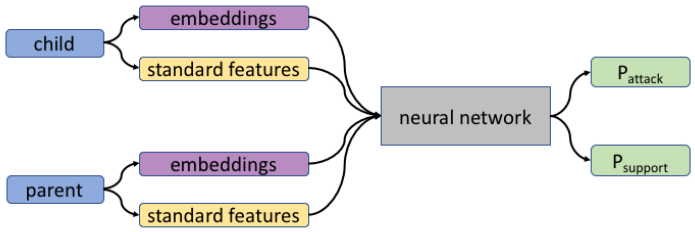


Figure 3.1.: General Deep Learning (DL) architecture. P_{attack} stands for the probability of *attack* class and $P_{support}$ stands for the probability of *support* class. The output is given by the class with the highest probability.

For our baseline models for 2 class RbAM we use GRUs [44] as the underlying neural network. We could have alternatively used LSTMs. Various works have compared LSTMs and GRUs, e.g. [46, 99], but they did not obtain conclusive results as to which architecture is better, suggesting that the design choice is dependant on the dataset and task. We focus on GRUs as we run several (10 x 8 x 4) experiments and GRUs take less time to train and are more efficient since LSTMs have more parameters (see Section 2.1.3 for descriptions of LSTM and GRU models). The advantage with respect to time of GRUs make GRUs suitable for this task.

3.2. Datasets for 2 class Relation-based Argument Mining

We describe the datasets comprising argumentative relations of support and attack that we used in our baseline models for RbAM. We assign identifiers (ids) to the dataset names for readability. We do not consider the two legal datasets built for relation prediction by Palau and Moens [141] and Teruel et al. [184] because the former is not available and the latter has a low inter-annotator agreement.

Persuasive essays (essay)

Stab and Gurevych [178] introduced a corpus of 402 persuasive essays annotated with discourse-level argumentation structures. The argumentation structure is that of a tree structure with the major claim being the root of the tree that represents the author's standpoint on the topic, which is supported or attacked by claims which in turn can be supported or attacked by premises. An example of (a part of) an essay is below:

Ever since researchers at the Roslin Institute in Edinburgh cloned an adult sheep, there has been an ongoing debate about whether cloning technology is morally and ethically right or not. Some people argue for and others against and there is still no agreement whether cloning technology should be permitted. However, as far as I'm concerned, $[$*cloning is an important technology for humankind*$]_{MajorClaim1}$ *since* $[$*it would be very useful for developing novel cures*$]_{Claim1}$.

First, $[$*cloning will be beneficial for many people who are in need of organ transplants*$]_{Claim2}$. $[$*Cloned organs will match perfectly to the blood group and tissue of patients*$]_{Premise1}$ *since* $[$*they can be raised from cloned stem cells of the patient*$]_{Premise2}$.

In this example, Claim1 and Claim2 each support the Major Claim, Premise1 supports Claim2 and Premise2 supports Premise1.

The Fleiss' κ [67] for support relations is 0.708 and 0.737 for attack relations which allows tentative conclusions [105]. Combining the argumentative relations between claims and major claims with the ones between premises and claims, we obtain 497 attack relations and 4841 support relations.

Microtexts (micro)

Peldszus and Stede [149] introduced a corpus of 112 microtexts covering con-

troversial issues where the length of each text is about 5 argumentative discourse units (ADUs). An example of a microtext can be seen in Figure 3.2. In this example, the second segment rebuts the first segment and the third segment undercuts the link between the second segment and the first segment. Segments four and five jointly support the main claim.

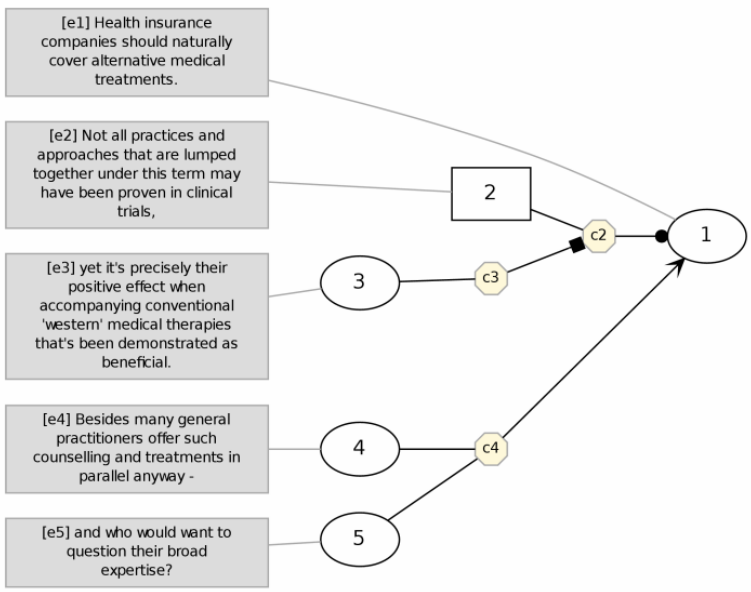


Figure 3.2.: An example of microtext and the associated argumentation graph.

Each argumentative relation can be of two types: support (normal and example), and attack (rebut which attacks the conclusion of an inference and undercut which attacks the connection between the premises and conclusion without attacking the conclusion directly). We focus only on normal supports (as examples are only rarely used) and rebut attacks (as we are not interested in attacks that target an edge thus challenging the acceptability of the inference). The dataset contains 108 attack relations and 263 support relations.

Nixon-Kennedy debate (nk)

Menini et al. [128] introduced a corpus from the Nixon - Kennedy presidential campaign in 1960 covering five topics: Cuba, disarmament, healthcare, minimum wage and unemployment. Below are two examples from the dataset:

- *We could have tried to inject ourselves into the Congo without honoring our*

commitments to the United Nations charter , just as Khrushchev seems to be trying to do . We could have turned Cuba into a second Hungary . But we can be eternally grateful that we have a man in the White House who did none of these things. **supports** *I don't take the views that the only alternative to a dictator is a Communist dictator. If the United States had just had its influence , and at that time the United States was extremely powerful in Cuba, it seems to me we could have persuaded Mr. Batista to hold free elections at the time he was permitted to go and permit the Cuban people to make their choice instead of letting Castro seize power through revolution. I think we are going to have a good deal of trouble in the future with Castro through all of Latin America.*

- *They are afraid of diplomatic policies that teeter on the brink of war. They are dismayed that our negotiators have no solid plans for disarmament. And they are discouraged by a philosophy that puts its faith in swapping threats and insults with the Russians.* **attacks** *It's a typically specious and frivolous maneuver. We have made a good-faith effort to advance the - advance toward disarmament - and make some progress by having a meeting of the Disarmament Commission. Now, when they make a proposal like this, it's a cynical attempt to prevent progress, that's what it is it shows that they don't really want disarmament.*

The dataset contains 378 attack relations and 353 support relations. The Fleiss' κ on 100 pairs annotated by 3 annotators is 0.63.

Debatepedia (db)

Cabrio and Villata [33] introduced a corpus from two platforms, Debatepedia[1] and ProCon[2], with arguments from the following topics: violent games/aggressiveness, China one-child policy, consider coca as a narcotic, child beauty contests, Arming Libyan rebels, random alcohol breath tests, Osama death photo, privatizing social security, Internet access as a right, ground zero mosque, mandatory military service, no fly zone over Libya, airport security profiling, solar energy, natural gas vehicles, use of cell phones/driving, marijuana legalization, gay marriage as a right, vegetarianism. Below are two examples from the dataset:

- *Research studies have yielded the conclusion that the effect of violent media*

[1] idebate.org/debatabase
[2] www.procon.org/

consumption on aggressive behavior is in the same ballpark statistically as the effect of smoking on lung cancer, the effect of lead exposure on children's intellectual development and the effect of asbestos on laryngeal cancer. **supports** *Violent video games are real danger to young minds.*

- *People know video game violence is fake.* **attacks** *Youth playing violent games exhibit more aggression.*

The dataset contains 141 attack relations and 179 support relations. The Cohen's κ on 100 pairs annotated by 2 annotators is 0.74.

IBM (ibm)

Bar-Haim et al. [9] introduced a dataset with topics randomly selected from the debate motions database at the International Debate Education Association (IDEA)[3]. Below are two examples from the dataset:

- *Children with many siblings receive fewer resources.* **supports** *This house supports the one-child policy of the republic of China.*
- *Virtually all developed countries today successfully promoted their national industries through protectionism.* **attacks** *This house would unleash the free market.*

The IBM Debater Claim Stance Dataset contains 1069 attack relations and 1325 support relations.

ComArg (com)

Boltužić and Šnajder [21] introduced ComArg, a corpus of online user comments related to two topics, gay marriages and under God in pledge collected from ProCon and IDEA. There are five types of relations: explicitly attacking the argument (A), vaguely/implicitly attacking the argument (a), no use of the argument (N), vaguely/implicitly supporting the argument (s), explicitly supporting the argument (S). For our experiments, we combine the two types of attacks (A and a) and the two types of supports (S and s).

Below are two examples from the dataset:

- *Religion should stay out of the public square, except when people exercise their right to the freedom of speech an expression. Having Under God in the pledge forces all people to pledge allegiance to a higher power they may*

[3] idebate.org/

not believe in. The separation of Church and State should disallow such favoritism. Can anyone fathom the reaction of believers if it said: One Nation, created by a big bang and inhabited by evolved creatures.... ? **supports** *Removing under god would promote religious tolerance*

- *Atheism doesn't mean the absence of religion - it means the absence of a god in one's belief system. Certain forms of Buddhism, for example are atheistic. Therefore, requiring a statement of belief in a god is unconstitutionally preferring a majority religious belief over a minority one. The point of the Pledge is to state allegiance to the flag and country. If one believes in a god, there are many, many other forums in which to express that belief without imposing it on others.* **attacks** *America is based on democracy and the pledge should reflect the belief of the American majority.*

ComArg contains 296 attack relations and 462 support relations. The Fleiss' κ is 0.49, Cohen's κ is 0.49, and Pearson's r is 0.71 on the dataset annotated by 3 annotators.

Web-content dataset (web)

Carstens and Toni [37] introduced a dataset consisting of arguments adapted from the Argument Corpus [193], in addition to arguments from news articles, movies, ethics and politics. Below are two examples from the dataset:

- *i agree did not like this either in fact i stopped watching once waltz was killed because i just didnt care anymore* **supports** *after all the attention and awards etc and an imdb rating of i was so shocked to finally see this film and have it be so bad*
- *samsung note has a bigger screen and a somewhat faster processor* **attacks** *htc one it is currently the best one in the market good quality superb specs*

The dataset contains 1301 attack relations and 1329 support relations. The Fleiss' κ on 200 instances annotated by 3 annotators is 0.4287.

CDCP (cdcp)

Park and Cardie [145] introduced a dataset consisting of support arguments only from user comments about rule proposals regarding Consumer Debt Collection Practices (CDCP) by the Consumer Financial Protection Bureau crawled from an

[4] regulationroom.org

eRulemaking website[4]. The support relations are of two types: reasons and evidence. Below are two examples from the dataset:

- *sundays really are when most people are spending whatever little time they have left before the workweek with friends and family* **supports** *i do not conduct business on sundays*
- *that is, a robo-call that tells you that you have a message or an account update, and the only way to get it is to call a special number with an extension, but when you call, it is just the same message asking where your payment is, is a waste of the consumer's time and the consumer's cellular resources (two phone calls, one received, one sent* **supports** *i support these restrictions on robo-calling and any calls during the work hours*

The dataset contains 1220 support relations. The Krippendorf's α [105] on the dataset annotated by 2 annotators is 44.1.

UKP Sentential (ukp)

Stab et al. [180] introduced a dataset of arguments extracted from online comments related to 8 controversial issues: abortion, cloning, death penalty, gun control, marijuana legalizat, minimum wage, nuclear energy, school uniforms. However, this dataset is aimed at argument search by means of identifying topic-relevant arguments where topics are controversial issues for which there exist an obvious polarity, either for or against. Below are two examples from the dataset:

- *Dr. Strouse has seen both the benefits and risks of cannabis use and is well-versed in the emerging scientific evidence regarding the effectiveness of cannabinoids in a variety of medical conditions and pain states, as well as epidemiologic evidence of legalized marijuana' s connection to a reduction in prescription drug use and opioid-related deaths* **supports** *marijuana legalization*
- *Would you want to live in a neighborhood filled with people who regularly smoke marijuana* **attacks** *marijuana legalization*

For our experiments, we modify the parent text from *topic* to a default seen as the natural language template *topic is good*. Hence from the previous example, we would have an argument for and an argument against "marijuana legalization is good".

The UKP Sentential dataset contains 5935 attack relations and 4759 support relations. The Cohen's κ on 200 pairs annotated by 2 annotators is 0.72.

AIFdb (aif)

AIFdb is a corpus of argument maps which follows the structure defined by AIF [109] (see Section 2.2.2.1 for more details). We select the following datasets from AIFdb and keep the English texts only:

- AraucariaDB;
- DbyD Argument Study (a selective study of argument in the philosophy of animal psychology from the Hathi Trust collection);
- Expert Opinion and Positive Consequences;
- Internet Argument Corpus (IAC - a corpus for research in political debate on internet forums);
- Mediation (here we compiled the following datasets: Dispute mediation, Dispute mediation: excerpts taken from publications, Mock mediation, Therapeutic, Bargaining, Meta-talk in mediation);
- Opposition (here we compiled the following datasets: Language Of Opposition Corpus 1, Android corpus, Ban corpus, Ipad corpus, Layoffs corpus, Twitter corpus).

Given that AIFdb has six types of relations, in order to use a 2 class RbAM model on AIFdb, we map the original set of relations to 2 class RbAM as follows: CA-nodes are mapped to attack and RA- and TA-nodes are mapped to support. The mapping results in the following number of relations: AraucariaDB: 31 attack relations and 2965 support relations; DbyD Argument Study: 245 attack relations and 894 support relations; Expert Opinion and Positive Consequences: 7 attack relations and 181 support relations; IAC: 7311 attack relations and 2165 support relations; Mediation: 251 attack relations and 902 support relations; Opposition: 1443 attack relations and no support relations.

Dataset summary

A summary of the datasets statistics can be found in Table 3.1.

Dataset	ID	Number attacks	Number supports
Essays	essay	497	4841
Microtexts	micro	108	263
Nixon-Kennedy	nk	378	353
Debatepedia	db	141	179
IBM	ibm	1069	1325
ComArg	com	296	462
Web-content	web	1301	1329
CDCP	cdcp	0	1220
UKP	ukp	5935	4759
AIFdb	aif	9854	7543

Table 3.1.: Summary of datasets.

3.3. Neural models

In this section we describe the features that we use for the neural models obtained as instances of the architecture in Figure 3.1. The choice of the four models is driven by the fact that we wanted to experiment which is the best way to feed the two texts into the network, i.e. whether processing them separately with a GRU is better than feeding both texts split by a separator token to a single GRU (*concatenation* vs *mix*), as well as testing whether an *attention* mechanism does improve results also in the RbAM task, and whether using the most salient features from an *autoencoder* would be better than using embeddings.

3.3.1. Features

We use four types of features: word embeddings, features related to sentiment, entailment, and syntactic features. The following applies to each text from the pair analysed, *child* and *parent* respectively. We refer to the last three types of features as *standard* features, and refer to all four types of features as *all* features.

Word embeddings are distributed representations of texts in an n-dimensional space. In all our models, we use pre-trained word representations. We also add a feature of entailment which we compute using AllenNLP[5], a textual entailment model that is based on a decomposable attention model [144]. The textual entailment module determines the probability between the two texts of entailment,

[5] allennlp.org/models

contradiction, and neutral, and we add as a feature the class with the highest probability.

The features related to sentiment are based on manipulation of senti wordnet [62] and sentiment of the entire text analysed. Every wordnet synset [133] is associated to three scores describing how objective, positive, and negative it is. For every word in the phrase, we select its first synset and compute its positive score and its negative score. The features related to sentiment for a text *t* that consists of *i* words are the following:

- sentiment score where $\sum_{w_i} pos_score(w_i) - neg_score(w_i)$
- number of positive words in *t* where a word w_i is considered positive if $pos_score(w_i) > 0 \wedge neg_score(w_i) > 0 \wedge pos_score(w_i) > neg_score(w_i)$
- number negative words where a word is considered negative if $pos_score(w_i) > 0 \wedge neg_score(w_i) > 0 \wedge neg_score(w_i) > pos_score(w_i)$
- number of neutral words where a word is considered neutral if not $(pos_score(w_i) > 0 \wedge neg_score(w_i) > 0)$
- sentiment class of *t* computed using VADER sentiment analyser [94]
- sentiment score of *t* computed using VADER

The last set of features are syntactic representing text statistics as well as word statistics with respect to the POS tag. We add the following 10 features extracted from text *t*: number of words, number of nouns, number of verbs, number of first person singular, number of second person singular and plural, number of third person singular and plural, number of first person plural, number of modals, number of modifiers (number of adverbs plus number of adjectives), and lexical diversity (calculated as number of unique words divided by the total number of words in *t*).

We now describe the four instances of the architecture in Figure 3.1 that we propose for determining the argumentative relation (attack or support) between two texts. The sizes of the hidden layers are the ones that performed the best. Details of the other hyperparameters and the number of hidden layers tried are given in Section 3.4.

3.3.2. Concatenation model (C)

In the concatenation model, each of the child and parent embeddings is passed through a GRU. We also concatenate the standard features that we obtain for the child and for the parent, respectively. The merged standard vector is then concatenated with the outputs of the GRUs. The resulting vector is passed through 2 dense

layers (of 256 neurons and 64 neurons, with sigmoid as activation function), to be then finally passed to softmax to determine whether the argumentative relation is one of attack or support. The concatenation model can be seen in Figure 3.3.

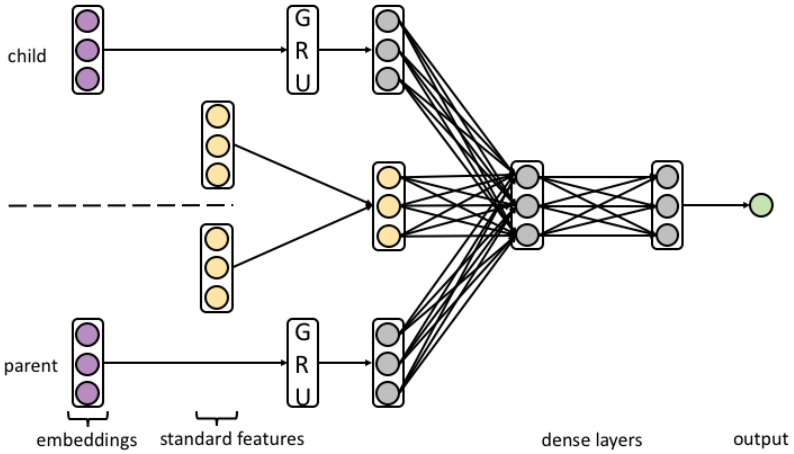


Figure 3.3.: Concatenation instance of DL architecture.

3.3.3. Mix model (M)

In the mix model, we first concatenate the child and parent embeddings and then pass them through a GRU, differently from the concatenation model where we pass each embedding vector through a GRU first. We concatenate the standard features that we obtain for the child and for the parent, respectively. The merged standard vector is then concatenated with the output of the GRU. From this stage, the network resembles the concatenation model: the resulting vector is passed through 2 dense layers (of 256 neurons and 64 neurons, with sigmoid as activation function), to be then finally passed to softmax to determine whether the argumentative relation is one of attack or support. The mix model can be seen in Figure 3.4.

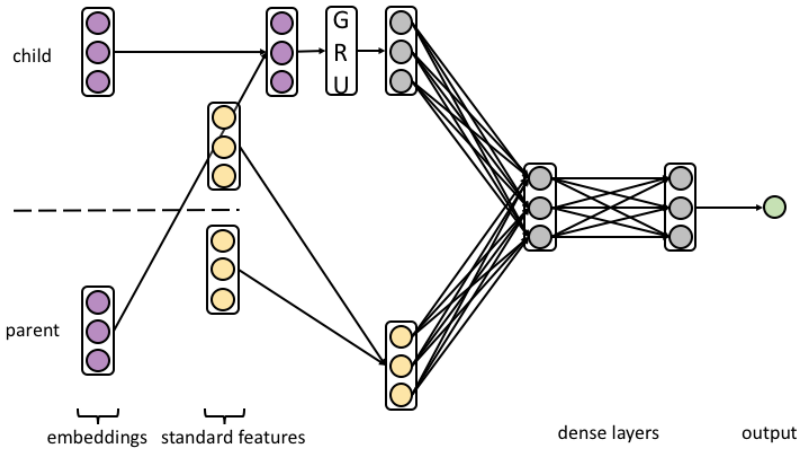


Figure 3.4.: Mix instance of DL architecture.

3.3.4. Autoencoder model (AE)

Autoencoders [61, 90] are unsupervised learning models based on Artificial Neural Networks (ANNs) which take a set of features as input and aim, through training, to reconstruct the inputs. Autoencoders can be used as feature selection methods to determine which features are redundant [88, 197].

In the autoencoder model, we first concatenate the child and parent tensors, to obtain a vector of size X. We use an autoencoder with one hidden layer defined as: (i) an encoder function $f(X) = \sigma(XW^{(1)})$, and (ii) a decoder function $\sigma(f(X)W^{(2)})$, where X is the input, $W^{(1)}, W^{(2)}$ are the weight parameters in the encoder and decoder, respectively. The size of the hidden layer is 128. We use sigmoid as activation function in the autoencoder and binary cross entropy as loss function. We concatenate the standard features that we obtain for the child and for the parent, respectively. The merged standard vector is then concatenated with the hidden layer in the autoencoder which represents the encoded dataset as dimensionally reduced features. The dimensionally reduced features are used as an alternative to word embeddings to test whether using the most salient features improves performance. The resulting vector is passed through a single dense layer (of 32 neurons, with sigmoid as activation function), to be then finally passed to softmax to determine whether the argumentative relation is one of attack or support. The autoencoder model can be seen in Figure 3.5.

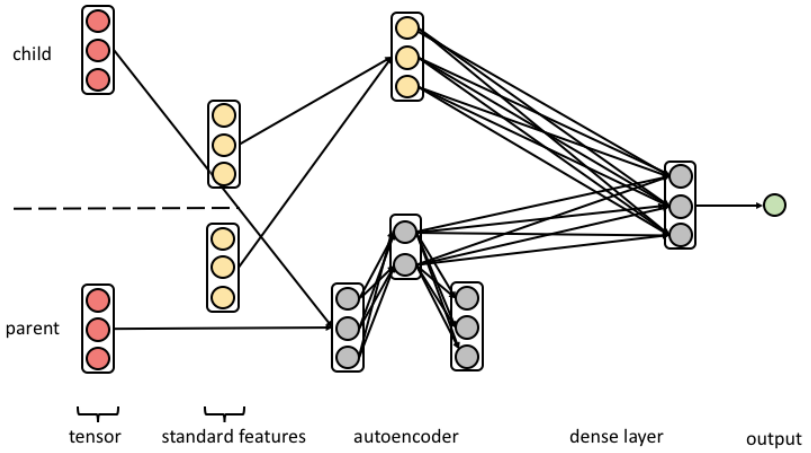


Figure 3.5.: Autoencoder instance of DL architecture.

3.3.5. Attention model (A)

Inspired by the demonstrated effectiveness of attention-based models [188, 200], we combine the GRU-based model with attention mechanisms.

In the attention model, each of the child and parent embeddings is passed through a GRU. Let $C \in \mathbb{R}^{L_c \times d}$ be the output the GRU produces when reading L_c words of Child C and $P \in \mathbb{R}^{L_p \times d}$ be the output the GRU produces when reading L_p words of Parent P. We compute attention in two directions: from Child C to Parent P and from Parent P to Child C. Here we illustrate the attention in one direction only. Let s_{ij} be the similarity matrix between the i-th child word and the j-th parent word, α the attention weight, c'_i the attended child vector, and c''_i the parent-aware representation of each child as follows:

$$
\begin{aligned}
s_{ij} = W^{1\times 2d}[c_{i:}; p_{j:}]^{2d\times 1} + b^{1\times 1} &\qquad s \in \mathbb{R}^{L_c \times L_p} \\
\alpha_{i:} = \text{softmax}(s_{i:}) &\qquad \alpha \in \mathbb{R}^{L_c \times L_p} \\
c'_{i:} = \alpha_{i:} p_{i:} &\qquad c' \in \mathbb{R}^{L_c \times d} \\
c''_{i:} = [c_{i:}; c'_{i:}] &\qquad c'' \in \mathbb{R}^{L_c \times 2d}
\end{aligned}
$$

where W is a trainable weight vector and $[;]$ is vector concatenation across row. The weights vectors W for the two directions are different. We also concatenate the standard features that we obtain for the child and for the parent, respectively. The merged standard vector is then concatenated with the outputs of the GRUs whose inputs are c'' and p''. The resulting vector is passed through a single

dense layer (of 128 neurons, with sigmoid as activation function), to be then finally passed to softmax to determine whether the argumentative relation is one of attack or support. The attention model can be seen in Figure 3.6.

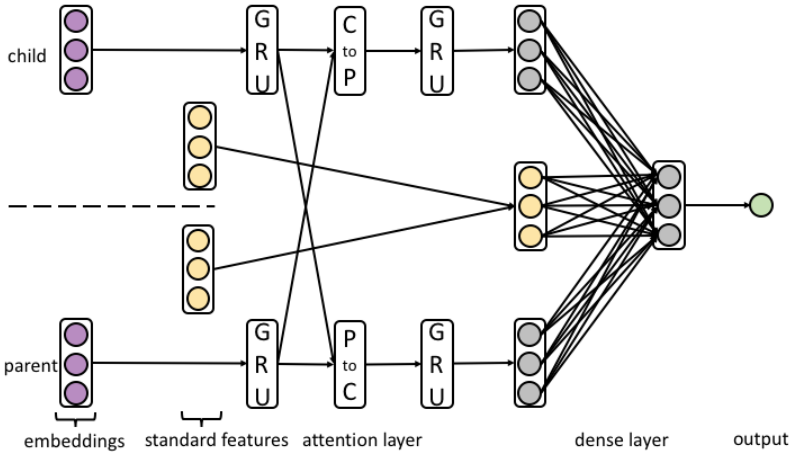


Figure 3.6.: Attention instance of DL architecture.

3.4. Experimental results

We have used for training the larger datasets, *aif*, *essay*, *ibm* and *web*. We resampled the minority class from the *essay* dataset and used our models on the oversampled dataset. The models are then tested on the remaining datasets with the average being computed on testing datasets.

We used Random Forests (RF) [27] as baseline with 15 trees in the forest and gini impurity criterion. On top of the *standard* features used for our neural models, we added the following features: TF-IDF, number of common nouns, verbs and adjectives between the two texts as in [128], a different sentiment score $\frac{\text{nr_pos}-\text{nr_neg}}{\text{nr_pos}+\text{nr_neg}+1}$ as in [9], all features being normalized. In addition, we added the Majority (Maj) class classifier as a baseline.

For our models, we experimented with two types of embeddings: GloVE [151] (300-dimensional) and FastText (FT) [98, 129] (300-dimensional). We used pretrained word representations in all our models. We used 100 as the sequence size as we noticed that there are few instances with more than 100 words. We used a batch size of 32 and trained for 10 epochs (as a higher number of epochs led to overfitting). For our models, we experimented with 1 and 2 hidden layers, and hidden layer sizes of 32, 64, 128, and 256, trying all possible combinations in order

to obtain the best configurations, and limiting to 2 hidden layers due to the small size of the data. We report only the best performing configurations in Section 3.3.

We conducted a feature ablation experiment (with embeddings being always used) and observed that syntactic features contribute the most to performance, with the other types of features bringing small improvements when used together only with embeddings. We report the results using embeddings and *syntactic* features and the results with *all* the features presented in Section 3.3.1. In addition, we have run experiments using two datasets for training: one of the large datasets (*aif*, *essay*, *ibm*, *web*) and one of the remaining datasets (represented as blanks in Tables[6]). The full set of experiments that predict for each class at least 0.2 F_1 can be found in the Appendix (Appendix A.1). Tables 3.2 and 3.3 show the (best) neural baselines for relation prediction in AM using *syntactic* and *all* features.

Amongst the proposed architectures, the attention and the concatenation models generally perform better. Using GloVe embeddings instead of FastText yields better results. The autoencoder does not give good results, which may be attributed to the fact that using the encoding of the most salient features is not enough in predicting the argumentative relation and that analysing the entire sequence is better.

Using only a single dataset for training, the models that perform the best are the *attention* model using *syntactic* features and the *mix* model using *all* features, both trained on the *essay* dataset. The best results are obtained when using another dataset along one of the larger datasets for training. This is because combining data from two domains we are able to learn better the types of argumentative relations. When using *syntactic* features, adding *micro*, *cdcp*, and *ukp* does not improve the results compared to using a single dataset for training. Indeed, *cdcp* has only one type of relation (i.e. support) resulting in an imbalanced dataset and in *ukp*, the parent argument is a topic, which does not improve the prediction task. When using *all* features, *nk* does not contribute to an increase in performance.

The best results are obtained using the attention mechanism with GloVE embeddings trained on the *web* and *essay* datasets using *syntactic* features (0.449 for F_1 A and 0.640 for F_1 S), and using the attention mechanism with GloVE embeddings using *all* features on the *ibm* and *essay* datasets (0.520 for F_1 A and 0.551 for F_1 S) and on the *web* and *essay* datasets (0.539 for F_1 A and 0.540 for F_1 S). Our baselines perform homogeneously over all existing datasets for relation prediction in AM while using generic features. Other works make use of features

[6]For readability, the columns in yellow represent the training datasets. F_1 A stands for the F_1 measure of the attack relation and F_1 S stands for the F_1 measure of the support (S) relation.

		essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
Maj	S	0.49	0.70	0.55	0.55	0.60	0.50	1.00	0.44	0.48	0.43	0.574
RF	F_1 A		0.24	0.22	0.25	0.03	0.27	-	0.22	0.43	0.31	0.496
	F_1 S		0.80	0.71	0.67	0.75	0.63	0.94	0.60	0.53	0.56	0.533
RF	F_1 A	0.60	0.40	0.43	0.56	0.38		-	0.52	0.45	0.54	0.485
	F_1 S	0.41	0.49	0.56	0.42	0.62		0.48	0.52	0.52	0.35	0.486
RF	F_1 A	0.67	0.45	0.63	0.62	0.56	0.66	-		0.68	0.71	0.623
	F_1 S	0.07	0.07	0.22	0.04	0.05	0.07	0.06		0.06	0.06	0.078
RF	F_1 A	0.67	0.45	0.61	0.62	0.56	0.61	-	0.71	0.68		0.614
	F_1 S	0.00	0.00	0.00	0.00	0.01	0.24	0.00	0.00	0.00		0.028
M G	F_1 A	0.49	0.40	0.50		0.50	0.53	-	0.57	0.48		0.496
	F_1 S	0.49	0.63	0.60		0.50	0.44	0.67	0.38	0.55		0.533
M FT	F_1 A	0.52	0.40	0.58	0.50	0.52		-	0.58	0.52		**0.517**
	F_1 S	0.47	0.50	0.61	0.54	0.54		0.60	0.42	0.53		**0.526**
M G	F_1 A	0.49	0.40	0.52	0.42	0.49		-	0.53	0.48		0.476
	F_1 S	0.51	0.64	0.61	0.60	0.52		0.69	0.43	0.56		0.570
M FT	F_1 A		0.36	0.42	0.45	0.46	0.47	-	0.71	0.50	0.46	0.479
	F_1 S		0.70	0.57	0.59	0.66	0.55	0.73	0.05	0.52	0.47	0.538
M G	F_1 A		0.33	0.36	0.35	0.36	0.38	-	0.67	0.43	0.40	0.410
	F_1 S		0.74	0.66	0.62	0.68	0.58	0.83	0.26	0.52	0.53	0.602
C G	F_1 A		0.37	0.42	0.51		0.50	-	0.60	0.45	0.48	0.476
	F_1 S		0.69	0.64	0.53		0.51	0.70	0.44	0.50	0.46	0.559
C FT	F_1 A		0.36		0.45	0.47	0.52	-	0.65	0.46	0.52	**0.490**
	F_1 S		0.64		0.55	0.67	0.53	0.72	0.34	0.48	0.47	**0.550**
C G	F_1 A		0.35	0.43	0.48	0.31	0.45	-	0.58		0.43	**0.433**
	F_1 S		0.71	0.68	0.58	0.70	0.54	0.77	0.47		0.50	**0.619**
C G	F_1 A	0.43	0.30	0.45		0.41		-	0.28	0.52	0.52	0.416
	F_1 S	0.60	0.69	0.65		0.54		0.72	0.59	0.46	0.55	0.600
A FT	F_1 A		0.37	0.50		0.49	0.48	-	0.63	0.46	0.49	0.489
	F_1 S		0.64	0.61		0.58	0.50	0.74	0.36	0.54	0.45	0.553
A G	F_1 A		0.37	0.58		0.53	0.53	-	0.61	0.59	0.55	**0.537**
	F_1 S		0.61	0.60		0.42	0.50	0.72	0.43	0.38	0.47	**0.516**
A G	F_1 A		0.36	0.48	0.43	0.39		-	0.52	0.45	0.51	**0.449**
	F_1 S		0.75	0.66	0.62	0.68		0.79	0.52	0.56	0.54	**0.640**
M FT	F_1 A		0.40	0.49	0.46	0.46		-	0.70	0.43	0.52	0.494
	F_1 S		0.64	0.59	0.55	0.72		0.70	0.13	0.48	0.46	0.534

Table 3.2.: Results using *syntactic* features on the datasets with attack (A) and support (S) relations. RF stands for Random Forests, A for autoencoder model, C for concatenation model, M for mix model, G for GloVE embeddings, and FT for FastText embeddings. The blanks represent the training datasets. The Average (Avg) does not include the results of the dataset(s) used for training.

		essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
M G	F1 0	0.49	0.41	0.51		0.50	0.52	-	0.57	0.49		**0.499**
	F1 1	0.49	0.63	0.60		0.52	0.45	0.69	0.37	0.54		**0.536**
C FT	F1 0	0.43	0.34	0.51	0.37	0.51		-	0.35	0.41		0.417
	F1 1	0.56	0.69	0.65	0.65	0.55		0.75	0.55	0.58		0.623
C G	F1 0	0.42	0.33	0.52	0.35	0.48		-	0.49	0.43		**0.431**
	F1 1	0.54	0.68	0.63	0.66	0.53		0.75	0.46	0.55		**0.600**
M G	F1 0	0.46	0.34	0.47	0.37	0.48		-	0.45	0.42		0.427
	F1 1	0.54	0.67	0.62	0.62	0.56		0.76	0.49	0.56		0.603
A FT	F1 0		0.37	0.39	0.39	0.41	0.44	-	0.60	0.37	0.45	0.428
	F1 1		0.72	0.60	0.59	0.67	0.56	0.82	0.41	0.57	0.50	0.604
A G	F1 0		0.30	0.42	0.40	0.38	0.43	-	0.53	0.35	0.48	**0.411**
	F1 1		0.73	0.61	0.60	0.68	0.55	0.82	0.49	0.56	0.53	**0.619**
M G	F1 0		0.37	0.43	0.43	0.40	0.46	-	0.71	-	0.46	**0.466**
	F1 1		0.71	0.64	0.61	0.70	0.55	0.78	0.11	0.78	0.51	**0.599**
C G	F1 0		0.36	0.44	0.52		0.50	-	0.58	0.46	0.48	0.477
	F1 1		0.65	0.64	0.52		0.50	0.70	0.47	0.51	0.46	0.556
C FT	F1 0		0.37		0.50	0.49	0.54	-	0.68	0.51	0.57	**0.523**
	F1 1		0.59		0.50	0.59	0.50	0.68	0.24	0.43	0.43	**0.495**
A G	F1 0		0.40	0.51	0.52	0.45	0.54		0.60	0.56	0.59	**0.521**
	F1 1		0.61	0.57	0.52	0.61	0.45		0.42	0.42	0.43	**0.504**
A FT	F1 0			0.38	0.35	0.41	0.41	-	0.51	0.35	0.42	0.404
	F1 1			0.64	0.60	0.67	0.57	0.83	0.50	0.55	0.52	0.610
A G	F1 0			0.44	0.40	0.41	0.39	-	0.43	0.41	0.43	0.416
	F1 1			0.64	0.62	0.67	0.56	0.83	0.54	0.55	0.53	0.618
M G	F1 0		0.42	0.48	0.50	0.36	0.58	-		0.46	0.57	0.481
	F1 1		0.60	0.62	0.57	0.69	0.44	0.59		0.46	0.43	0.550
A G	F1 0		0.36	0.54		0.50	0.51	-	0.59	0.59	0.55	**0.520**
	F1 1		0.67	0.63		0.49	0.51	0.74	0.47	0.41	0.49	**0.551**
M FT	F1 0		0.38	0.52		0.52	0.50	-	0.55	0.51	0.49	0.496
	F1 1		0.63	0.62		0.45	0.51	0.69	0.49	0.48	0.43	0.538
A FT	F1 0		0.40	0.43	0.43	0.37		-	0.55	0.26	0.50	**0.420**
	F1 1		0.72	0.64	0.58	0.72		0.77	0.47	0.59	0.51	**0.625**
A G	F1 0		0.43	0.54	0.49	0.46		-	0.59	0.63	0.63	**0.539**
	F1 1		0.68	0.55	0.57	0.56		0.65	0.46	0.38	0.47	**0.540**
C G	F1 0		0.44	0.47	0.54	0.43		-	0.61	0.48	0.59	0.509
	F1 1		0.60	0.59	0.44	0.63		0.48	0.45	0.46	0.37	0.503

Table 3.3.: Results using *all* features on the datasets with attack (A) and support (S) relations. RF stands for Random Forests, A for autoencoder model, C for concatenation model, M for mix model, G for GloVE embeddings, and FT for FastText embeddings. The blanks represent the training datasets. The Average (Avg) does not include the results of the dataset(s) used for training.

specific to the dataset, making it difficult to test those models on the other datasets. For instance, for the *essay* dataset, Stab and Gurevych [178] use structural features such as number of preceding and following tokens in the covering sentence, number of components in paragraph, number of preceding and following components in paragraph, relative position of the argument component in paragraph. For the other datasets, Stab et al. [180] use topic similarity features (as the *parent* argument is a topic), Menini et al. [128] use the position of the topic and similarity with other related/unrelated pair from the dataset, keyword embeddings of topics from the dataset.

In terms of results reported on the datasets we have conducted our experiments on, most works perform a cross-validation evaluation or, in the case of datasets consisting of several topics, the models proposed are trained on some of the topics and tested on the remaining topics. For the *essay* dataset, an Integer Linear Programming model was used to achieve 0.947 F_1 for the support class and 0.413 F_1 for the attack class on the testing dataset using cross-validation to select the model [178]. Using SVM, 0.946 F_1 for the support class and 0.456 F_1 for the attack class were obtained [178]. Using a modification of the Integer Linear Programming model to accommodate the lack of some features used for the *essay* dataset but not present in the *micro* dataset, 0.855 F_1 was obtained for the support class and 0.628 F_1 for the attack class. On the *micro* dataset, an evidence graph model was used to achieve 0.71 F_1 using cross-validation [149]. On the *nk* dataset, 0.77 F_1 for the attack class and 0.75 F_1 for the support class were obtained using SVM and cross-validation [128]. SVM accuracy results on the testing dataset using coverage (i.e. number of claims identified over the number of total claims) were reported in [9] as follows: 0.849 accuracy for 10% coverage, 0.740 accuracy for 60% coverage, 0.632 accuracy for 100% coverage. Random Forests were evaluated on the *web* and *aif* datasets using cross-validation, achieving 0.717 F_1 and 0.831 F_1, respectively [38]. Structured SVMs were evaluated in a cross-validation setting on the *cdcp* and *ukp* datasets using various types of factor graphs, full and strict [138]. On the *cdcp* dataset, F_1 was 0.493 on the full graph and 0.50 on the strict graph whereas on the *ukp* dataset, F_1 was 0.689 on the full graph and 0.671 on the strict graph. No results on the two-class datasets were reported for *db*, *com*, and *ukp* datasets. The results on *ukp* treat either supporting and attacking arguments as a single category or considering three types of relations: support, attack, neither. The latter type of reporting results on three classes is also given on the *com* dataset.

3.5. Summary and Discussion

In this chapter we have focused on the task of RbAM with 2 classes (*support* and *attack*). We addressed the issue of RbAM models that are hardly portable from one application domain to another due to the features used. We proposed a set of strong cross-dataset baselines based on four different instances of a neural architecture, and have shown that our models perform homogeneously over all existing datasets for relation prediction in AM consisting of *support* and *attack* relations.

While the model is better than average and provides a baseline for future research, further improvements can be made. One line of future work includes the addition of features related to the specific domain that was used during training to test whether these bring any improvements in performance (i.e. features related to political speeches in the case of nk or features related to structure of essays). Another direction for future work is related to transfer learning [142, 198]. We plan to experiment with transfer learning by using the model trained on one dataset as the starting point for models on the remaining datasets. In particular, the model will be trained twice. Firstly, we train the model on the dataset of choice, and secondly, we keep the trained model's weights and train it on the new dataset.

There are limitations to the 2 class RbAM task that makes it unsuitable to be applied "in the wild". This is due to the fact that in real world applications, text is unstructured, we do not have the clear 'child' and 'parent' arguments, but we need to first discard the pairs that are neither in a support nor in an attack relation. We will see how to overcome this problem in Chapter 4, where we focus on 3 class RbAM and its applications "in the wild".

4. Deep Learning for 3 class Relation-based Argument Mining

In this chapter we propose an LSTM-based model to address the 3 class RbAM problem (i.e. distinguishing between attack, support, and neither attack nor support relations) and then use it across two different applications: using 3 class RbAM to extract Bipolar Argumentation Frameworks (BAFs) from a factual text (Section 4.3.1) and from a dialogue (Section 4.3.2). We give the motivation for this work in Section 4.1. We describe our model in Section 4.2. We compare the mined BAFs from a factual text and a dialogue with the ones extracted by annotators in Section 4.3. We summarise the research in this chapter and propose directions for future work in Section 4.4.

4.1. Motivation

In Chapter 3 we focused on 2 class RbAM problem (i.e. distinguishing between attack and support relations). However, this approach is not sufficient to be deployed "in the wild" (i.e. to port the approach to applications rather than cross dataset experimentation). The main reason is that, in real world applications, we first need to discard the pairs that are neither in a support nor in an attack relation.

To address this problem, in this chapter we focus on 3 class RbAM (i.e. distinguishing between attack, support, and neither attack nor support relations). We propose an LSTM-based model for 3 class RbAM and show experimentally that using one dataset of 3 class RbAM performs better than using all available 3 class RbAM datasets.

Up to now, we have considered RbAM as a stand-alone task. We focus next on the deployment of 3 class RbAM in other types of applications, thus "in the wild". We focus on 3 class RbAM to determine argumentative relations as this represents a step towards extracting BAFs from text. We compare the BAFs extracted using the LSTM-based model with the ones extracted by annotators from two texts: a

factual text with reasons for and against renting a flat and an excerpt dialogue from the film *Twelve Angry Men*[1]. Having a model which is comparable to annotators can lead to the automatic creation of corpora, highly needed in experimentation.

4.2. Deep Learning for 3 class Relation-based Argument Mining

We define neural models to address the three class problem, analysing the available relational datasets for this task. The DL architecture can be seen in Figure 4.1. It aims at determining whether the *child* attacks, supports, or neither attacks nor supports the *parent*.

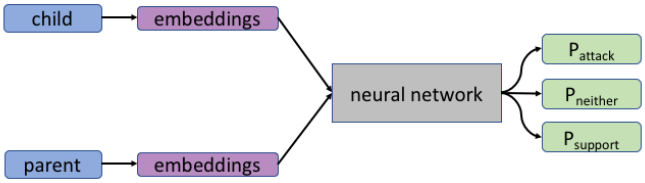


Figure 4.1.: DL architecture for 3 class RbAM. P_{attack} stands for the probability of *attack* class, $P_{neither}$ stands for the probability of *neither support nor attack* class, and $P_{support}$ stands for the probability of *support* class.

In particular, we use two parallel (attention-based) LSTMs to model the two texts, respectively. Similarly to the models described in Chapter 3, we opted for two neural architectures to model the two texts independently of one another, and then concatenate them. In contrast to the models presented in Chapter 3, here we focus on embeddings only and leave out standard features such as entailment and text statistics as the 3 class RbAM models are targeted at being deployed "in the wild", thus the texts will be of unknown length and will use an unknown style of language depending on application that the standard features will fail at capturing compared to embeddings.

[1] www-sop.inria.fr/NoDE/NoDE-xml.html#12AngryMen

We have also experimented with GRUs as an alternative to LSTMs as we have done in Chapter 3. However, the performance was lower. The results when using GRUs can be found in Appendix B.

4.2.1. Datasets for 3 class Relation-based Argument Mining

We describe the datasets comprising argumentative relations of support, attack, and neither support nor attack that we used in our 3 class RbAM models. We use the ids of the datasets as assigned in Section 3.2: *web* [37], *nk* [128], *com* [21], *ukp* [180]. We illustrate an example of each of the three classes, attack, support, neither attack nor support, from the selected datasets: web (Table 4.1), nk (Table 4.2), com (Table 4.3), and ukp (Table 4.4). Table 4.5 shows the number of each instances from each class on the selected datasets.

Relation	neither	support	attack
Child	the writer of this piece is working for a salary	have to agree brian theres a whole half hour of boring as he goes across the expanse of the desert that could be edited out	seriously when people don't like slow films there are always people that say you are incompetent and only like big explosions and big special effects this film tried too hard to be something it wasn't
Parent	you are ignoring centuries of good scholarship including that in the last century specifically engaging the arguments of historical criticism and socalled higher reasoning	i turned this movie off halfway through because i was bored out of my skull then i come here and its got a high rating granted its got a good storyline good acting nice cinematography but somehow still managed to bore me to tears	it's because you have a low attention span and can only handle films with hyperbolic acting action and constant fireworks explosions

Table 4.1.: Example of instances in the web dataset [37].

Relation	neither	support	attack
Child	Take my opponent, for example. He says on the Cuban situation that he did not intend to suggest that the American Government should intervene in the internal affairs of Cuba. And yet the words he used, the U.S. Government should support the anti-Castro forces in Cuba have been interpreted by his friends among the press, by all the people abroad, by all of the allies of the United States , as meaning exactly that.	They would have thought of the people and we would have had a bill that would have done something about it. I say to you today, and I pledge to you, that America, first, can not afford to allow any pockets of unemployment to exist. We can not afford not to use the tremendous human resources that are here in this area that are not being used.	I say in that connection that with all the criticism of President Eisenhower's administration, they can't take away the solid fact that under his leadership we got the United States out of one war; we've kept her out of other wars, and we do have peace without surrender today. Yes, there are problems, problems like the one in Cuba that my opponent is discussing tonight without offering any solution for, and, incidentally, may I say, looking at that problem, I will have more to say about that at a later time, and he had better look to the record of his administration before he starts criticizing ours in that field. He talks about dictators, and I can only say that the number of dictators we inherited from the Truman administration is far greater than the number we have today in Latin America, and that's something he must remember.
Parent	The Associated Press this month featured an excellent survey by Relman Morin entitled U.S. Prestige Is Damaged in Asia - citing our image in that continent as that of a well-meaning but confused and confusing giant. To the Asian, this report said, the administration's policy on Cuba was a clear case of American indecision, fumbling, and possible weakness. The events of last summer, the survey reported the U-2, summit, and Toyko chain of events appear to have shown that the United States was weak, indecisive, incapable of evaluating trends, anticipating developments and formulating action to meet them.	Throughout the entire Nation men and women almost 4 million of them are looking for jobs. Economists tell us that an unemployment rate of 6 percent is the danger signal. When a community passes that point it is officially regarded as an area of substantial labor surplus	Major policy on issues such as Cuban security is made at the highest levels - in the National Security Council and elsewhere and it is the party in power which must accept full responsibility for this disaster. The story of the transformation of Cuba from a friendly ally to a Communist base is in large measure the story of a government in Washington which lacked the imagination and compassion to understand the needs of the Cuban people which lacked the leadership and vigor to move forward to meet those needs and which lacked the foresight and vision to see the inevitable results of its own failures. And it is a tragic irony that even while these policies of failure here were being pursued our policymakers received repeated and urgent warnings that international communism was becoming a moving force behind Mr. Castro and the revolution that our interest and the interests of freedom were in danger that a new Soviet satellite was in the making.

Table 4.2.: Example of instances in the nk dataset [128].

Relation	neither	support	attack
Child	I've noticed some pro-people saying, We can't force our religion on others, but atheists can force theirs on us? Technically speaking, Atheism isn't a religion, just the lack of one. So really, we're not forcing any sort of religion on you, we just want to have yours removed from the Pledge.	church and state should be seperate. saying this why should under God be in the pledge? many americans that love their country are not christians so why should they have to say under God	America was most definitely NOT founded on religion, let alone monotheism. Every argument I have ever heard supporting religious reference in the Pledge has been, quite frankly, nonsensical. We're a secular nation, so we should have a secular Pledge
Parent	Likely to be seen as a state sanctioned condemnation of religion	Removing under god would promote religious tolerance	Under God is part of American tradition and history

Table 4.3.: Example of instances in the com dataset [21].

Relation	neither	support	attack
Child	Such a strategy does not simply attempt to bypass the 'macro' level of existing structures in the way that some focusses on alternatives do, such as promoting changes in lifestyles only at the level of the individual	In fact, to equal the electrical output of a single 3.000 megawatt-electric nuclear generating station, it would take roughly 2.000 wind turbines or 140 square kilometres of solar panels	Wind farms can be up and running in less than a year
Parent	nuclear energy	nuclear energy	nuclear energy

Table 4.4.: Example of instances in the ukp dataset [180].

Relation / Dataset	web	com	ukp	nk
neither	1973	1540	14353	731
support	1329	462	4759	353
attack	1301	296	5935	378

Table 4.5.: The split of the selected datasets into the three types of relations.

The *web* dataset is balanced and covers topics such as UKIP, opinions about movies, technology, and politics. Unlike the *web* dataset, the *ukp* dataset does not contain fully formed arguments but is aimed at argument search by means of identifying topic-relevant arguments where topics are controversial issues for which there exist an obvious polarity, either for or against. The *com* dataset contains two

topics only, "gay marriages" and "under God in pledge". These topics can be seen as being too narrow for ML algorithms trained on these datasets, hence they may not perform well when deployed in the wild. These datasets are also (highly) imbalanced and, in both cases, the relation is between an argument and a topic, hence not a fully constructed argument. The *nk* dataset consists of political speeches from 1960s. Here again, the restriction of topics in addition to stylistic changes in language since 1960 may pose a problem for ML algorithms trained on this dataset to be efficiently deployed "in the wild".

4.2.2. Neural models

We use LSTMs to encode each text separately and then concatenate the output vectors in order to classify the argumentative relation. Whilst we do not impose texts to be single sentences, we do however limit the input sequences to 50 words. We pad the inputs with size smaller than this threshold with zeros at the end to produce sequences of exactly 50 words. We initialise the word embeddings with the 100-dimensional GloVe vectors and the words that do not appear in the vectors are treated as unknown. Each LSTM model produces a vector representation of the text being analysed without any context from the other text. We concatenate the two vectors obtained from the LSTMs (by performing tensor concatenation) and feed the resulting vector to a softmax classifier to determine the label from $L = \{attack, support, neither\}$ for the relation between the two texts. Figure 4.2 describes the architecture that we use for determining the type of relation from L between texts.

We also add attention to our LSTM-based model. We compute attention in two directions: from Child *C* to Parent *P* and from Parent *P* to Child *C* as detailed in Section 3.3. Figure 4.3 describes the architecture with attention incorporated that we use for determining the type of relation from L between texts.

4.2.3. Experimental results

Due to the differences of the datasets described in Section 4.2.1, we run two types of experiments: one using instances from the *web* dataset only, and one using instances from all 3 class RbAM datasets: *web*, *com*, *ukp*, *nk*.

As a baseline, we have used the majority class algorithm (i.e. predicting *neither* class) and have considered several methods for RbAM explored in the literature, namely SVMs and Random Forests [9, 21, 128, 138]. For SVMs we use $C = 40$,

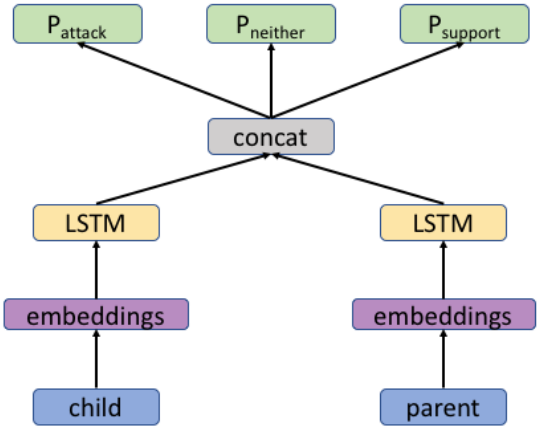


Figure 4.2.: The LSTM-based architecture.

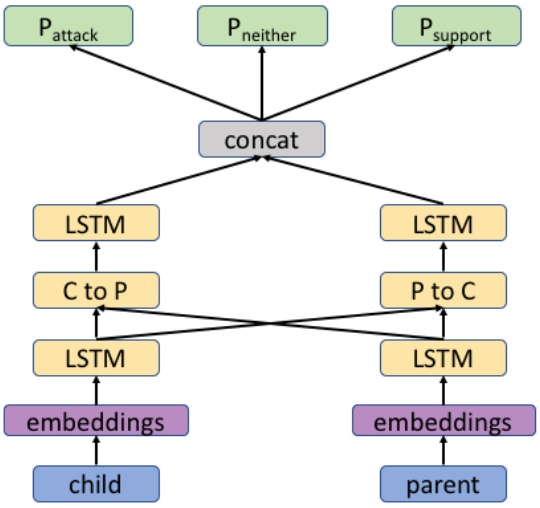


Figure 4.3.: The LSTM-based architecture with attention.

Feature	Detail
Number of words	For each sentence
Average word length	For each sentence
Sentiment polarity	For each sentence
Jaccard similarity	Size of the intersection of words in sentences compared to the size of union of words in sentences
Levenshtein distance	Count of replace and delete operations required to transform one sentence into the other
Word order	Normalized difference of word order between the sentences
Malik	Sum of maximum word similarity scores of words in same POS class normalized by sum of sentence's lengths (path and lch)
Combined semantic and syntactic	Linear combination of semantic vector similarity and word order similarity (path and lch)

Table 4.6.: Overview of features used in determining relations between pairs of sentences.

while for Random Forests we use 10 trees in the forest and gini impurity criterion. The classification models for the baselines use the features shown in Table 4.6. In particular, for the 'combined semantic and syntactic' feature, we use two similarity measures between words: *path* represents the shortest path that connects the senses in the is-a (hypernym/hypnoym) taxonomy and *lch* represents the Leacock-Chodorow similarity, namely the shortest path between the senses divided by double the maximum depth in the taxonomy in which the senses occur.

We run 5 stratified fold cross-validation (so that each fold is a good representative of the whole). We report the baseline results in terms of F_1 measure using 5 fold cross-validation (CV) in Table 4.7. The majority class gives better results using only the *web* dataset compared to using all datasets due to the class imbalance. The same pattern can be observed with respect to Random Forests and SVMs: majority class on the *web* dataset obtains the best results. On the remaining baselines, using all datasets for training gives better results than using solely the *web* dataset, with improvements of 4.6% when using Random Forests, and 10% when using SVMs. The best results, excluding the results obtained using the majority class algorithm, are obtained using SVMs trained on all datasets, resulting in a F_1 measure of 47.6%.

In the experiments using LSTMs, we set the dimension of the word embedding to be 100 and the LSTM dimension to be 32. The model parameters were optimised using the Adam method [100] with learning rate 0.001. We trained for 20

Algorithm	Dataset	Average F_1 CV
Majority class	*web*	60.0
Majority class	*web*, *com*, *ukp*, *nk*	20.0
Random Forests	*web*	40.8
Random Forests	*web*, *com*, *ukp*, *nk*	45.4
SVM	*web*	37.6
SVM	*web*, *com*, *ukp*, *nk*	47.6

Table 4.7.: Baseline results using 5 fold CV.

Algorithm	Dataset	Average F_1 CV	standard deviation
LSTM	*web*	51.9	0.3
LSTM	*web*, *com*, *ukp*, *nk*	50.6	0.6
Attention LSTM	*web*	42.5	0.4
Attention LSTM	*web*, *com*, *ukp*, *nk*	49.1	0.7

Table 4.8.: Results using 5x5 fold CV.

epochs or until the performance on the development set stopped improving using a mini-batch size of 32. We used sigmoid as activation function and cross-entropy loss. We perform 5 runs of 5 stratified fold CV to show our models are stable. The results using LSTM-based models can be seen in Table 4.8.

The standard deviation is less than 0.7 for 5 runs of CV for all models, showing that our models are stable. Using LSTMs with attention trained on all datasets results in an improvement in F_1 of 6.6% compared to using only the *web* dataset for training. However, the attention does not result in better performance compared to LSTM-based models without attention. This may be attributed to the fact that the datasets are different and having the *parent* argument as a topic in the training dataset does not help the attention-based mechanism to learn better. The best results are obtained using the LSTM-based model without attention trained on the *web* dataset with 51.9% F_1.

With respect to the baselines, the LSTM-based model trained on the *web* dataset performs better with improvements in F_1 of 11.1% compared to Random Forests (the best performing algorithm trained solely on the *web* dataset) and improvements in F_1 of 4.3% when compared to the best performing algorithm used as baseline, namely SVMs trained on all datasets.

We did not use the results reported in [38] as baselines as these results are obtained for datasets from various sources, including but not limited to the data we have used and the results on each dataset coming from a different source are not

reported. In terms of results reported on the datasets we have conducted our experiments on, SVMs were used on the *nk* dataset obtaining, using cross-validation, 71% F_1 on the *neither* class, 77% F_1 on the *attack* class and 75% F_1 on the *support* class. On the *com* dataset, SVMs with cross-validation resulted in a micro F_1 of 81.6%, whereas on the *web* dataset, SVMs with cross-validation yielded 62.1% F_1. On the *ukp* dataset, in a cross-topic experiment, 66.62% macro F_1 was obtained for the task of distinguishing between the *neither* class and *support* and *attack* relations treated as a single category, whereas on the 3 class problem, 42.8% F_1 was obtained using BiLSTM with topic information added as a term to the LSTM gates.

In the remainder of this chapter, we will use the (best performing amongst alternatives) LSTM-based model trained on the *web* dataset in various applications.

4.3. Comparing mined BAFs with annotators

We describe the use of trained classifiers "in the wild", analysing two entire, unseen texts: a factual text describing reasons for and against renting a flat and an excerpt dialogue taken from the movie *Twelve Angry Men*.

We asked human annotators to identify the argumentative relations from these texts. Given that all texts are argumentative, the result is a BAF. These annotators were masters Computing students who had been lectured on AM, RbAM, and BAFs but were otherwise non-expert annotators. We compared the argumentative relations extracted using trained classifiers with the ones extracted by the annotators. We experimented with various ML techniques to determine the type of argumentative relations between texts: SVMs, Random Forests (RFs), and the best performant LSTM-based model as reported in Section 4.2.3, all techniques trained on the *web* dataset.

For the first experiment, we used the BAFs generated by 19 annotators from a factual text. For the second experiment we used the BAFs generated by 21 annotators from a dialogue. We did not require the identification of the exact boundaries of arguments within the dialogue, instead we specified these boundaries (i.e. each paragraph contains an argument) and asked the annotators to focus on the identification of the relations between arguments. In both experiments, we also included in our analysis the BAF that we manually extracted from the texts. We present the first experiment of mining BAFs from a factual text in Section 4.3.1 and the second experiment of mining BAFs from a dialogue in Section 4.3.2.

4.3.1. Mining BAFs from a factual text

Our first experiment focused on testing whether ML classifiers perform similarly to humans when deployed on factual texts. We asked 19 annotators to manually extract a BAF from the following text which presents reasons for and against Mary renting a specific flat:

$\mathbf{B_0}$ *Mary should rent the flat.*
$\mathbf{B_1}$ *The flat is near Mary's work.*
$\mathbf{B_2}$ *The flat is located in a well illuminated and safe area.*
$\mathbf{B_3}$ *The flat is in a quiet area, because most of the neighbours are retired.*
$\mathbf{B_4}$ *The flat is tiny.*
$\mathbf{B_5}$ *Despite the size, spaces are well distributed within the flat.*
$\mathbf{B_6}$ *The flat shows signs of humidity problems.*
$\mathbf{B_7}$ *There are rumours that a nightclub will open in the area.*
$\mathbf{B_8}$ *Humidity problems take time to resolve.*
$\mathbf{B_9}$ *Local legislation forbids the opening of a nightclub in the area.*
$\mathbf{B_{10}}$ *The building's insurance covers fixing any problems promptly.*
$\mathbf{B_{11}}$ *A renowned architect designed the building to optimise space distribution.*

We also included in the annotations by the 19 users the BAF that we manually extracted from the text, given in Figure 4.4.

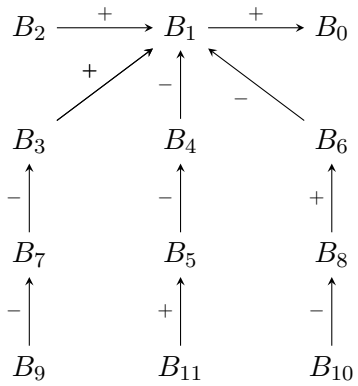


Figure 4.4.: Our manually extracted BAF from the flat example.

Without being given specific instructions about the boundaries of arguments, several annotators identified arguments at sentence level, whereas others have split the sentences into multiple arguments. We report results for the "merged" version

	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}
B_0												
B_1	**20s**											
B_2	**19s**	$1s$										
B_3	**19s**	$1s$										
B_4	**19a**	$1a$										
B_5					**19a** $1s$							
B_6	**19a**	$1a$										
B_7	**9a**			**8a** $1s$			$1s$					
B_8							$3a$ **14s**	$1a$				
B_9				$1s$			$1a$	**16a**	$1s$			
B_{10}	$2s$						**9a** $1s$		**8a**			
B_{11}	$1s$				$2a$	$1a$ **18s**						

Table 4.9.: The relations (*a* for *attack* and *s* for *support*) and the number of times they have been indicated by the annotators for the "merged" version.

of arguments in Section 4.3.1.1 and results for the "unmerged" version of arguments in Section 4.3.1.2.

4.3.1.1. Merged arguments

Table 4.9 shows the number of each type of relation identified by annotators (*support/attack*). From now on, we focus only on the relations on which at least 6 annotators have agreed. This is because fewer relations than 6 were deemed to be insignificant in evaluating trained classifiers and 6 represents more than a quarter of the number of annotators used here. When double annotations occurred, we used majority voting to select the relation type. The relations (and the number of times each relation has been identified) are shown in Figure 4.5.

We used Fleiss' kappa [67] and Krippendorff's alpha [105] to compute the inter-annotator agreement. Both methods gave an agreement of 0.655, showing substantial agreement. The number of relations identified by each annotator compared to the number of relations identified by at least 6 of the remaining annotators yields an average of 79.11% agreement.

We report the performance of our trained ML classifiers in determining the type of relations from the ones identified by annotators (in other words, we determine recall). The recall of our ML classifiers is shown in Figure 4.6. RFs perform

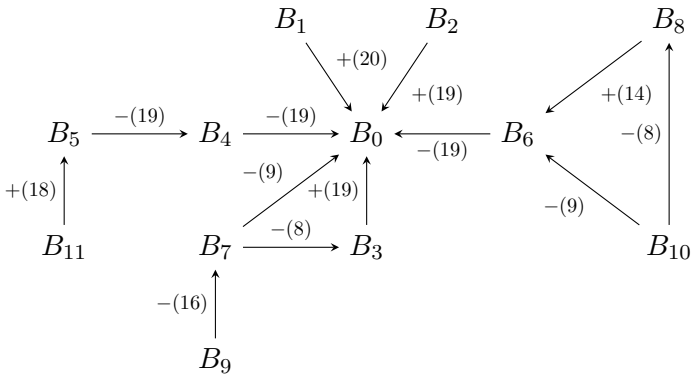


Figure 4.5.: Argumentative relations and the number of times these have been annotated for the "merged" version.

the best, identifying 77% of the relations labelled by at least 14 annotators, being able to identify both *support* and *attack* relations, unlike LSTM and SVM which classify all relations as *attack*. We believe this is because the length of arguments was shorter than those in our training dataset.

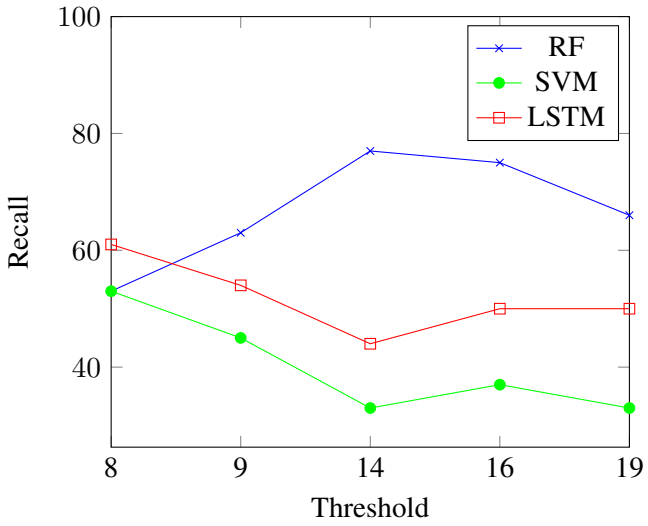


Figure 4.6.: Recall of ML classifiers we used to determine argumentative relations in the "merged" version of the factual text. Threshold t represents the number of times the relations have been identified by at least t annotators.

4.3.1.2. Unmerged arguments

Some annotators split the sentences in different arguments. This applies to $\mathbf{B_2}$ and $\mathbf{B_3}$ and they are illustrated in Table 4.10 as follows:

	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}
B_0																
B_1	**20s**															
B_2	**17s**	$1s$														
B_3	**7s**	$1s$														
B_4	**19a**	$1a$														
B_5					**19a** $1s$											
B_6	**19a**	$1a$														
B_7	**9a**			$3a$ $1s$			$1s$								$5a$	
B_8							$3a$ **14s**	$1a$								
B_9				$1s$			$1a$	**16a**	$1s$							
B_{10}	$2s$						**9a** $1s$		**8a**							
B_{11}	$1s$				$2a$	$1a$ **18s**										
B_{12}	$2s$															
B_{13}	$2s$															
B_{14}	**12s**															
B_{15}															**12s**	

Table 4.10.: The relations (*a* for *attack* and *s* for *support*) and the number of times they have been indicated by the annotators for the "unmerged" version.

$\mathbf{B_2}$ *The flat is located in a well illuminated and safe area.*
is identified in three forms:
$\mathbf{B_2}$ *The flat is located in a well illuminated and safe area.*
$\mathbf{B_{12}}$ *The flat is located in a well illuminated area.*
$\mathbf{B_{13}}$ *The flat is located in a safe area.*

whereas:

$\mathbf{B_3}$ *The flat is in a quiet area, because most of the neighbours are retired.*
is identified in three forms:
$\mathbf{B_3}$ *The flat is in a quiet area, because most of the neighbours are retired.*
$\mathbf{B_{14}}$ *The flat is in a quiet area.*
$\mathbf{B_{15}}$ *Most of the neighbours are retired.*

The relations (and the number of times each relation has been identified) are shown in Figure 4.7.

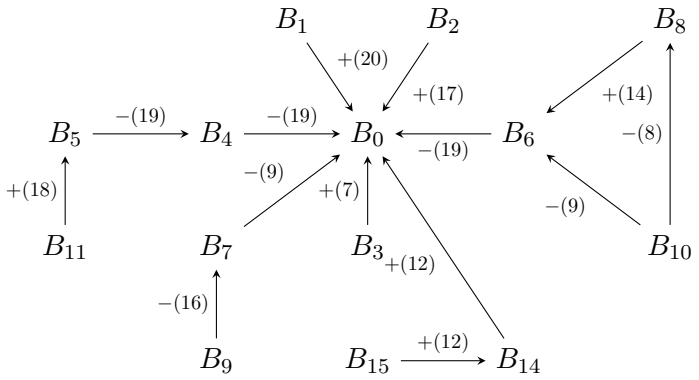


Figure 4.7.: Argumentative relations and the number of times these have been annotated for the "unmerged" version.

The inter-annotator agreement using Fleiss' kappa is 0.564 and Krippendorff's alpha is 0.565, showing moderate agreement. The number of relations identified by each annotator compared to the relations identified by at least 6 of the remaining annotators, yields an average of 74.26% agreement.

The recall of our ML classifiers is shown in Figure 4.8. Again, RFs perform the best, identifying 75% of the relations that have been identified by at least 14 annotators being able to identify both *support* and *attack* relations, unlike LSTM and SVM which classify all relations as *attack*. As the threshold increases, LSTM improves because it identifies all relations as attacks and the attack relations have been identified by at least 19 annotators. RFs identify the *support* relation between (B_{14}, B_0) and (B_{15}, B_{14}) as well as the relation between (B_3, B_0) related to argument B_3 which has been split into two arguments by the annotators. The relations from arguments B_{12} and B_{13} are not considered as these had been identified by only 2 annotators, this number being smaller than our threshold of at least 6 annotators.

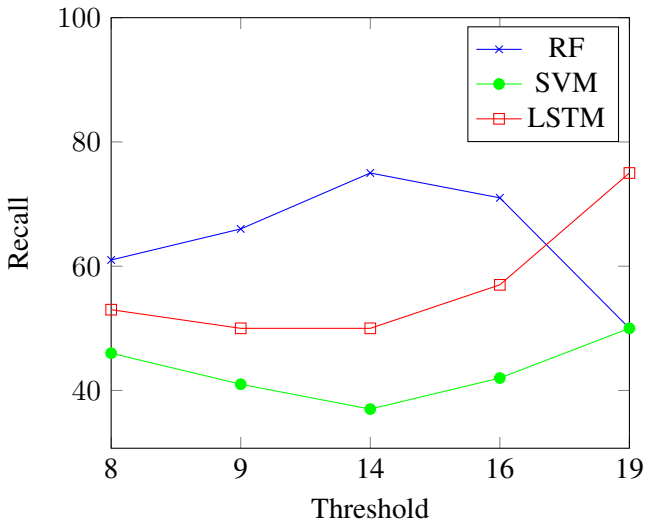


Figure 4.8.: Recall of ML classifiers we used to determine argumentative relations in the "unmerged" version of the factual text. Threshold t represents the number of times the relations have been identified by at least t annotators.

4.3.2. Mining BAFs from a dialogue

We set the task to extract a BAF (by identifying argumentative relations between arguments) from a dialogue excerpt taken from the film *Twelve Angry Men*, which revolves around the reasoning of twelve men that deliberate on a homicide trial. The dialogue excerpt used in the task is shown below:

$\mathbf{A_0}$ *The boy is not guilty.*
$\mathbf{A_1}$ *I never saw a guiltier man in my life. You sat right in court and heard the same thing I did. The man's a dangerous killer.*
$\mathbf{A_2}$ *He's 19 years old [too young to be guilty].*
$\mathbf{A_3}$ *Old enough. He knifed his own father, four inches into the chest. An innocent little 19-year old kid. They proved it a dozen different ways.*
$\mathbf{A_4}$ *It's not so easy for me to raise my hand and send a boy off to die without talking about it first.*
$\mathbf{A_5}$ *I think the guy's guilty. You couldn't change my mind if you talked for a hundred years.*
$\mathbf{A_6}$ *This boy's been kicked around all his life. You know, living in a slum, his mother dead since he was nine. That's not a very good head start. He's a tough, angry kid. I think maybe we owe him a few words [before considering him guilty].*

$\mathbf{A_7}$ *We don't owe him a thing. He got a fair trial. You're not going to tell us that we're supposed to believe him, knowing what he is. I've lived among 'em all my life. You can't believe a word they say.*
$\mathbf{A_8}$ *I just think he's guilty. I thought it was obvious. I mean nobody proved otherwise.*
$\mathbf{A_9}$ *Nobody has to prove otherwise. The burden of proof is on the prosecution. The defendant doesn't have to open his mouth. That's in the Constitution. The Fifth Amendment.*
$\mathbf{A_{10}}$ *At 10 minutes after 12 the old man who lived underneath the room where the murder took place heard loud noises in the upstairs apartment. He heard the kid say to his father, "I'm gonna kill you!" A second later he heard a body falling and saw the kid running down the stairs. Then he called the police. They found the father with a knife in his chest.*
$\mathbf{A_{11}}$ *And the coroner fixed the time of death at around midnight.*
$\mathbf{A_{12}}$ *The boy's story is flimsy. He claimed he was at the movies. That's a little ridiculous, isn't it? He couldn't even remember what pictures he saw.*
$\mathbf{A_{13}}$ *What about the woman across the street? If her testimony [to have seen the killing] don't prove [that the boy is guilty], then nothing does.*
$\mathbf{A_{14}}$ *[She saw the killing] through the windows of a passing elevated train [so testimony unreliable].*
$\mathbf{A_{15}}$ *And they proved in court that you can look through the windows of a passing train at night and see what's happening on the other side.*
$\mathbf{A_{16}}$ *I started to be convinced, with the testimony from those people across the hall. Didn't they say something about an argument between the father and the boy around 7 o'clock that night?*
$\mathbf{A_{17}}$ *8 o'clock. They heard the father hit the boy twice and then saw the boy walk angrily out of the house. [It doesn't prove that the boy killed him].*

4.3.2.1. Annotation task

We instructed the annotators that their annotations should lead to a BAF $\langle Args, \mathcal{L}^-, \mathcal{L}^+ \rangle$, with $Args = \{A_0, \ldots, A_{17}\}$, namely we instructed them that each paragraph in the dialogue gives exactly one argument. We also instructed the annotators that their BAFs should follow the discussion flow in which one argument supports or attacks another argument previously expressed in the dialogue

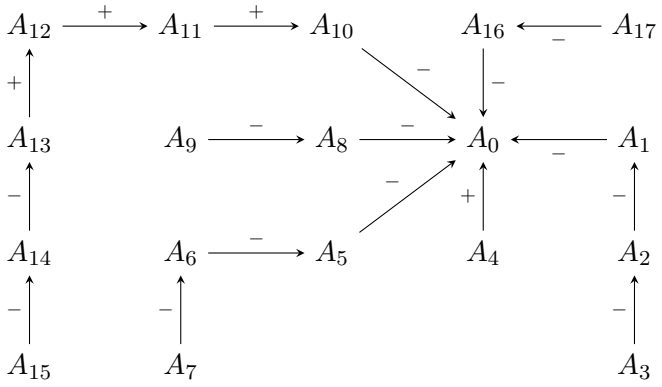


Figure 4.9.: Our manually extracted BAF from the excerpt dialogue.

and that the BAF should thus be "minimal" in the sense that each argument (except the first one) has one outgoing relation:

$$\forall a \in \mathit{Args} \setminus \{A_0\}\ \exists b \in \mathit{Args} \text{ s.t. } (a, b) \in \mathcal{L}^- \cup \mathcal{L}^+$$

In the case of an argument supporting and/or attacking more than one argument, to guarantee "minimality", the relation to the most recent argument only should be included. Let $<$ be a total order over Args s.t. $A_i < A_j$ iff $i < j$, for $i, j = 0 \ldots 17$. Then, the "minimality" requirement amounts to the following:

if $b_1 < b_2 < a$ then
  if $(a, b_2) \in \mathcal{L}^- \cup \mathcal{L}^+$
  then $(a, b_1) \notin \mathcal{L}^- \cup \mathcal{L}^+$

The reason for preferring relations from more recent arguments is that this allows to limit the number of argumentative relations in BAFs, and thus generate BAFs that can be more easily understood in settings with a vast amount of information such as debates or online comments.

Figure 4.9 shows the BAF that we extracted from the excerpt dialogue fulfilling the conditions set for the annotation task.

We obtained 22 annotations (21 annotators in addition to the BAF we extracted from the text). Table 4.11 shows the number of each type of relation identified by all annotators, including our annotation. Note that, whilst given clear instructions for annotation, there were a few annotators who did not follow the "minimality" requirement (i.e. $(A_0, A_1), (A_2, A_4), (A_6, A_7)$ violate the requirement). Note also that some relations were identified as both *attack* and *support* (e.g. (A_{17}, A_{16})).

	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	A_{16}	A_{17}
A_0		$1a$																
A_1	**20a**																	
A_2	**15s**	**11a**			$1s$													
A_3	$2a$	$3s$	**22a**															
A_4	**6s**	$2a$	$2s$	**6a** $1s$														
A_5	**10a**	$4s$			**10a** $1s$													
A_6	**6s**	$1a$			**9s**	**10a**		$1a$										
A_7	$1a$	$1s$				$2s$	**21a**											
A_8	**18a**	$3s$		$1s$		$1s$	$2a$	$1s$										
A_9	$1s$				$1s$				**22a**									
A_{10}	**13a** $1s$	$4s$								$4a$								
A_{11}	$1a$	$1s$									**18s**							
A_{12}	**11a**					$1s$		$3s$		$1a$	$4s$	$4s$						
A_{13}	**12a**	$4s$ $1a$		$1s$		$1s$				$1s$	$2s$		$2s$					
A_{14}	$1s$													**21a** $1s$				
A_{15}	$1a$													$4s$	**18a** $1s$			
A_{16}	**10a** $1s$	$4s$								$1a$	$3s$					$1s$		
A_{17}	$4s$											$1a$					**12a** $10s$	

Table 4.11.: The relations (*a* for *attack* and *s* for *support*) and the number of times they have been indicated by the annotators.

As it is the case with natural language, text can be interpreted differently, resulting in differences in annotations. One problem that arises in the annotation of this particular excerpt and which represents one of the main reasons for disagreement is ambiguity. One reason might be the different granularities considered when identifying the arguments from each paragraph. The most controversial pair of arguments is (A_{17} and A_{16}). 12 annotators identified an *attack* relation whereas 10 annotators identified a *support* relation. In all other cases of doubly annotated pairs, one of the annotations is dominant.

Here again, we focus only on the relations on which at least 6 annotators have agreed as 6 represents more than a quarter of the number of annotators, and use majority voting to select the relation type when double annotations occurred. Thus the relation between A_{17} and A_{16} was chosen to be *attack*. All pairs that where doubly annotated can be seen as examples of ambiguous relations: some annotators considered A_{16} to mean "I started to be convinced *that he is not guilty*" instead of reading it as "Didn't they say something about an argument... I started to be convinced *that he is guilty*".

We used Fleiss' kappa and Krippendorff's alpha to compute the inter-annotator agreement. Both methods gave an agreement of 0.457, showing moderate agreement between the annotators. We also computed the number of relations identified by each annotator compared to the relations identified by at least 6 of the remaining annotators, yielding an average of 60.45% agreement. The relations (and the number of times each relation has been identified) are shown in Figure 4.10.

4.3.2.2. Results

Whilst the classifiers identify other relations on top of the ones identified by annotators, we are interested in comparing the argumentative relations identified automatically with the ones identified by annotators. The recall of the methods we experimented with is shown in Figure 4.11. RFs perform the worst, with best recall performance of 22%. This is because RFs fail to learn how to classify *attack* relations. This can also be seen in the graph as all relations that have been identified by at least 20 annotators are *attacks* and RFs fail to classify these correctly. On the other hand, SVMs achieve good results classifying all relations as *attack*, which is the dominant type of relation in our BAF. The LSTM-based method is the one that performs the best, being able to classify both *attack* and *support* relations with recall above 80%.

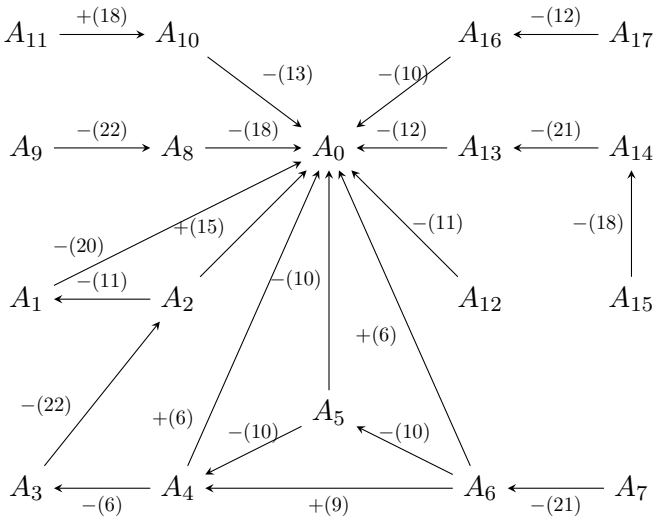


Figure 4.10.: Argumentative relations and the number of times these have been annotated.

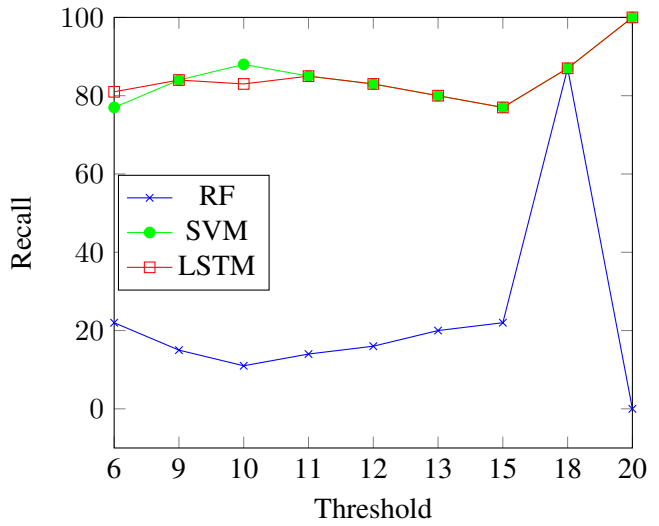


Figure 4.11.: Recall of ML classifiers we used to determine argumentative relations. Threshold t represents the number of times the relations have been identified by at least t annotators.

The dialogue is one example of a real world application where the model for identifying the argumentative relations and hence construct the BAF can be applied. It illustrates how difficult the RbAM task can be as different granularities for the arguments need to be considered when analysing the entire paragraph. Whilst in some cases the paragraph could have been split into several arguments, we opted to set each paragraph as a single argument. Even in this case, some arguments were interpreted differently by annotators, e.g. the relation between A_{17} and A_{16}. Additionally, some arguments are even more complex. For example, A_{12} contains the reported argument of the defendant, the juror's attack on that argument, a rhetorical question expressing a counter-conclusion and then a premise for that counter-conclusion. Whilst it is a very difficult example of an argument with many subtleties, our technique is analysing the entire argument as a single entity.

As it is a difficult task for humans, we can also expect the model to have some difficulties in classifying all types of relations correctly. Whilst our focus here was on recall in order to compare with annotators, our LSTM-based model is also able to classify argumentative relations that had not been identified by annotators. For example, our network shows that A_{16} *supports* A_{10}, and that A_{14} *supports* A_6. Whether identifying all possible relations between all arguments is better than extracting a "minimal" BAF is left for future work.

4.4. Summary and Discussion

We proposed a LSTM-based model to address the 3 class RbAM problem that can be deployed "in the wild" for different applications, where text is unstructured.

We deployed the LSTM-based model to extract BAFs from a factual text and from an excerpt dialogue from the film *Twelve Angry Men* to test whether trained classifiers can achieve comparable outcomes to those of humans. The techniques analysed phrases as single (argumentative) entities, disregarding individual components such as rhetorical questions and counter-conclusions. Whilst given instructions to consider phrases as single arguments, some annotators overlooked these instructions, and this was noticed in the type of relations identified that contained those arguments.

We observed that annotating argumentative relations is a difficult task for human annotators, even for trained ones [150] as highlighted by some of the issues identified during annotation phase. Future work includes testing the approaches on other texts as our aim is to develop classifiers that achieve comparable results to human

annotators in a wide range of settings. These classifiers could then be used to create more corpora, highly needed for experimentation. Further work also includes identifying the arguments relevant for the issue considered [190] and defining a way to combine them rather than assuming that arguments are represented by single phrases.

We have seen the transition of 3 class RbAM from a stand-alone task to a prerequisite task used "in the wild" to construct BAFs that we then compared with the ones extracted by annotators from a factual text and from a dialogue. We will see in the next chapter a further transition for RbAM, in particular how RbAM can be applied to mine BAFs in order to benefit applications.

5. Using mined Bipolar Argumentation Frameworks in applications

In this chapter we propose methods for extracting BAFs from texts using 3 class RbAM and use them in two applications. We give the motivation for this work in Section 5.1. We mine BAFs to determine movie scores and provide explanations for recommending movies in Section 5.2. We mine BAFs to determine and to provide explanations for the 'goodness' of hotels and show how BAFs can contribute argumentative features that improve ML classifiers in detecting deceptive reviews in Section 5.3. We summarise the research in this chapter and propose directions for future work in Section 5.4.

5.1. Motivation

In Chapter 4, we have considered 3 class RbAM as a stand-alone task and then as a means of constructing BAFs that we compared with the ones extracted by annotators. Here, we take one step further and use 3 class RbAM to mine BAFs that can benefit applications. We will see the applications of BAFs mined from text and their advantages in two settings.

For the final two types of experiments we turn to (social) media. The use of social media has become a regular habit for many and has changed the way people interact with each other. In an age in which e-commerce and audio/video streaming are dominant markets for consumers, products' online reviews are fast becoming the preferred method of quality control for users. The aggregation of these reviews allows users to check the quality of a product while avoiding reviews which may be incoherent, irrelevant or even malicious.

Within the movie domain, Rotten Tomatoes[1] (RT) is a popular review site that aggregates critics' reviews to obtain an overall percentage of critics who like the movie and critics who do not. The RT score is simplified to a binary classification

[1] rottentomatoes.com

for the movie of Fresh or Rotten, based on whether it is greater or equal to 60% or not, respectively. This simplification into RT score, Fresh/Rotten classification, gives users a quick way to determine whether a movie is worth watching or not. However, the 60% threshold means that a critic's mixed review that is slightly positive overall will have the same weight as a rave review from another critic, leading to the case where a movie with a maximum RT score could be composed of only *generally* positive reviews. Also, the RT score does not take into account user preferences and so factors which decrease the RT score may not have any relevance in a user's personal selection criteria, meaning movies may be overlooked when they may actually be perfectly suited to a user's tastes. Thus, a method to *explain* the aggregation is needed so that users can decide for themselves.

To this end, we define a novel form of argumentative dialogical agent (ADA) for mining arguments from reviews by means of NLP techniques (Sentiment Analysis and the best performant LSTM-based model as reported in Section 4.2.3) applied to extract a Quantitative BAF (QBAF) for any chosen movie with gradual semantics for QBAFs for deriving a dialectical strength measure for movies, as an alternative to the RT score. We show that ADA provides a dialectical strength which is comparable with the RT score and supplements a movie's strength with dialogical explanations obtained from BAFs extracted from movie reviews that empower users to interact with the system for more information about a movie's aggregated review.

Detecting deceptive reviews is an important problem and has recently received a great deal of attention (see [47] for an overview of the ML techniques and features used to detect review spam). It has an effect in e-commerce as deceptive reviews may persuade potential customers to buy a company's product/service (if they are positive) or discourage customers from purchasing (if they are negative). Some reviews may be maliciously written by competitors in order to defame a company's products or to promote their own products/services. The state-of-the-art in this context is to extract features from reviews using standard syntactic analysis given by NLP when using ML techniques [47]. When dealing with large texts, e.g. drawn from online product reviews or online debates, analysing every possible pair of pieces of texts in order to determine relations is simply not feasible.

We propose a methodology to extract BAFs using the best performant LSTM-based model as reported in Section 4.2.3 from natural language text where the arguments that form the BAFs are clustered based on the topics extracted from the text being analysed. In order to construct BAFs, we use a temporal dimension to

decide *which texts to compare* to determine the relation as well as to determine *the type of relation* between texts (i.e. support, attack, or neither). Thus, we construct topic-dependent BAFs by determining the relations between arguments that refer to the same topic, following the temporal order, in such a way that more recent texts can relate to less recent ones, but not vice versa. The rationale behind the topic-oriented approach is that arguments that mention different topics are highly unlikely to be related (i.e. neither argument supports nor attacks the other argument). The rationale behind the reliance on a temporal ordering is that it allows to limit the number of relations in BAFs. Our illustration demonstrates that this approach allows to generate BAFs that summarise texts, thus being more easily understood, and providing *explanations* about the 'goodness' of the item being reviewed (hotels in our running example). In addition, we focus on analysing whether reviews are deceptive by analysing the interaction or the influence that these texts have on the others, thus exploiting contextual information. We experiment with the use of argumentative features extracted using quantitative gradual semantics from BAFs along Random Forests in two domains (hotels and restaurants), using the gold standard datasets [115, 139] and show that combining DL techniques for RbAM with argumentative reasoning improves results compared to standard ML classifiers.

Our work draws primarily on RbAM and argumentative reasoning with AFs in order to (1) mine the BAFs and hence provide the interaction of arguments that can be used to determine how a product's overall score was produced in review websites and (2) improve ML classifiers by creating new features from evaluating the mined BAFs to detect deceptive reviews. Additionally, we make first steps in addressing a significant problem of systems, which is their inability to explain their outputs as the reviews are aggregated to obtain an overall percentage of users who like the item under review and users who do not. In particular, we provide:

- dialogical explanations that describe the strengths of movies obtained from BAFs extracted from text. Dialogical argumentation has been studied in various settings, for example: dialogue games for argumentation [158], persuasion dialogue [159], and strategic argumentation [87].
- explanations based on aspect strengths that describe an item (in our case hotels) obtained from BAFs extracted from text. In the same spirit, Pazienza et al. [147] introduced Bipolar Weighted Argumentation Frameworks with attack and support relations to which weights, negative weights for attacks and positive weights for supports, are assigned. In contrast to this work, we

assign strengths to arguments and use quantitative gradual semantics (see Section 2.3.2) to evaluate the graph, differently from the acceptability semantics used in [147].

Whilst we illustrate our methodologies in two domains, movies in Section 5.2 and hotels in Section 5.3, our approach can in principle be applied to other settings.

5.2. Mining Bipolar Argumentation Frameworks to determine movie scores

We define an argumentative dialogical agent (ADA) for mining arguments from movie reviews. ADA relies upon a *feature*-based characterisation of reviews and mines *review aggregations* (RAs) from snippets drawn from reviews by critics to obtain *votes* (on both movies and their (sub-)features) that are in turn used to generate QBAFs. We mine RAs from snippets using NLP methods, namely RbAM by deploying the best performant LSTM-based model as reported in Section 4.2.3. We also experiment with Sentiment Analysis (SA) for obtaining votes as an alternative to RbAM in order to compare the two approaches. ADA uses NLP to determine whether snippets provide positive or negative votes for ((sub-)features of) movies, by looking for arguments supporting or attacking, respectively, the (features of the) movies. An overview of ADA can be seen in Figure 5.1.

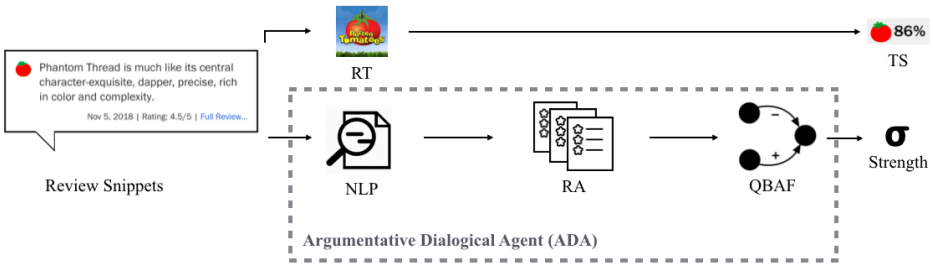


Figure 5.1.: An overview of ADA for extracting aggregated scores from (snippets from) reviews, compared with the original RT method of computing a RT score (the Tomato Score - TS in the figure) as an aggregation method.

5.2.1. Feature-based characterisation of movies

Definition 5.2.1. Let $\mathcal{M}$ be a given set of movies, and $m \in \mathcal{M}$ be any movie. A *feature-based characterisation of* m is a finite set $\mathcal{F}$ of *features* with *sub-features*

$\mathcal{F}' \subset \mathcal{F}$ such that each $f' \in \mathcal{F}'$ has a unique *parent* $p(f') \in \mathcal{F}$; for any $f \in \mathcal{F} \backslash \mathcal{F}'$, we define $p(f) = m$.

A sub-feature is more specific than its parent feature. For example, for the movie $m =$ *Wonder Wheel*[2], a feature may be *acting*, which may be the parent of the sub-feature *Kate Winslet*. We will refer to elements of $\mathcal{F} \backslash \mathcal{F}'$ only as features, and to elements of $\mathcal{F}'$ as sub-features. Also, we will refer to a sub-feature with parent f as a sub-feature of f. Note that this feature-based characterisation may be obtained automatically from metadata and from the top critics' snippets that appear on the RT movie pages. By doing so, for *Wonder Wheel*, we may obtain features $\{f_A, f_D, f_W, f_T\}$, where f_A is *acting*, f_D is *directing*, f_W is *writing* and f_T is *themes*[3].

The sub-features in $\mathcal{F}'$ may be of different types, namely *single* (for features f_D and f_W, if we only consider movies with a single director or writer) or *multiple* (for f_A, since movies will generally have more than one actor: *Wonder Wheel* has *Jim Belushi*, *Justin Timberlake* and *Kate Winslet* as sub-features of f_A, and f_T, since movies will generally be associated with several themes). In the case of single sub-features, the feature can be equated with the sub-feature (for *Wonder Wheel*, *Woody Allen* is the sole director and so this sub-feature can be represented by f_D itself). Furthermore, sub-features may be *predetermined*, namely obtained from meta-data (as for the sub-features with parents f_A, f_D, f_W in the running example), or *mined* from (snippets of) reviews (for *Wonder Wheel* the sub-feature *amusement park* of f_T may be mined rather than predetermined). To determine the mined sub-features of a movie we use semantic information such as the semantic network ConceptNet[4] to identify related terms for f_T. For example, for *Wonder Wheel*, the system automatically identifies the sub-feature *amusement park* (f'_{T1}) as several reviews mention the related terms *Coney Island* and *fairground*, as in *'like the fairground ride for which it's named, Wonder Wheel is entertaining'*.

Using this feature-based characterisation of a movie and snippets from the movie reviews by critics, ADA uses NLP to generate votes on arguments, amounting to the movie in question and its (sub-)features. The result is a *review aggregation* for the movie, which the agent then transforms into a *QBAF* that we obtain as described next.

[2] www.rottentomatoes.com/m/wonder_wheel/reviews/?type=top_critics

[3] In the experiments in Section 5.2.2 we limited the analysis to exactly these four features only as these are the ones that occur most frequently in the movie domain.

[4] conceptnet.io

5.2.1.1. Extracting Review Aggregations

Let $m \in \mathcal{M}$ be any movie and $\mathcal{F}$ be a feature-based characterisation of m as given in Definition 5.2.1. Let $Args$ denote $\{m\} \cup \mathcal{F}$, referred to as the set of *arguments*. We then define the following:

Definition 5.2.2. A *review aggregation* for m is a triple $\mathcal{R}(m) = \langle \mathcal{F}, \mathcal{C}, \mathcal{V} \rangle$ where

- $\mathcal{C}$ is a finite, non-empty set of *critics*;
- $\mathcal{V} : \mathcal{C} \times Args \to \{-, +\}$ is a partial function, with $\mathcal{V}(c, \alpha)$ representing the *vote* of critic $c \in \mathcal{C}$ on argument $\alpha \in Args$.

A positive/negative vote from a critic on a (sub-)feature of the movie signifies positive/negative stance on that (sub-)feature and a positive/negative vote on m signifies positive/negative stance on the overall movie.

A review aggregation can be *augmented* by exploiting the parent relation: a vote for/against an argument can be seen as a vote for/against the argument's parent.

Definition 5.2.3. Given a review aggregation $\mathcal{R}_0(m) = \langle \mathcal{F}, \mathcal{C}, \mathcal{V}_0 \rangle$, an *augmented review aggregation* $\mathcal{R}(m) = \langle \mathcal{F}, \mathcal{C}, \mathcal{V} \rangle$ is such that for any $c \in \mathcal{C}$ and any $\alpha \in Args$:

- if $\mathcal{V}_0(c, \alpha)$ is defined, then $\mathcal{V}(c, \alpha) = \mathcal{V}_0(c, \alpha)$; else
- if $|\{\beta \in Args | p(\beta) = f_A \wedge \mathcal{V}_0(c, \beta) = +\}| > |\{\gamma \in Args | p(\gamma) = f_A \wedge \mathcal{V}_0(c, \gamma) = -\}|$ then $\mathcal{V}(c, f_A) = +$; else
- if $|\{\beta \in Args | p(\beta) = f_A \wedge \mathcal{V}_0(c, \beta) = +\}| < |\{\gamma \in Args | p(\gamma) = f_A \wedge \mathcal{V}_0(c, \gamma) = -\}|$ then $\mathcal{V}(c, f_A) = -$; else
- if $|\{\beta \in Args | p(\beta) = f_T \wedge \mathcal{V}_0(c, \beta) = +\}| > |\{\gamma \in Args | p(\gamma) = f_T \wedge \mathcal{V}_0(c, \gamma) = -\}|$ then $\mathcal{V}(c, f_T) = +$; else
- if $|\{\beta \in Args | p(\beta) = f_T \wedge \mathcal{V}_0(c, \beta) = +\}| < |\{\gamma \in Args | p(\gamma) = f_T \wedge \mathcal{V}_0(c, \gamma) = -\}|$ then $\mathcal{V}(c, f_T) = -$; else
- if $|\{\beta \in Args | p(\beta) = m \wedge \mathcal{V}(c, \beta) = +\}| > |\{\gamma \in Args | p(\gamma) = m \wedge \mathcal{V}(c, \gamma) = -\}|$ then $\mathcal{V}(c, m) = +$; else
- if $|\{\beta \in Args | p(\beta) = m \wedge \mathcal{V}(c, \beta) = +\}| < |\{\gamma \in Args | p(\gamma) = m \wedge \mathcal{V}(c, \gamma) = -\}|$ then $\mathcal{V}(c, m) = -$; else
- $\mathcal{V}(c, \alpha)$ is undefined.

In the case of f_A, which is a *multiple* feature, we also consider the augmented vote (if any) when determining whether the movie has augmented votes. This is because a movie generally has several actors, whose performances may differ. In the case of f_D and f_W, if we only consider movies with a single director or writer,

the features are equated with the sub-feature. If the movie has more than one director or writer, then their augmented votes are not considered when determining the augmented votes of the movie as the contributions in directing and writing cannot be split (i.e. we cannot have director X was better than director Y for the same movie). While being a *multiple* feature, we do not consider the augmented votes of themes for the augmented votes of movies as we mine themes from texts and movies may not have themes. The importance of acting in the augmentation is also due to the fact that it appears in movie metadata, whereas themes do not.

For example, let c's vote on f_A be undefined, c's vote on sub-feature f'_{A1} of f_A be $+$ and there be no $-$ votes from c on any other sub-features of f_A. We then assume that c's overall stance on f_A is positive and therefore set c's vote on f_A to +. This notion of augmented review aggregation combats the brevity of the snippets causing the review aggregation being too sparsely populated.

ADA uses NLP techniques to extract review aggregations, from which augmented review aggregations are then obtained automatically as in Definition 5.2.3. The NLP techniques are used for analysing each critic's review independently, tokenising each review into sentences and splitting sentences into phrases when specific keywords (*but, although, though, otherwise, however, unless, whereas, despite*) occur. Each phrase may then constitute a potential argument with vote from its critic in the review aggregation.

For illustration, consider the following review for m = *Wonder Wheel* from a critic:

c_1: *Allen, 82, has his ups and downs, and while there have been more downs than ups lately he is always worth the benefit of the doubt. But Wonder Wheel is a ride to nowhere*.

ADA extracts two phrases:

- p_1: *Allen, 82, has his ups and downs, and while there have been more downs than ups lately he is always worth the benefit of the doubt*
- p_2: *Wonder Wheel is a ride to nowhere*

A review comprising a single sentence such as:

c_2: *Despite a stunning performance by Winslet and some beautiful cinematography by Vittorio Storaro, but Wonder Wheel loses its charms quickly and you'll soon be begging to get off this particular ride*
may split into:

- p_3: *Despite a stunning performance by Winslet and some beautiful cine-*

matography by Vittorio Storaro

- p_4: *Wonder Wheel loses its charms quickly and you'll soon be begging to get off this particular ride*

Finally, consider the following review from a critic:

c_3: *As we watch Allen worry and nitpick over the way women fret over aging, painting ginny as pathetic, jealous, insecure, and clownish, it's dull, unoriginal, and offensive. Frankly, we've had enough Woody Allen takes on this subject.*

Here ADA extracts two different phrases concerning *Woody Allen*:

- p_5: *As we watch Allen worry and nitpick over the way women fret over aging, painting ginny as pathetic, jealous, insecure, and clownish, it's dull, unoriginal, and offensive*
- p_6: *Frankly, we've had enough Woody Allen takes on this subject.*

In order to determine the arguments on which the votes act, ADA uses a *glossary* G using movie-related words for each feature as well as for movies in general. G is as follows (for any $m \in \mathcal{M}$):
$G(m) = \{movie, film, work\}$;
$G(f_D) = \{director\}$;
$G(f_A) = \{acting, cast, portrayal, performance\}$;
$G(f_W) = \{writer, writing, screenplay, screenwriter, screenwriting, script, storyline, character\}$.

When determining the argument on which a vote acts, sub-features take precedence over features. A mention of "Kate Winslet" (f'_{A1}) (with or without a word from $G(f_A)$) connects with f'_{A1}, whereas a sole mention of any word from $G(f_A)$ connects with f_A. A text that contains two entities (a sub-feature or a word from the glossary) corresponding to different (sub-)features results in two arguments (and votes), one for each (sub-)feature identified.

Once arguments have been identified, votes on them can also be extracted by NLP techniques. In our experiments in Section 5.2.2, we experiment with and compare two alternative NLP techniques to determine votes. These techniques are SA and identifying attack/support by RbAM, as described next.

5.2.1.2. Mining votes using Sentiment Analysis

The sentiment polarity of each phrase is translated into a (negative or positive) vote from the corresponding critic. Furthermore, we impose a threshold on the sentiment polarity to filter out phrases that can be deemed to be "neutral" and therefore cannot be considered to be votes. Votes are then assigned to arguments based on occurrences of words from G.

For example, ADA may deem the phrase p_1 given earlier to have a polarity of 0.611 and to be assigned to f_D, therefore giving $\mathcal{V}(c_1, f_D) = +$. Similarly, p_2 may result in a sentiment/target argument pair of $(-0.518,\ m)$, therefore giving $\mathcal{V}(c_1, m) = -$. If the neutrality threshold is 0.6 for the absolute value of the polarity, a positive vote corresponding to p_1 is assigned to f_D, whereas the negative vote corresponding to p_2 is not assigned to m.

From the review from c_2, ADA may extract one vote for the sub-feature "Kate Winslet" (f'_{A1}) and one for the movie in general. Thus, p_3 gives $(0.833, f'_{A1})$ therefore $\mathcal{V}(c_2, f'_{A1}) = +$, while p_4 gives $(-0.604, m)$ therefore $\mathcal{V}(c_2, m) = -$. It should be noted that if a feature *cinematography* had been included in our $\mathcal{F}$ then we would have had another vote from c_2. This could be achieved by using more metadata of the movies and hence an occurrence of Storaro would correspond to a vote on cinematography. Additionally, we could have also opted for a more fine grained typology such as main vs supporting actors that could be obtained from metadata, but decided against this and opted to treat all actors the same as only a few are mentioned in the reviews and having this separation would make the critics votes sparse. We could also identify more features by using topic modelling and then aggregate similar terms to identify the clusters that lead to the top features (similar to the four we selected) or using other (meta-data) features related to movies such as Oscar categories (e.g. costume, music) and IMDb categories (e.g. box-office). However, in the RT setting, these are not topics which are commented on most often in review snippets and decided not to include them. However, if one of these topics stands out, we expect several critics to mention them in the snippet reviews and thus our system will identify them as *themes*.

If the review of a single critic results in several phrases associated with an argument with different polarities, ADA takes the one with the highest sentiment magnitude to determine the vote on that argument. For example, given: p_5: $(-0.659,\ f_D)$ and p_6: $(0.500,\ f_D)$, then p_5 supercedes and $\mathcal{V}(c_3, f_D) = -$. Here the sentiment for p_6 is incorrect but the neutrality threshold leads to it being ignored.

ADA determines the votes for the mined f_T in the same way as for the other features, when using SA. For example, given p_7: *like the fairground ride for which it's named, Wonder Wheel is entertaining* leading to $(0.741, f'_{T1})$, ADA obtains $\mathcal{V}(c, f'_{T1}) = +$.

5.2.1.3. Mining votes using Relation-based Argument Mining

We use the best performant LSTM-based model as shown in Section 4.2.3 to identify attack/support/neither attack nor support relations between chunks of text, where attack leads to a negative vote and support to a positive vote. For example, consider the feature f_A and a sentence that mentions acting. RbAM may determine whether there is an argumentative relation between the sentence and the pre-set argument f_A, which may be read as the natural language argument: *the acting was good*, and this relation can be used to give a vote. For illustration, considering the earlier phrases extracted from critic c_1, RbAM may give that p_1 supports f_D, and therefore $\mathcal{V}(c_1, f_D) = +$, and p_2 attacks m, and therefore $\mathcal{V}(c_1, m) = -$. In this setting, neutrality (of the form discussed for SA) is given by the third relation (neither attack not support).

5.2.1.4. Formulating QBAFs from Review Aggregations

In order to obtain a QBAF from a review aggregation, we determine: the arguments, the arguments' base scores, and between which arguments attacks and supports are present. Having already identified the arguments, we use an aggregation of critics' votes for base scores, and we impose that a (sub-)feature attacks or supports its parent argument depending on its aggregated stance, as follows:

Definition 5.2.4. Let $\mathcal{R}(m) = \langle \mathcal{F}, \mathcal{C}, \mathcal{V} \rangle$ be any (augmented) review aggregation for $m \in \mathcal{M}$. For any $\gamma \in Args = \mathcal{F} \cup \{m\}$, let $\mathcal{V}^+(\gamma) = |\{c \in \mathcal{C} | \mathcal{V}(c, \gamma) = +\}|$ and $\mathcal{V}^-(\gamma) = |\{c \in \mathcal{C} | \mathcal{V}(c, \gamma) = -\}|$. Then, the *QBAF corresponding to* $\mathcal{R}(m)$ is $\langle Args, \mathcal{L}^-, \mathcal{L}^+, \tau \rangle$ such that

$$\mathcal{L}^- = \{(\alpha,\beta) \in \mathcal{F}^2 | \beta = p(\alpha) \wedge \mathcal{V}^+(\beta) > \mathcal{V}^-(\beta) \wedge \mathcal{V}^+(\alpha) < \mathcal{V}^-(\alpha)\} \cup$$
$$\{(\alpha,\beta) \in \mathcal{F}^2 | \beta = p(\alpha) \wedge \mathcal{V}^+(\beta) < \mathcal{V}^-(\beta) \wedge \mathcal{V}^+(\alpha) > \mathcal{V}^-(\alpha)\} \cup$$
$$\{(\alpha,m) | \alpha \in \mathcal{F} \wedge m = p(\alpha) \wedge \mathcal{V}^+(\alpha) < \mathcal{V}^-(\alpha)\};$$
$$\mathcal{L}^+ = \{(\alpha,\beta) \in \mathcal{F}^2 | \beta = p(\alpha) \wedge \mathcal{V}^+(\beta) \geq \mathcal{V}^-(\beta) \wedge \mathcal{V}^+(\alpha) \geq \mathcal{V}^-(\alpha)\} \cup$$
$$\{(\alpha,\beta) \in \mathcal{F}^2 | \beta = p(\alpha) \wedge \mathcal{V}^+(\beta) \leq \mathcal{V}^-(\beta) \wedge \mathcal{V}^+(\alpha) \leq \mathcal{V}^-(\alpha)\} \cup$$
$$\{(\alpha,m) | \alpha \in \mathcal{F} \wedge m = p(\alpha) \wedge \mathcal{V}^+(\alpha) \geq \mathcal{V}^-(\alpha)\};$$

$\tau(m) = 0.5 + 0.5 \cdot \frac{\mathcal{V}^+(m) - \mathcal{V}^-(m)}{|\mathcal{C}|}$ and $\forall f \in \mathcal{F}$, $\tau(f) = \frac{|\mathcal{V}^+(f) - \mathcal{V}^-(f)|}{|\mathcal{C}|}$.

An attack is defined as either from a feature with dominant negative votes (with respect to positive votes) towards the movie itself or from a sub-feature with dominant negative (positive) votes towards a feature with dominant positive (negative, respectively) votes. The latter type of attack can be exemplified by a sub-feature of f_A with positive stance attacking the negative (due to other votes/arguments) feature f_A, which attacks m. Conversely, a support is defined as either from a feature with dominant positive votes towards the movie itself or from a sub-feature with dominant positive (negative) votes towards a feature with dominant positive (negative, respectively) votes. The latter type of support can be exemplified by a sub-feature of f_A with negative stance supporting the negative feature f_A, which attacks m. It should be noted that (sub-)features with equal positive and negative votes are treated as supporters, though we could have assigned no relation.

We adapt the base score, $\tau(m) \in [0, 1]$, from [161]. Intuitively, $\tau(m) = 1$ represents all critics having a positive stance on the movie while $\tau(m) = 0$ requires universally negative stance. The base score of a (sub-)feature f is again in $[0, 1]$ where, differently to movies since a feature already represents positive/negative sentiment towards the argument it supports/attacks, $\tau(f) = 0$ represents no dominant negative/positive stance from the critics on f while $\tau(f) = 1$ represents universally negative/positive stance on f.

5.2.2. Determining Rotten Tomatoes scores: experimental results

In this section, we evaluate empirically and qualitatively different realisations of ADA. In particular, we consider which method for vote mining (amongst SA and RbAM) and which gradual semantics for QBAFs (REB, QuAD, DF-QuAD) allow

for better correlation with the RT score. We chose these gradual semantics as, to the best of our knowledge, they are the only semantics in the literature that are applied to BAFs and have a base score (see Section 2.3.2 for more details about each of these semantics). The base score is an important aspect as it gives the flexibility of assigning or computing an intrinsic value for each argument. This is beneficial in applications with which users can interact as it enables the users to change the base score of arguments based on their own preferences, thus adapting the (general) BAFs to their own preferences.

5.2.2.1. Experimental Results

We tested our proposed method on box-office movies from January 2015 to August 2018 inclusive and on the top 100 movies of all time[5], giving a total of 1281 movies after removing those which did not have reviews. We conducted experiments to determine the performance of our predicted ratings compared to the RT score using both SA and RbAM techniques. We report the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) on the scale 0-100, along with scaling the score to a 0-9 point scale to reflect the 'grade' of the movie (i.e. $7 = [70, 79]$ and $9 = [90, 100]$). We also show the classification report of predicting whether a movie is certified Fresh or Rotten, where movies with RT score ≥ 60 are classified as Fresh and movies with RT score < 60 are classified as Rotten. In all of our experiments, we analyse movies for which we obtain votes from at least 33% of the critics who reviewed the movies, removing those for which the NLP techniques mined few votes, resulting in approximately 900 movies. In particular, there are 35219 snippets that we analyse (which totals 840874 tokens) leading to 33554 arguments to be considered.

The first experiment uses SA, for which we deployed an off-the-shelf classifier on movie reviews [143]. The second experiment uses RbAM, for which we deployed the best performant LSTM-based model as reported in Section 4.2.3. The MAE and RMSE can be seen in Table 5.1. Here we are interested in obtaining an error which is close to 0.

Since ratings are highly subjective and a review that can be deemed to be slightly positive will be classified as Fresh in RT, the same as a highly positive review, we focus on the MAE score where individual differences are weighted equally in the average. Using SA, DF-QuAD is the best performing gradual semantics for

[5] rottentomatoes.com/top/bestofrt/

		% Scale			Grade Scale		
		MAE	RMSE	PCC	MAE	RMSE	PCC
SA	QuAD	16.004	19.306	0.731	1.535	1.925	0.708
	DF-QuAD	15.978	19.329	0.729	1.527	1.927	0.707
	REB	17.530	20.296	0.752	1.693	2.005	0.725
RbAM	QuAD	16.404	19.662	0.714	1.582	1.960	0.689
	DF-QuAD	16.343	19.677	0.712	1.573	1.961	0.687
	REB	18.125	20.911	0.734	1.735	2.045	0.712

Table 5.1.: Errors between strength and Rotten Tomatoes score for SA and RbAM.

QBAFs, followed by QuAD and REB in turn. It achieves 15.978 MAE on the scores and 1.527 on the 'graded' scores. Similarly, using RbAM, DF-QuAD is the best performing semantics, followed by QuAD and REB in turn. It achieves 16.343 MAE on the scores and 1.573 on the 'graded' scores. The errors for RbAM are slightly higher than those computed using SA. One of the reasons may be that the LSTM filters 'neutral' relations between sentences from reviews and the argument for the feature they are associated with, which, using SA, were deemed to be positive/negative. We also report the Pearson Correlation Coefficient (PCC) which showed similar results across all of the semantics.

Table 5.2 shows the classification report. Here there seems to be less difference between SA and RbAM, with RbAM performing better than SA by 0.01 F_1 for the Fresh class when using QuAD, 0.01 F_1 for both classes when using DF-QuAD, and no difference in F_1 when using REB. In terms of gradual semantics, F_1 shows a slight advantage of REB over (DF-)QuAD.

The mined sub-features of f_T are an important aspect of our QBAF (see Table 5.3 for examples). These sub-features can modify a movie's score by increasing or decreasing it based on the phrases connected to them. For example, these sub-features strengthened *Call Me by Your Name* from 77 to 88 (RT score 98), *Phantom Thread* from 88 to 90 (RT score 98), and *The Godfather* from 78 to 90 (RT score 95).

From a computational point of view, the strengths can be calculated in polynomial time (since the QBAFs we determine by SA and RbAM are acyclic). When a new review is added, the pre-trained SA and RbAM methods simply extend the QBAF. The real computational cost is in training the SA and RbAM models, which is however completed upfront, and offline.

			Precision	Recall	F_1
SA	QuAD	Fresh	0.84	0.93	0.88
		Rotten	0.80	0.62	0.70
	DF-QuAD	Fresh	0.84	0.94	0.88
		Rotten	0.81	0.59	0.68
	REB	Fresh	0.88	0.88	0.88
		Rotten	0.74	0.74	0.74
RbAM	QuAD	Fresh	0.85	0.92	0.89
		Rotten	0.78	0.63	0.70
	DF-QuAD	Fresh	0.85	0.93	0.89
		Rotten	0.79	0.61	0.69
	REB	Fresh	0.89	0.88	0.88
		Rotten	0.73	0.75	0.74

Table 5.2.: Classification report using SA (682 Fresh and 306 Rotten movies) and RbAM (620 Fresh and 267 Rotten movies). The difference in the number of movies considered is attributed to filtering, using different methods, of 'neutral' relations in SA and RbAM respectively, which results in a movie being discarded if it had not received votes from at least 33% critics.

5.2.2.2. Qualitative Assessment

Focusing on the DF-QuAD algorithm, which showed the smallest error (MAE) with respect to the RT score, we obtain movie scores close to the RT score such as *RBG* (86 vs RT score 90), exact scores such as *ET The Extraterrestrial* (97), and also results in contrast to the RT score such as *The Greatest Showman* (79 vs RT score 36). One reason for this is the errors which occur in extracting votes. For example, from the review: *The director, Michael Gracey, delivers quick doses of excitement in splashy scenes but has little feel for the choreographic action* we only obtain *...doses of excitement in splashy scenes*, resulting in a positive vote. Another is the failure to identify any votes from reviews such as the following (negative) text: *In a broader sense, the mishmash does recall the real Barnum, who once sewed half a fish to half an ape and called it a mermaid*, as it does not mention any (sub-)feature or opinion about the movie itself. With further developments of our method (e.g. anaphora resolution), the frequency and impact of these errors will be reduced.

The minor differences between QuAD and DF-QuAD are due to the fact that they only differ in certain situations (see [162] for more details). The greater accuracy of (DF-)QuAD compared to REB is due to the fact that attackers or supporters will, in general, have less of an effect on an argument's strength and so its strength is *shifted* towards its base score, when compared with (DF-)QuAD. Such

Movie	Theme	Keywords
All Eyez on Me	rap music	hip-hop, rapper
Lady Bird	adolescence	adolescence, coming of age
The Greatest Showman	entertainment	impresario, amusement park
A Quiet Place	terror	fright, terror, dread, taut, suspense
Call Me By Your Name	affection	first love, love, affection, romance
Godfather	mafia	don, mafia
I, Tonya	competitor	competition, outsider
Get Out	race	black people, of color
La La Land	relationship	rapport, intimacy, one another
Moana	melody	riff, song
Phantom Thread	clothing	clothe, fashion, dress, designer

Table 5.3.: Examples of movies' extracted themes.

a behaviour is not desired in this setting. For example, for *Lady Bird*, which has strong supporters, the RT score, QuAD, DF-QuAD and REB results are 100, 92, 92 and 85, respectively. Another issues with REB is that there is a *blind spot* in the strength results. If we express the algorithm in the form $y = 1 - \frac{1-\tau(\alpha)^2}{1+\tau(\alpha)\cdot e^x}$ we can see that as x approaches ∞ (the balance of reasoning moves in favour of support), $y = 1$, while as x approaches $-\infty$ (the balance of reasoning moves in favour of attack), $y = \tau(\alpha)^2$, that means that the attackers cannot weaken the argument further than the square of its base score. Thus with REB, as the votes on an argument become stronger, the argument becomes increasingly resistant to the weakening effect of negative reasoning. This also causes an asymmetry between attackers and supporters, diminishing the effects of attackers compared to supporters which is not desirable in this setting.

Table 5.2 shows that REB, with both SA and RbAM, has a slight advantage over (DF-)QuAD when considering F_1. In terms of Precision, REB is more accurate than (DF-)QuAD for the Fresh class but not for the Rotten class, which is due to REB's shifting effect towards the base score. Given that the Rotten class comprises arguments with a strength of less than 0.6 and that the mid-point of the base score

is 0.5, this means that REB wrongly shifts some movies into the Rotten class while those it classes as Fresh are more likely to be correctly classified. The Recall results back up this finding since REB correctly classifies many more Rotten movies than (DF-)QuAD.

5.2.3. Explanations from mined Bipolar Argumentation Frameworks: an illustration

In Section 5.2.1.4 we have shown how the dialogical agent ADA extracts arguments that amount to a given movie and its (sub-)features which are then transformed into a QBAF. In this section we show how ADA uses the obtained QBAFs to extract dialogical explanations for the strengths of the movies (which correlate with their RT score) obtained via gradual semantics of these frameworks and engages human users within dialogical exchanges about the RT score for movies, using these explanations. An overview of how ADA incorporates explanations is shown in Figure 5.2.

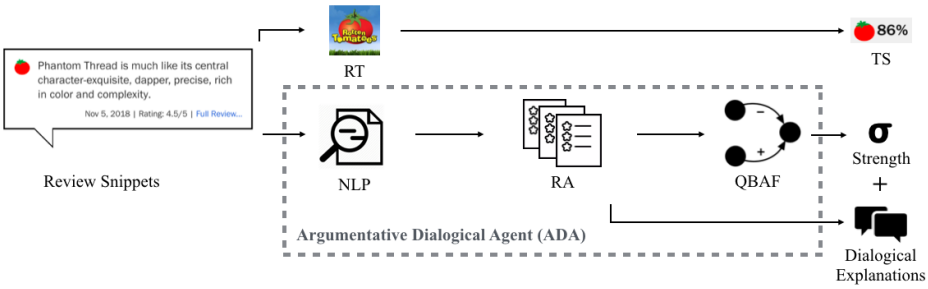


Figure 5.2.: An overview of ADA for extracting aggregated scores with dialogical explanations from (snippets from) reviews. TS (Tomato Score) in the figure represents the RT score. We add Dialogical Explanations to Figure 5.1.

We illustrate our methodology when applied to the reviews in Table 5.4, which represent an excerpt from the top critics reviews of the movie *Inception*[6].

[6] www.rottentomatoes.com/m/inception/reviews/?type=top_critics

$\mathbf{R_1}$ A spectacular fantasy thriller based on Nolan's own original screenplay, it is the smartest CGI head-trip since The Matrix.
$\mathbf{R_2}$ A heist film of thrilling, almost delirious complexity.
$\mathbf{R_3}$ Inception is a boldly constructed wonder with plenty of – as one describes it – "paradoxical architecture."
$\mathbf{R_4}$ Inception is that rare film that can be enjoyed on superficial and progressively deeper levels, a feat that uncannily mimics the mind-bending journey its protagonist takes.
$\mathbf{R_5}$ Mr. DiCaprio exercises impressive control in portraying a man on the verge of losing his grip, but Mr. Nolan has not, in the end, given Cobb a rich enough inner life to sustain the performance.
$\mathbf{R_6}$ In this smart sci-fi puzzle box, director Christopher Nolan transports the audience to a dreamscape that begins with the familiar and then takes a radical, imaginative leap.
$\mathbf{R_7}$ In this wildly ingenious chess game, grandmaster Nolan plants ideas in our heads that disturb and dazzle. The result is a knockout. But be warned: it dreams big.
$\mathbf{R_8}$ At first, Inception left me cold, feeling as if I'd just eavesdropped on somebody's bad acid trip. Now I find I can't get the film out of my mind, which is really the whole point of it, isn't it?
$\mathbf{R_9}$ A devilishly complicated, fiendishly enjoyable sci-fi voyage across a dreamscape that is thoroughly compelling.

Table 5.4.: Reviews for the movie *Inception*.

From $\mathbf{R_5}$ the system identifies the following potential votes, with polarity as indicated:

- *Mr. DiCaprio exercises impressive control in portraying a man on the verge of losing his grip* (0.61)
- *Mr. Nolan has not, in the end, given Cobb a rich enough inner life to sustain the performance* (-0.5)

From $\mathbf{R_7}$ the system identifies the following potential votes, with polarity as indicated:

- *In this wildly ingenious chess game, grandmaster Nolan plants ideas in our heads that disturb and dazzle* (0.8)
- *The result is a knockout* (0.5)
- *be warned: it dreams big* (0.4)

From $\mathbf{R_8}$ the system identifies the following potential votes, with polarity as indicated:

- *At first, Inception left me cold, feeling as if I'd just eavesdropped on some-*

body's bad acid trip (-0.7)

- *now I find I can't get the film out of my mind, which is really the whole point of it, isn't it?* (0.8)

Determining the argument on which the votes act is driven by a glossary of movie terms associated to each of the m, f_A, f_D, f_W, with f_T being an exception as themes are inferred from the text. Themes are automatically identified from texts. Here we have the theme *dream* as we have occurrences of the work *dreamscape*[7].

From $\mathbf{R_5}$, the system keeps the vote with polarity 0.61, discarding the other one due to the threshold. From $\mathbf{R_7}$, the system keeps only the first vote due to threshold. Note that if the other two phrases had polarity above threshold, the system would still have discarded them as they do not mention any movie terms. From $\mathbf{R_8}$, as the system identifies two phrases with different polarities associated to the same argument (i.e. movie), it takes the one with the highest sentiment magnitude (i.e. 0.8).

We focus on sentiment polarity to determine positive/negative votes. The votes and their associated arguments can be seen in Table 5.5. Note that a_x represents the argument extracted from review x where a_x is unique for each (sub-)feature.

Augmentation of overall stance towards the sub-features occurs when there is no vote from a review coming from a critic to a feature but there exist votes to its sub-features. For example, a_1 is directed at f_D and is augmented to m. a_5 is directed at "Leonardo DiCaprio" and is augmented to f_A. In addition, a_5 is also augmented to m.

Figure 5.3 shows the QBAF extracted using the arguments from Table 5.5 and applying DF-QuAD to compute arguments' strength.

We have focused on the movie *Inception* to illustrate some of the decisions we take in extracting votes as well as how augmentation works. In the remaining, we focus on different movies, that can provide more interesting explanations.

A user may interact with ADA by requesting an explanation of an argument (movie or (sub-)feature).

Definition 5.2.5. Given any (augmented) review aggregation $\langle \mathcal{F}, \mathcal{C}, \mathcal{V} \rangle$ of any $m \in \mathcal{M}$ and corresponding QBAF $\langle Args, \mathcal{L}^-, \mathcal{L}^+, \tau \rangle$ with strength σ, an *argumentation dialogue* between a user and ADA consists of *explanation requests*

[7]Note that when analysing the entire set of critics reviews, we have occurrences of *dream*, *dreamscape*, *dreaming*, all related to the *dream* theme.

Feature	Vote id	Vote
movie	a_2	a heist film of thrilling, almost delirious complexity (+)
	a_3	Inception is a boldly constructed wonder with plenty of – as one describes it – ”paradoxical architecture” (+)
	a_4	Inception is that rare film that can be enjoyed on superficial and progressively deeper levels, a feat that uncannily mimics the mind-bending journey its protagonist takes (+)
	a_8	now I find I can't get the film out of my mind, which is really the whole point of it, isn't it (+)
	a_1 a_5 a_6 a_7	augmented (+) augmented (+) augmented (+) augmented (+)
director Chris Nolan	a_1	a spectacular fantasy thriller based on Nolan's own original screenplay, it is the smartest cgi head-trip since the Matrix (+)
	a_6	in this smart sci-fi puzzle box, director Christopher Nolan transports the audience to a dreamscape that begins with the familiar and then takes a radical, imaginative leap (+)
	a_7	in this wildly ingenious chess game, grandmaster Nolan plants ideas in our heads that disturb and dazzle (+)
writer Chris Nolan	a_1	a spectacular fantasy thriller based on Nolan's own original screenplay, it is the smartest cgi head-trip since the Matrix (+)
acting	a_5	augmented (+)
Leonardo DiCaprio	a_5	Mr. DiCaprio exercises impressive control in portraying a man on the verge of losing his grip (+)
themes	a_6	augmented (+)
	a_9	augmented (+)
dream	a_6	in this smart sci-fi puzzle box, director Christopher Nolan transports the audience to a dreamscape that begins with the familiar and then takes a radical, imaginative leap (+)
	a_9	a devilishly complicated, fiendishly enjoyable sci-fi voyage across a dreamscape that is thoroughly compelling (+)

Table 5.5.: Votes (and their polarity) extracted from the reviews in Table 5.4.

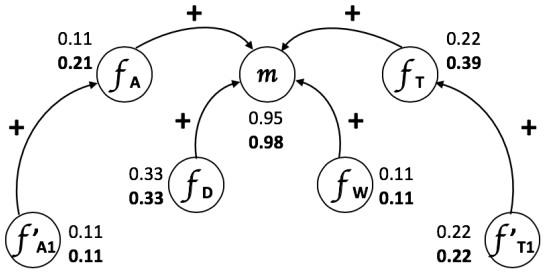


Figure 5.3.: QBAF for the movie *Inception* with base scores (normal font) and DF-QuAD strengths (bold font).

$\mathcal{Q}(\alpha)$ for $\alpha \in Args$ from the user, to which ADA responds with *explanations* $\mathcal{X}(\alpha)$.

For illustration we define a simple argumentation dialogue.

Definition 5.2.6. Let $r_a^+, r_a^-, r_b^+, r_b^-$ be functions giving *positive primary*, *negative primary*, *positive secondary* and *negative secondary* (respectively) phrases about an argument, as in Figure 5.4. For any $S \subseteq Args$, if $S = \emptyset$ let $max(S) = \emptyset$; else, let $max(S) = argmax_{s \in S}\sigma(s)$.

$r_a^+(\gamma) = \{$because (the) γ was/were great$\}$;
$r_a^-(\gamma) = \{$because (the) γ was/were poor$\}$;
$r_b^+(\gamma) = \{$although (the) γ was/were great$\}$;
$r_b^-(\gamma) = \{$although (the) γ was/were poor$\}$;
$r_a^+(\emptyset) = r_a^-(\emptyset) = r_b^+(\emptyset) = r_b^-(\emptyset) = \{\}$.

Figure 5.4.: Functions $r_a^+, r_a^-, r_b^+, r_b^-$, for any $\gamma \in Args$

Then, a *simple argumentation dialogue* is such that for any $\alpha \in Args$:

if $\alpha =$ m and $\sigma(\alpha) < 0.6$ and $\exists \beta \in \mathcal{L}^-(\alpha) \cup \mathcal{L}^+(\alpha)$ s.t. $\sigma(\beta) > 0$:
$\mathcal{Q}(\alpha) = \{$Why was α poorly rated?$\}$
$\mathcal{X}(\alpha) = \{$This movie was poorly rated$\}+$
$r_a^-(max(\mathcal{L}^-(\alpha))) + r_b^+(max(\mathcal{L}^+(\alpha)))$; else

if $\alpha =$ m and $\sigma(\alpha) \geq 0.6$ and $\exists \beta \in \mathcal{L}^-(\alpha) \cup \mathcal{L}^+(\alpha)$ s.t. $\sigma(\beta) > 0$:

$\mathcal{Q}(\alpha) = \{$Why was α highly rated?$\}$

$\mathcal{X}(\alpha) = \{$This movie was highly rated$\}+$

$r_a^+(max(\mathcal{L}^+(\alpha))) + r_b^-(max(\mathcal{L}^-(\alpha)))$; else

if $\alpha \in$ $\mathcal{F}$ and $\mathcal{V}^+(\alpha) < \mathcal{V}^-(\alpha)$ and $\exists \beta \in \mathcal{L}^-(\alpha) \cup \mathcal{L}^+(\alpha)$ s.t. $\sigma(\beta) > 0$:

$\mathcal{Q}(\alpha) = \{$Why was/were (the) α considered to be poor?$\}$

$\mathcal{X}(\alpha) = \{$(The) α was/were considered to be poor$\}+$

$r_a^-(max(\mathcal{L}^-(\alpha))) + r_b^+(max(\mathcal{L}^+(\alpha)))$; else

if $\alpha \in$ $\mathcal{F}$ and $\mathcal{V}^+(\alpha) \geq \mathcal{V}^-(\alpha)$ and $\exists \beta \in \mathcal{L}^-(\alpha) \cup \mathcal{L}^+(\alpha)$ s.t. $\sigma(\beta) > 0$:

$\mathcal{Q}(\alpha) = \{$Why was/were (the) α considered to be great?$\}$

$\mathcal{X}(\alpha) = \{$(The) α was/were considered to be great$\}+$

$r_a^+(max(\mathcal{L}^+(\alpha))) + r_b^-(max(\mathcal{L}^-(\alpha)))$; else

if $\mathcal{V}^+$ $(\alpha) < \mathcal{V}^-(\alpha)$ and $\nexists \beta \in \mathcal{L}^-(\alpha) \cup \mathcal{L}^+(\alpha)$ s.t. $\sigma(\beta) > 0$:

$\mathcal{Q}(\alpha) = \{$What did critics say about (the) α being poor?$\}$

$\mathcal{X}(\alpha) = \{[p$ from $c \in \mathcal{C}$ constituting $\mathcal{V}(c, \alpha) = -]\}$; else

if $\mathcal{V}^+$ $(\alpha) \geq \mathcal{V}^-(\alpha)$ and $\nexists \beta \in \mathcal{L}^-(\alpha) \cup \mathcal{L}^+(\alpha)$ s.t. $\sigma(\beta) > 0$:

$\mathcal{Q}(\alpha) = \{$What did critics say about (the) α being great?$\}$

$\mathcal{X}(\alpha) = \{[p$ from $c \in \mathcal{C}$ constituting $\mathcal{V}(c, \alpha) = +]\}$.

The intuition behind this definition is that a simple explanation of the reasoning for each argument's strength may consist of its strongest attacker and its strongest supporter linked by *because* and *although* connectives, depending on whether the argument is Fresh or Rotten in the case of movies or had mostly positive or negative votes in the case of features. If an argument has no attackers or supporters, we use critics' phrases (which constitute the votes towards the argument's base score) to explain its strength. It should be noted that this form of explanation is only possible due to the use of the QBAF mechanism. Other formalisms or se-

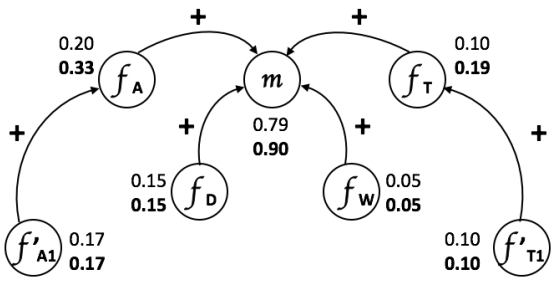


Figure 5.5.: QBAF for the movie *Phantom Thread* with base scores (normal font) and DF-QuAD strengths (bold font).

mantics may lead to different explanations which may be better suited to different contexts. For example, the *Propagation Semantics* [22], defined for AAFs (restrictions of QBAFs comprising only arguments and attacks) are defined globally, i.e. an indirect (not immediate) attacker can have as much influence as a direct attacker. In this case, the main justification may be an argument which is not in the set of direct attackers or supporters.

For example, consider the QBAF in Figure 5.5 for the movie *Phantom Thread* (RT score=98). Here, all four features are supporters of the movie and therefore represent positive reasoning for the strength, with *Daniel Day-Lewis* (f'_{A1}) supporting the movie's strongest feature, f_A. A simple argumentation dialogue between a user and ADA, may then be as follows:

user: *Why was Phantom Thread highly rated?*
ADA: *This movie was highly rated because the acting was great.*
user: *Why was the acting considered to be great?*
ADA: *The acting was considered to be great because Daniel Day-Lewis was great.*
user: *What did critics say about Daniel Day-Lewis being great?*
ADA: *"...Daniel Day-Lewis remains our greatest actor..."*
user: *Why were the themes considered to be great?*
ADA: *The themes were considered to be great because (the theme) clothing was great.*
user: *What did critics say about (the theme) clothing being great?*
ADA: *"...it's set in an evocative ecosphere of haute couture fashion..."*

This dialogue is fairly repetitive given this movie's almost universally positive reviews (as well as the argumentation dialogue being fairly simple for illustration).

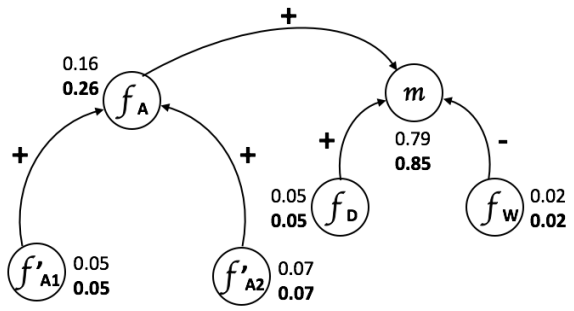


Figure 5.6.: QBAF for the movie *The Post* with base scores (normal font) and DF-QuAD strengths (bold font).

Consider the QBAF for *The Post* (RT score=91) in Figure 5.6. This QBAF has a different structure from that for *Phantom Thread* as no themes were mined for this movie, there is an attack and it has two sub-features supporting f_A. We can see that f_D positively contributed very little and f_W was actually considered to be poor since it attacks m. However, the acting from *Tom Hanks* (f'_{A1}) and, particularly, *Meryl Streep* (f'_{A2}) contributed to the high strength. The argumentation dialogue may then be:

user: *Why was The Post highly rated?*
ADA: *This movie was highly rated because the acting was great, although the writing was poor.*
user: *Why was the acting considered to be great?*
ADA: *Its acting was considered to be great because Meryl Streep was great.*
user: *What did critics say about Meryl Streep being great?*
ADA: *"...Streep's hesitations, rue, and ultimate valor are soul-deep..."*

User studies may prove to be useful to decide on the best way to provide the explanations, e.g. graphical, visual or linguistic, for different contexts and users, as in [76]. Further work on generating dialogical explanations instead of the current, rather simple but still easy to understand, templates for explanations, could lead to a more evolved dialogue between ADA and users. There are a plethora of ways in which more complicated argumentation dialogues could be defined to give more interesting and varied interactions. For example, the strength scale could be separated to a greater degree and attackers/supporters other than those with the maximum strength could be considered for different levels of sentiment, rather

than simply *great/poor*, e.g. *the writing was okay but wasn't great*; functions giving phrases could depend upon the type of argument for a more natural sounding phrase, e.g. for an actress: *Meryl Streep put in an excellent performance*; or arguments could be considered in tandem, e.g. *the acting was exceptional thanks to Tom Hanks and, particularly, Meryl Streep*.

5.3. Mining Bipolar Argumentation Frameworks from reviews for deception detection

We propose a methodology for mining BAFs from reviews that, evaluated using quantitative gradual semantics contributes a novel *argumentative feature*. This argumentative feature, when used in combination with other features in standard supervised classifiers, outperforms the latter in the task of detecting whether reviews are genuine or deceptive.

Consider the following two reviews about a hotel which form the BAF as shown in Figure 5.7.

$\mathbf{R_1}$: *I recommend the hotel, it has nice rooms*

$\mathbf{R_2}$: *The room was very dusty*

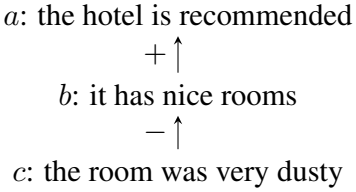


Figure 5.7.: Example BAF. Here, b supports a (indicated as a + edge) and c attacks b (indicated as a - edge).

The BAFs extracted from reviews can be seen as arguments for evaluating the 'goodness' of the item being reviewed. Once a BAF has been extracted from text, we can use argumentative reasoning to evaluate the arguments in the BAF and in particular the dialectical 'goodness' of an item being reviewed. For example, for the BAF in Figure 5.7, the dialectical strength of argument a is greater when having only the support from argument b compared to having the support from b which is, in turn, attacked by argument c.

An overview of how RbAM and argumentative reasoning and features are used for detecting deceptive reviews in our system is given in Figure 5.8. BAFs are extracted using RbAM. The dialectical strength of these BAFs computed using DF-QuAD contributes new argumentative features (arg features in the Figure) which, along with other syntactic features previously identified in studies of deception, are fed into Random Forests (RF) to determine whether a review is truthful or deceptive. We chose DF-QuAD as a gradual semantics due to the fact that it addresses the discontinuity problem of QuAD (in QuAD, adding a weak supporter/attacker to an argument with very strong attackers/supporters will increase/decrease its strength abruptly to a high/low value [162]), and it does not have the issue of REB that has an asymmetry between attackers and supporters as explained in Section 5.2.2.

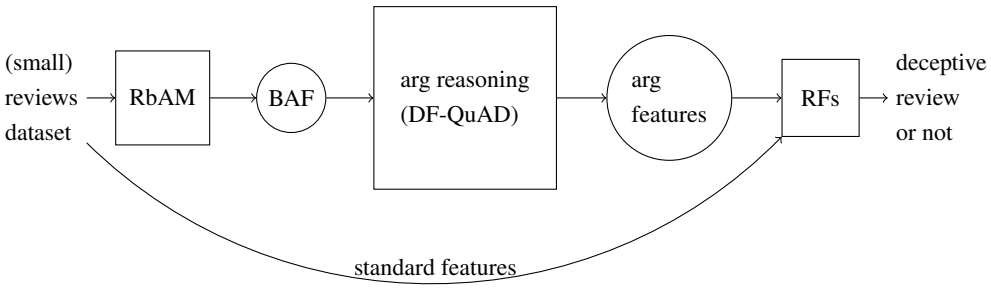


Figure 5.8.: Overview of how RbAM and argumentative (arg) reasoning and features are used for detecting deceptive reviews.

Our new argumentative features capture the impact of each review on determining how 'good' an item is with respect to all reviews about that item. Thus, our argumentative features can be seen as adding a semantic layer to the analysis of reviews as it uses information from discourse and the wider context represented by the other reviews about that item. We show experimentally that combining DL for RbAM and argumentative reasoning outperforms standard supervised ML techniques in detecting deceptive reviews while also being able to handle the small datasets problem.

5.3.1. Mining Bipolar Argumentation Frameworks methodology

Our approach to detecting deceptive reviews is based on mining BAFs that are clustered based on the topics extracted from reviews. Here, we automatically ex-

tract topics from reviews instead of relying on meta-data. We explore different approaches for identifying topics in reviews, ranging from associating each noun encountered in reviews to a topic, to more advanced techniques related to topic modelling such as Latent Dirichlet Allocation (LDA) [19] and Non-negative Matrix Factorization (NMF) [111]. We compare two methods for RbAM used for constructing the BAFs: a supervised classifier which uses syntactic and semantic features, and the best performant LSTM-based model described in Section 4.2.3. The BAFs extracted from the reviews will serve to provide new argumentative features which are then used, along with other features, to determine whether a review is deceptive or not.

5.3.1.1. Building a topic-dependent BAF

The procedure for constructing a topic-dependent BAF is described below:

1. Split each review into sentences (where each sentence is a potential argument)
2. Identify topics in reviews and the sentences (potential arguments) related to each topic
3. For each topic, run the RbAM classifier on the sentences associated to this topic to determine the relations between them
4. Construct BAF

This process is done for the set of reviews associated to each *item* in the review (i.e. the hotel dataset contains reviews for 20 hotels, we run this process for each hotel). We assume that each argument extracted from the reviews is contained in a sentence. Thus, each review is mapped to one or more such arguments.

5.3.1.2. Identifying arguments

The first step amounts to identifying parts of the reviews that form the BAF's arguments. Identifying arguments in text is a complex task in general, and may require identifying components and boundaries of arguments [118]. Here we simply opt for equating sentences and potential arguments. Concretely, the system splits each review into sentences with a pre-trained tokenizer for English from *nltk* [18]. Sentences containing *but*, *although*, *though*, *otherwise*, *however*, *unless*, *whereas*, *despite*, are split since generally the phrases before and after these separators express different sentiments (e.g. *The staff was nice but the room was messy* results in two sentences with different sentiments). The sentiment polarity of each sentence is

determined using Sentiment Analysis from the *pattern.en* module [50] which uses a lexicon of frequently used adjectives in product reviews annotated with scores for sentiment polarity.

For example, consider the following reviews about some hotel H:

$\mathbf{R_1}$: *It had nice rooms but terrible food*[8]

$\mathbf{R_2}$: *Their service was amazing and we absolutely loved the room. They do not offer free Wi-Fi so they expect you to pay to get Wi-Fi...*

From $\mathbf{R_1}$ the system extracts the following arguments, with polarity as indicated:

$a_{1,1}$: *It had nice rooms* (+)
$a_{1,2}$: *(It had) terrible food* (-)

whereas from $\mathbf{R_2}$ we obtain:

$a_{2,1}$: *service was amazing* (+)
$a_{2,2}$: *absolutely loved the room* (+)
$a_{2,3}$: *they do not offer free Wi-Fi so they expect you to pay to get Wi-Fi* (-)[9]

5.3.1.3. Clustering arguments by topics

The second step amounts to grouping potential arguments resulting from the first step by topic, to facilitate identifying (and render more meaningful) relations of attack and support between these arguments in the third step. Topic extraction can be performed in many alternative ways, including associating the lemmatized nouns from the reviews to topics (e.g., given the example reviews, topics may be *room, food, service, Wi-Fi*), LDA and NMF. We choose to deploy LDA and NMF, as these are able to better uncover the underlying semantic structure of a set of documents by identifying (words that belong to) topics. LDA is a generative probabilistic model for discrete data. We use LDA with online variational Bayes algorithm. Each document is represented as a bag of words where the vocabulary

[8] In our notation, $a_{x,y}$ represents the yth argument from review x .
[9] Note that we use components of argumentative sentences to stand for the full sentences. For example, $a_{1,1}$ stands for *The hotel was good as it had nice rooms.*

represents the set of words in the dataset. Being a probabilistic graphical model, we only need term count features for LDA. NMF finds two non-negative matrices (W, H) whose product approximates the non-negative matrix X representing the corpus of documents. Given N documents and M words in the vocabulary, $X = W \cdot H$ where the number of topics $K \ll N, M$, $X \in \mathbb{R}^{N \times M}$, $W \in \mathbb{R}^{N \times K}$, and $H \in \mathbb{R}^{K \times M}$. We compute the TF-IDF features and we use NMF with 0.1 regularization to optimise the squared Frobenius norm.

For both LDA and NMF, we automatically remove stop words and ignore the terms that appear in only one document or in at least 95% of the documents. In the experiments described in Section 5.3.2, we identify 35 topics using LDA and NMF and select the top 25 words for each topic. We have tried different values for the number of topics as well subjectively analysing some of the topics extracted to see if they are coherent. We have opted for the number of topics and words per topic that gave the best performances in the experiments. Other techniques could be used in order to select model parameters (e.g. heuristic approaches [205]).

After having identified topics, the system identifies the sentences/arguments related to these topics. We use $Args_t$ to refer to the arguments related to topic t. In the case of topics being associated to nouns, we extract the sentences that contain that specific topic/noun. For LDA/NMF, we extract the sentences that contain any of the top words associated to the topics extracted by these methods. We focus on the arguments related to topics and leave out the arguments related only to the item as we are interested in the *reasons* why the item is good/bad and not just the opinion (i.e. *The hotel is good* does not provide any reasons whereas *The hotel is good, the location is central* gives a reason for the hotel being good which is the location). Here as well, we discard phrases that do not mention any topic identified. With further developments of our method (e.g. anaphora resolution), the frequency and impact of discarding phrases that should have been considered will be reduced.

5.3.1.4. Determining argumentative relations

The third step amounts to determining dialectical relations between any pair of arguments associated with the same topic. In order to limit the number of comparisons, for each topic, we assume that a newer argument (with respect to time) can either *support*, *attack*, or *neither support nor attack* a previous argument, but not vice versa. Here we use the best performant model as reported in Section 4.2.3.

5.3.1.5. BAFs from reviews

The fourth and final step amounts to constructing topic-dependent BAFs from reviews, for the purpose of assessing how 'good' the item being reviewed is by assessing the dialectical strength of a special argument G (for 'good') in the constructed BAFs. In turn, each topic t identified at step 2 gives a special argument G_t (for 'good as far as t is concerned') supporting G. Thus, intuitively, the stronger the various G_t, in the computed BAFs, the stronger G. In addition to these special arguments G_t supporting G, the BAFs also include relations between arguments related to topic t drawn from reviews and G_t, so that a newer argument (with respect to time) can either *support*, *attack*, or *neither support nor attack* a previous argument or G_t, but not vice versa. If an argument a_t, related to topic t, does not support nor attack another argument related to t from the same or some other review, as determined by RbAM at step 3, then this argument a_t will either support or attack G_t, according to its polarity as determined by Sentiment Analysis.

Definition 5.3.1. Let the arguments mined from the set of reviews $\mathbf{R}$ with the set of topics $\mathfrak{T}$ be $Args = \{\alpha_{x,y} | 1 \leq x \leq |\mathbf{R}|, y \geq 1\} \cup \{G\} \cup \{G_t | t \in \mathfrak{T}\}$. For all $t \in \mathfrak{T}$, let $Args_t$ be the set of arguments related to topic t and let $<$ be a total order over $Args_t$ defined by $a < b$ iff a is temporally ordered before b, and let $\mathfrak{s}(Args_t) = a_1, a_2, \ldots, a_{n-1}, a_n$ be a temporally ordered list of arguments in $Args_t$. Let $\mathfrak{f}$ be the classifier that determines the argumentative relation of attack (0), support (1), neither attack nor support (2), and $\mathfrak{p}$ be sentiment classifier that determines the positive ($+$) or negative ($-$) polarity. The BAF resulting for $\mathbf{R}$ and $\mathfrak{T}$ is $\langle Args, \mathcal{L}^-, \mathcal{L}^+ \rangle = \textbf{Algorithm}(\mathfrak{T}, Args, <)$ such that:

Algorithm

input:$(\mathfrak{T}, Args, <)$

output:$\langle Args, \mathcal{L}^-, \mathcal{L}^+ \rangle$

$\mathcal{L}^+ := \{(G_t, G) | t \in \mathfrak{T}\}$

if	$\mathfrak{f}(a_n, a_{n-1}) = 0$	then $\mathcal{L}^- := \mathcal{L}^- \cup \{(a_n, a_{n-1})\}$
else if	$\mathfrak{f}(a_n, a_{n-1}) = 1$	then $\mathcal{L}^+ := \mathcal{L}^+ \cup \{(a_n, a_{n-1})\}$
else if	$\mathfrak{f}(a_n, a_{n-2}) = 0$	then $\mathcal{L}^- := \mathcal{L}^- \cup \{(a_n, a_{n-2})\}$
else if	$\mathfrak{f}(a_n, a_{n-2}) = 1$	then $\mathcal{L}^+ := \mathcal{L}^+ \cup \{(a_n, a_{n-2})\}$
...		
else if	$\mathfrak{f}(a_n, a_1) = 0$	then $\mathcal{L}^- := \mathcal{L}^- \cup \{(a_n, a_1)\}$
else if	$\mathfrak{f}(a_n, a_1) = 1$	then $\mathcal{L}^+ := \mathcal{L}^+ \cup \{(a_n, a_1)\}$
else if	$\mathfrak{p}(a_n) = -$	then $\mathcal{L}^- := \mathcal{L}^- \cup \{(a_n, G_t)\}$
else	$\mathfrak{p}(a_n) = +$	then $\mathcal{L}^+ := \mathcal{L}^+ \cup \{(a_n, G_t)\}$

For example, given reviews $\mathbf{R_1}$ and $\mathbf{R_2}$ from Section 5.3.1.2 and using nouns from reviews as topics as in Section 5.3.1.3, we obtain the BAF $\langle Args, \mathcal{L}^-, \mathcal{L}^+ \rangle$ with:

$$\begin{aligned} Args = &\{G, G_{\text{room}}, G_{\text{food}}, G_{\text{service}}, G_{\text{Wi-Fi}}, a_{1,1}, a_{1,2}, a_{2,1}, a_{2,2}, a_{2,3}\} \\ \mathcal{L}^- = &\{(a_{1,2}, G_{\text{food}}), (a_{2,3}, G_{\text{Wi-Fi}})\} \\ \mathcal{L}^+ = &\{(a_{2,2}, a_{1,1}), (a_{1,1}, G_{\text{room}}), (a_{2,1}, G_{\text{service}}), \\ &(G_{\text{room}}, G), (G_{\text{food}}, G), (G_{\text{service}}, G), (G_{\text{Wi-Fi}}, G)\} \end{aligned}$$

shown graphically in Figure 5.9.

We have imposed that arguments from more recent reviews can attack or support only arguments from less recent reviews or the special G_t arguments, rather than any arguments, independent of the order in which the reviews arose. We believe that this is legitimate, as it mimicks what humans experience when they write a review. In this case, they have full access to all previous reviews thus being able to agree, disagree or neither agree nor disagree with these reviews. This choice is also practical, allowing to limit the comparisons performed by RbAM. It would be interesting to experiment with BAFs obtained from reviews without imposing the temporal restriction over comparisons, to check in particular whether the resulting

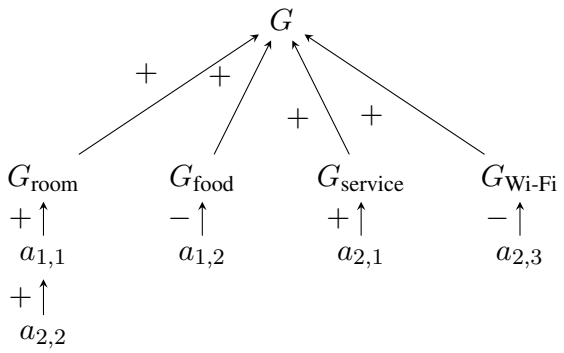


Figure 5.9.: Topic-dependent BAF obtained from $\mathbf{R_1}$, $\mathbf{R_2}$ in Section 5.3.1.

BAFs could provide more effective argumentative features.

5.3.1.6. From BAFs to argumentative features

In order to detect deceptive reviews, in addition to standard features used in previous studies, we associate argumentative features to each review, representing the impact of the review on how 'good' an item (e.g. hotel or restaurant) is with respect to all reviews about that item. These new features are obtained from measuring the strength of arguments in the BAF built from all reviews related to the chosen item and in the BAF built from all reviews for that item except the one whose impact we aim at determining.

The BAFs obtained from sets of reviews, as described in Section 5.3.1.1, are, by construction, guaranteed to be in the restricted form of sets of trees. Note that these trees may have any (finite) breadth when choosing topics based on the nouns identified in the reviews or, in our specific set-up, breadth 35 when determining topics using LDA/NMF, and any depth as determined by the relations between arguments extracted from reviews. We then compute the strengths of arguments in the BAF built from all reviews for that item except the one whose impact we aim at determining.

For illustration, consider the BAF extracted earlier from reviews $\mathbf{R_1}$, $\mathbf{R_2}$ (see Figure 5.9). We assume a base score $\tau(a) = 0.5$ for all $a \in \mathit{Args}$ (we will use this same base score for all arguments in our experiments). The impact of review r is then given by the difference between the measure of how 'good' the hotel/restaurant is deemed to be given all reviews R and how 'good' it is deemed to be given $\mathbf{R} \setminus \{r\}$.

In our example, if $\mathbf{R} = \{\mathbf{R_1}, \mathbf{R_2}\}$, to calculate the impact of $\mathbf{R_1}$ requires re-

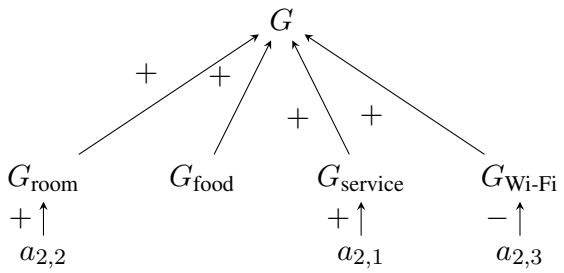


Figure 5.10.: BAF obtained from removing (arguments from) $\mathbf{R_1}$.

moving from our earlier BAF all arguments from $\mathbf{R_1}$, giving the BAF shown in Figure 5.10. The strength of G can be seen as a measure of how 'good' the product is deemed to be according to the reviews under consideration. Note that $a_{2,2}$ supports G_{room} as its polarity is positive and there is no previous argument that $a_{2,2}$ can be compared with.

5.3.2. Detecting deceptive reviews: experimental results

The gold standard for deceptive reviews consists of positive and negative hotel reviews of 20 Chicago hotels [139] extended more recently to include deceptive reviews written by domain experts (employees) and Amazon Mechanical Turkers, and truthful reviews written by customers from three domains: hotels, restaurants and doctors [115]. Existing studies have focused on detecting deceptive hotel reviews [140], identifying positive and negative deceptive hotel reviews [139] and cross-domain deception on the more recent data set [115].

The hotel dataset that we use consists of 1600 positive and negative reviews from this gold standard about 20 Chicago hotels: 400 truthful positive reviews from TripAdvisor, 400 truthful negative reviews from 6 online review websites, 400 deceptive positive reviews and 400 deceptive negative reviews from Turkers [139]. The restaurant dataset that we use consists of 400 reviews about 10 restaurants, 200 deceptive reviews and 200 truthful reviews [115].

We report the classification results on the task of determining whether a review is truthful or false in two domains, hotel and restaurant. We evaluate the performance of various techniques of extracting topics from reviews as presented in Section 5.3.1.3 and the impact our novel argumentative features have on the classifier's performance. All the results are obtained using 5 fold cross-validation and an ensemble method, RFs, with 10 trees in the forest and gini impurity criterion.

As baseline, we extract features used previously in studies of deception. These can be divided into two main groups: features related to the review and features related to the reviewer [68, 97, 114, 170]. Several types of features have been identified quantity, specificity, diversity, non-immediacy, as well as task specific features such as affect, expressivity, complexity, uncertainty and informality [71, 207]. As we only have the reviews and no information about users who wrote the reviews, we focus only on the features related to the review. These features are the result of POS tag analysis using *nltk* and are summarised in Table 5.6.

Category	**Features**
Personalization	Number of self references Number of 2nd person pronouns Number of other references Number of group pronouns
Quantity	Number of sentences Number of words Number of nouns Number of verbs
Complexity	Average sentence length Average word length
Diversity	Lexical
Uncertainty	Number of modal verbs Number of modifiers

Table 5.6.: Features and the associated category.

Additionally, we include TF-IDF features obtained from all reviews. To calculate these, we use the lemmas obtained by analysing the lowercase form of words and their POS tag.

We present results of different approaches of constructing BAFs from reviews and including the argumentative features related to the impact each review has on the 'goodness' of the item (hotel or restaurant) being reviewed. We experimented with two techniques for topic modelling, LDA and NMF (hence having features representing the impact of each review on the 'goodness' of the item being reviewed for each of these methods respectively).

For each approach of topic modelling, we identify the topic that has the highest probability of being associated to the review, as well as all the topics with probability greater than 0.2 of of being associated to the review. In both cases, we extract the sentences that contain any of the top words related to the topics that have been

associated to the review. To identify the relations between arguments associated to each topic, we chose the best performing LSTM-based model as reported in Section 4.2.3. We also report results when using the topic-noun approach and an RF classifier with features shown in Table 4.6. For each method of constructing the BAF, we create a new argumentative feature from computing the difference between the strength of arguments from all reviews and the strength of the arguments from all reviews except the one whose impact we aim at determining.

The classifiers' performances on the hotel dataset are shown in Table 5.7. We see that adding the TF-IDF features gives 76% accuracy, resulting in a dramatic improvement of 12% compared to the baseline where the syntactic features from Table 5.6 were used. Using argumentative features extracted from the BAF constructed from topics being associated to nouns in reviews and using RFs for RbAM yields lower results compared to using syntactic features and TF-IDF features, achieving 74.88% accuracy. Using argumentative features extracted from the BAF constructed using LDA and NMF for topic modelling and LSTMs for RbAM yields better results than using a standard classifier (RFs) and a simple topic extraction method (nouns $\sim$ topics) with accuracy 76.38%. Indeed, the best results are obtained using more advanced techniques for topic modelling rather than simple associations of topics $\sim$ nouns, and LSTMs for RbAM, with 0.38% improvement compared to using syntactic features and TF-IDF features.

Standard features	**TF-IDF**	**Argumentative features**	**Topic model**	**RFs**	
				Accuracy	$\mathbf{F_1}$
✓	✗	✗	✗	63.81	63.60
✓	✓	✗	✗	76.00	75.83
✓	✓	✓	Nouns + RFs	74.88	74.71
✓	✓	✓	LDA & NMF + LSTM	76.38	76.18
✗	✗	✓	LDA & NMF + LSTM	50.00	33.33

Table 5.7.: Classifiers' performances on the hotel dataset using 5 fold cross-validation.

The classifiers' performances on the restaurant dataset are shown in Table 5.8. Here as well, we see that adding the TF-IDF features results in a dramatic improvement of 9% compared to the baseline. In contrast to the hotel dataset, using argumentative features extracted from the BAF constructed from topics being associated to nouns in reviews and using RFs for RbAM results in an improvement of 1.5% compared to using only syntactic features. Using argumentative features extracted from the BAF constructed using LDA and NMF for topic modelling and LSTMs for RbAM also gives better results than using a standard classifier (RFs)

and a simple topic extraction method (nouns $\sim$ topics) with accuracy 72.5%. Here again, the best results are obtained using more advanced techniques for topic modelling and LSTMs for RbAM, with 2.75% improvement compared to using syntactic features and TF-IDF features.

Standard features	TF-IDF	Argumentative features	Topic model	RFs	
				Accuracy	$\mathbf{F_1}$
✓	✗	✗	✗	60.75	60.60
✓	✓	✗	✗	69.75	69.69
✓	✓	✓	Nouns + RFs	71.25	71.15
✓	✓	✓	LDA & NMF + LSTM	72.50	72.40
✗	✗	✓	LDA & NMF + LSTM	50.00	33.33

Table 5.8.: Classifiers' performances on the restaurant dataset using 5 fold cross-validation.

We showed that combining DL and argumentative reasoning outperforms standard ML techniques for deception detection in both domains, hotel and restaurant. The results show that argumentative reasoning can indeed be used to improve classifications.

We did not carry out any direct comparison with the results documented in the papers that introduced the reviews datasets we used [115, 139, 140] as the tasks we focused on were different from the ones in the original papers. Concretely, in [140] the experiments were run on a subset of the hotel dataset we used in this thesis whereas in [139] the focus was on classifier performances on the hotel reviews dataset based on the sentiment of the reviews (i.e. positive deceptive opinions and negative deceptive opinions). Furthermore, in [115] the focus was on classifier performances in cross-domain adaptation and on performances on intra-domain multi-class classification tasks, with the aim to classify reviews based on their source (reviews written by customers, employees, turkers). 81% F_1 was obtained in [17] using Convolutional Neural Networks. Whilst the same hotel dataset was used in their experiments, the dataset was randomly split into a 80:20 ratio for training and testing, and no cross-validation was performed, thus making it difficult to compare the results. On the hotel dataset, recent studies that have been published obtained better performance: 89% accuracy using Generative Adversarial Networks was obtained in [1] whereas 87.8% F_1 using SVMs and stratified cross-validation based on the sentiment polarity was obtained in [36]. No results are reported on the restaurant dataset. In this work, our motivation was to show the usefulness of *argumentative features* for this task rather than beating the state-

of-the-art for the deception task.

The datasets differ in size: the hotel dataset contains 1600 reviews whereas the restaurant dataset contains 400 reviews. We obtained improvements compared to the baseline when using more advanced topic modelling techniques and LSTMs to determine relations between arguments extracted from the reviews.

However, the datasets used are small. Moreover, the size of the restaurant dataset represents a quarter of the size of hotel dataset. We are interested to see whether, using a subset of the hotel dataset, we can still achieve comparable or better results compared to using the entire dataset to further explore the suitability of our methodology of combining DL for RbAM and argumentative reasoning by means of BAFs to cope with small datasets.

The results shown in Table 5.9 are obtained using 5 fold cross-validation. In all experiments we use LDA (as it has been shown that it learns more coherent topics than NMF [182]) to extract topics from reviews and the LSTM-based model. We see that, using a subset of the hotel dataset, we achieve better results combining DL with argumentative reasoning compared to using the entire dataset. We obtain 1% improvement in accuracy when using a quarter of the dataset, and 3% accuracy improvement when using three quarters or half of the hotel dataset. Using One-Way ANOVA, the result is significant at $p < 0.05$ (the f-ratio value is 114.71494, the p-value is < 0.00001).

Total number of reviews	**RFs**	
	Accuracy	$\mathbf{F_1}$
1600	76.69	76.48
1200	79.83	79.74
800	79.63	79.48
400	77.75	77.62

Table 5.9.: Classifier performance on the hotel dataset using 5 fold cross-validation on subsets of the dataset.

5.3.3. Explanations from mined Bipolar Argumentation Frameworks: an illustration

In Section 5.3.1.5 we have shown how we automatically extract arguments that amount to topics associated to a given hotel which are then transformed into a BAF. In this section we illustrate our proposed methodology for mining topic-

dependent BAFs from reviews about a hotel and show how the obtained BAFs can be used to derive explanations. The BAFs extracted from reviews can be seen as arguments for evaluating the 'goodness' of the item being reviewed and thus showing the reasons as to whether to recommend that product or service by providing an argumentation chain of other users' arguments and the strength of each topic identified in the reviews.

We apply our methodology on the reviews given in Table 5.10, which represent an excerpt from a dataset consisting of positive and negative hotel (deceptive or truthful) reviews [139].

From $\mathbf{R_1}$, the system identifies the following pieces of texts containing potential arguments, with polarity (+ for positive and - for negative sentiment) as indicated:

room was gorgeous (+)
the bar is very boring and the restaurant is not that great (-)

whereas from $\mathbf{R_{10}}$ the system identifies the following potential argument, with polarity as indicated:

located in a hard to find location, the restaurant is uncomfortably crowded, the staff is hard to reach (-)[10]

We will see next that some potential arguments can be split further.

For our reviews, if we associate the nouns encountered in *at least a few reviews* to a topic, then the system identifies the following topics: *staff*, *room*, *internet*, *bar*, *restaurant*, *location*. The intuition here is that, in the case of online reviews, if a topic is controversial/debatable, then it will be mentioned by at least a few users (either to support the argument given initially or to attack it).

The system then identifies the sentences/arguments related to these topics. In the case of topics being associated to nouns, the system extracts the sentences that contain these specific nouns. For example, from $\mathbf{R_1}$, the system identifies the following arguments, with polarity and topics as indicated:

room: $a_{1,1}$: *room was gorgeous* (+)

[10]Note that we use components of argumentative sentences to stand for the full sentences. For example, the first part of the latter potential argument stands for *"The hotel is located in a hard to find location in Chicago"*.

$\mathbf{R_1}$ This hotel is absolutely beautiful. Our room was gorgeous. I do have 1 complaint. The bar is very boring and the restaurant is not that great, but other than that I loved it. I would definitely stay there again.
$\mathbf{R_2}$ Very pleasant front staff, large rooms, and free Internet for those that are members of the loyalty program.
$\mathbf{R_3}$ Staff were helpful & friendly, room was huge with fantastic view of the river. Adjoining Bar/Restaurant, great food.
$\mathbf{R_4}$ I stayed here last August and I'm truly glad that I will never have to stay here again. The website does a great job of creating an illusion. The rooms are so much smaller than it seems on the website. The wireless internet is free but it is extremely slow. All and all, I would not recommend this hotel to anyone.
$\mathbf{R_5}$ It is one of the nicest hotels i have stayed at in my life, clean, comfortable and pretty. The rooms were clean and the staff is very caring.
$\mathbf{R_6}$ I have stayed in hotels all over the world, and this is probably the worst that I've ever experienced. The staff was unaccommodating, the front desk staff was condescending and not even remotely helpful. The room was not clean. Don't waste your time or your money here.
$\mathbf{R_7}$ The staff are polite and well poised. The rooms, hallways and facilities were exceptionally clean and tidy. During my stay, I stopped at their restaurant where I had one of the best American style meal in a while. Overall, this hotel is a place I would surely stay at again if given the chance to visit Chicago for a second time. It is truly exceptional.
$\mathbf{R_8}$ I loved the location and the amenities offered by this hotel. The room was charming with a window seat and a water view. Free wireless internet were a plus here. The staff was helpful and attentive. I would definitely stay here again.
$\mathbf{R_9}$ There was only one person from the staff at the front desk when we arrived, preoccupied with something on their computer so our presence was not acknowledged for several minutes. Then when we got to our room, I found it to be incredibly dusty. Overall it was a good stay but those two inconveniences made us question the amount of money we paid for it.
$\mathbf{R_{10}}$ The hotel is located in a hard to find location in Chicago, the restaurant is uncomfortably crowded, the staff is hard to reach, overall it was not a pleasant hotel stay.

Table 5.10.: Reviews for a hotel in Chicago.

bar: $a_{1,2}$: *the bar is very boring* (-)
restaurant: $a_{1,3}$: *the restaurant is not that great* (-)

whereas from $\mathbf{R_{10}}$ the system identifies:

location: $a_{10,1}$: *located in a hard to find location* (-)
restaurant: $a_{10,2}$: *the restaurant is uncomfortably crowded* (-)
staff: $a_{10,3}$: *the staff is hard to reach* (-)

The arguments and their associated topic can be seen in Table 5.11. We use RbAM to determine relations between sentences/arguments associated to topics identified in the previous step. The arguments in the BAF include a single special argument G (for 'good') as well as a special argument G_t per topic t such that each G_t supports G. We use a temporal approach for determining the relations between arguments related to topic t drawn from reviews and the special argument G_t.

Figure 5.11 shows the BAF extracted from the reviews in Table 5.10. The extracted arguments occurring as nodes in the BAF are shown in Table 5.11. Note that not all texts in the reviews contribute to the arguments in the BAF. For example, the first sentence from $\mathbf{R_1}$: *This hotel is absolutely beautiful* does not represent an argument as it mentions 'hotel' and not a specific topic related to hotel and we are interested in the *reasons* why this hotel was good.

As an illustration, consider the first three arguments about the topic *room*: $a_{1,1}$, $a_{2,2}$, $a_{3,2}$. As $a_{1,1}$ is the first argument that mentions the topic, it is connected to G_{room} according to its polarity (i.e. $a_{1,1}$ *supports* G_{room}). In this case step the LSTM-based classifier deemed that $a_{2,2}$ neither attacks nor supports $a_{1,1}$ (we only consider relations from $a_{2,2}$ to $a_{1,1}$ as per our temporal approach). Then $a_{2,2}$ *supports* G_{room} as it has a positive polarity and there is no other less recent argument to be compared with. Here the classifier also identifies a support relation between $a_{3,2}$ and $a_{2,2}$. Thus, this relation is included in the BAF.

Now consider argument $a_{3,2}$. While this can be deemed to support both $a_{2,2}$ and G_{room}, it only supports the latter in the BAF in Figure 5.11. Indeed, in our temporal approach, the system firsts checks $a_{3,2}$ against $a_{2,2}$. If a relation is found between these arguments, then the system does not check for the relation between $a_{3,2}$ and G_{room} as we want a "minimal" BAF, in terms of the number of relations it accommodates. Similarly, for $a_{4,1}$, the system checks for a relation between this argument and the most recent one, in this case $a_{3,2}$. If the system had not

Topic	Argument id	Argument
staff	$a_{2,1}$	very pleasant front staff
	$a_{3,1}$	staff were helpful & friendly
	$a_{5,2}$	the staff is very caring
	$a_{6,1}$	the staff was unaccommodating, the front desk staff was condescending and not even remotely helpful
	$a_{7,1}$	the staff are polite and well poised
	$a_{8,4}$	the staff was helpful and attentive
	$a_{9,1}$	there was only one person from the staff at the front desk when we arrived, preoccupied with something on their computer so our presence was not acknowledged for several minutes
	$a_{10,3}$	the staff is hard to reach
room	$a_{1,1}$	room was gorgeous
	$a_{2,2}$	large rooms
	$a_{3,2}$	room was huge with fantastic view
	$a_{4,1}$	the rooms are so much smaller than it seems on the website
	$a_{5,1}$	the rooms were clean
	$a_{6,2}$	the room was not clean
	$a_{7,2}$	the rooms were exceptionally clean and tidy
	$a_{8,2}$	the room was charming with a window seat and a water view
	$a_{9,2}$	our room, I found it to be incredibly dusty.
internet	$a_{2,3}$	free Internet for those that are members of the loyalty program
	$a_{4,2}$	the wireless internet is extremely slow
	$a_{8,3}$	free wireless internet were a plus here
bar	$a_{1,2}$	the bar is very boring
	$a_{3,3}$	adjoining Bar, great food
restaurant	$a_{1,3}$	the restaurant is not that great
	$a_{3,4}$	adjoining Restaurant, great food
	$a_{7,3}$	at their restaurant where I had one of the best American style meal in a while
	$a_{10,2}$	the restaurant is uncomfortably crowded
location	$a_{8,1}$	I loved the location
	$a_{10,1}$	located in a hard to find location

Table 5.11.: Arguments extracted from the reviews in Table 5.10.

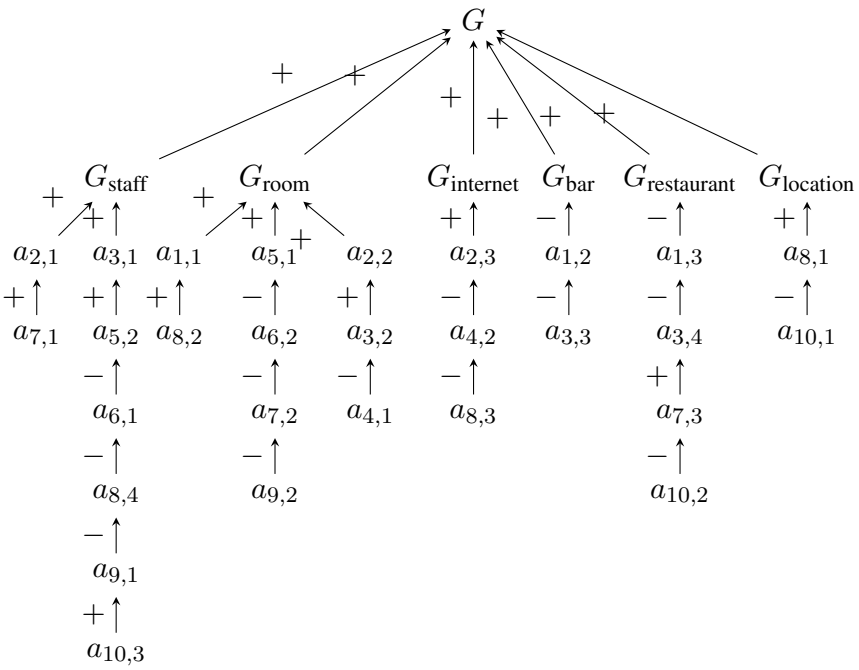


Figure 5.11.: BAF obtained from the reviews in Table 5.10.

identified any relation between these two arguments, then $a_{4,1}$ would have been checked against $a_{2,2}$, the next "related" argument.

The extracted BAFs also provide the underlying structure for generating explanations. A user may request explanations of an argument related to the 'goodness' of the hotel (i.e. G or G_t).

Definition 5.3.2. Given a mined BAF $\langle Args, \mathcal{L}^-, \mathcal{L}^+ \rangle$ and a corresponding QBAF $\langle Args, \mathcal{L}^-, \mathcal{L}^+, \tau \rangle$ with strength σ where $\tau(a) = 0.5$ for any $a \in Args$, an *argumentation dialogue* between a user and the system consists of *explanation requests* $\mathcal{Q}(\alpha)$ for $\alpha \in \{G, G_t\}$ from the user, to which the system responds with *explanations* $\mathcal{X}(\alpha)$.

Definition 5.3.3. A *simple argumentation dialogue* is such that for any $\alpha \in \{G, G_t\}$ where t represents the set of topics identified in reviews:

if $\alpha = G$:

$$\mathcal{Q}(\alpha) = \{\text{What are the poorly rated aspects?}\}$$
$$\mathcal{X}(\alpha) = \{\text{The poorly rated aspects are}\}+$$
$$\{t|\sigma(G_t) \leq 0.5\}; \text{ else}$$

if $\alpha = G$:

$$\mathcal{Q}(\alpha) = \{\text{What are the highly rated aspects?}\}$$
$$\mathcal{X}(\alpha) = \{\text{The highly rated aspects are}\}+$$
$$\{t|\sigma(G_t) > 0.5\}; \text{ else}$$

if $\alpha = G_t$:

$$\mathcal{Q}(\alpha) = \{\text{How is } t \text{ evaluated?}\}$$
$$\mathcal{X}(\alpha) = \{t \text{ is evaluated at } \sigma(G_t)\}$$

For illustration, using the DF-QuAD method that quantifies the strength of arguments (in a $[0, 1]$ interval) by aggregating the strength of their attackers and supporters, in the case of the reviews example in Table 5.10, the strengths for our G_t arguments are:
$\sigma(G_{\text{staff}}) = 0.955$
$\sigma(G_{\text{room}}) = 0.967$
$\sigma(G_{\text{internet}}) = 0.687$
$\sigma(G_{\text{bar}}) = 0.375$
$\sigma(G_{\text{restaurant}}) = 0.406$
$\sigma(G_{\text{location}}) = 0.625$

An argumentation dialogue between a user and the system may be as follows:

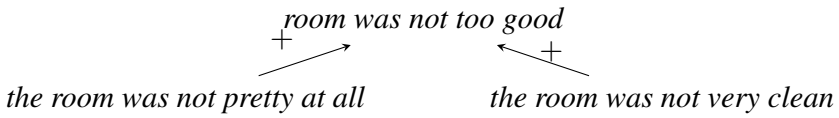
user: *What are the poorly rated aspects?*
system: *The poorly rated aspects are bar and restaurant.*
user: *What are the highly rated aspects?*
system: *The highly rated aspects are staff, room, internet, location.*
user: *How is $room$ evaluated?*
system: *Room is evaluated at 0.967.*
user: *How is $location$ evaluated?*
system: *Location is evaluated at 0.625.*

These explanations provide a measure of how good the hotel is, along the various dimensions (topics) considered, according to the available reviews, and can also be used to compare the hotel with others.

Our computed strengths for G_t arguments show that the strongest points of the hotel in question are the staff and the rooms (with G_{staff} and G_{room} both above 0.95), whereas the bar and restaurant are the low points (G_{bar} and $G_{\text{restaurant}}$ below or close to 0.4). We can also deduce that the internet and the location are rather average (with strengths between 0.6 and 0.7). This evaluation allows users to determine whether the hotel is suitable for them as users are interested in different aspects when they choose a product or service.

Further, BAFs could be employed in tasks such as summarisation as they provide a structured and concise view of the aspects (topics) mentioned in text. In our example hotel, from the BAF in Figure 5.11, we could, for example, hypothesise that *internet* used to be good, since the first review mentioning *internet* was positive ($a_{2,3}$), but has since been unstable, as the next review mentioning it is negative ($a_{4,2}$) and is followed by a positive review (see $a_{8,3}$, which attacks $a_{4,2}$).

BAFs can also help in identifying arguments that are widely accepted as well as identifying conflicting viewpoints that arise in debates. Consider the following simple example:



Here the reviews lead to the conclusion that the *rooms* were not good in the particular hotel under consideration, and the root argument in the graph is widely accepted.

5.4. Summary and Discussion

In this chapter we have focused on analysing texts from (social) media as the e-commerce and audio/video streaming have become dominant markets for consumers, with products' online reviews being the preferred method of quality control for users. These types of websites that aggregate reviews make it easier for users to check the quality of a product while avoiding reviews which may be incoherent, irrelevant or even malicious. In this setting, we used RbAM and BAFs to

determine how a movie's overall score was produced from analysing reviews from RT as well as identifying the 'goodness' of aspects extracted from hotel reviews. In addition, we also improved ML classifiers in detecting deceptive reviews by creating new features from evaluating the mined BAFs.

Within the movie domain, we have defined a novel form of dialogical agent, ADA, which extracts a form of AFs, QBAFs, from RT, a review aggregation website that suffers from a lack of reasoning and explainability. ADA uses NLP and quantitative argumentation to construct the QBAFs that are mined from reviews. We experimented with three gradual semantics for QBAFs, which gave comparable results to the RT score. Our method empowers existing techniques (Sentiment Analysis, RbAM, gradual evaluation for QBAFs) to be deployed, surpassing the individual components it integrates. It is extensible in the sense that we can add more features), and the extensions do not require the models to be retrained (as the training occurs independently of the features).

There is a correspondence between AM and Sentiment Analysis, and in particular of aspect-based Sentiment Analysis which is similar in spirit to our methodologies for our topic/feature characterisations of texts when mining BAFs. Some of the proposed approaches used for identifying aspects (and polarity towards these aspects) in texts make use of common nouns, predefined sets of aspects, rule-based algorithms, dependency parsers, topic modelling, Conditional Random Fields [163] or DL techniques such as Recurrent Neural Networks and Convolutional Neural Networks [53]. Our focus was to identify aspects from reviews and we have achieved this using a variety of methods: meta-data of movie domain, semantic networks to identify related terms that can be considered to belong to the same theme, frequently occurring nouns, and topic modelling.

For future work, the framework could be developed by representing the phrases as arguments (see [146]) which, combined with more sophisticated mining techniques, would allow the recognition of similar phrases that can be represented as a single argument as well as identifying whether there is an argumentative relation between these phrases, rather than having the parent argument as a natural language template. This method could be tested on larger datasets, e.g. the "all critics" section of RT or the full reviews (as opposed to the snippets). The advantage of ADA lies in its explanations as QBAFs provide the underlying structure for generating dialogical explanations for users. As we have used BAFs as the underlying structure for explanations, we plan to run user studies to decide on the best way to provide the explanations, e.g. graphical, visual or linguistic. Deploying

ADA on different aggregation websites is also left for future work.

Within the hotel domain, we proposed a methodology for mining BAFs using RbAM. In particular, our methodology uses RbAM to construct BAFs by determining relations between texts that refer to the same topic, along a temporal dimension whereby more recent texts may either support or attack less recent ones, but not vice versa. We have illustrated our methodology on hotel reviews and provided explanations about the 'goodness' of the hotel. Further work is needed to explore the applicability of our approach in other settings supported by user experimentation to determine the usefulness of BAFs seen as summaries and of explanations drawn from them. We also plan to test whether the temporal dimension is useful in other settings, different from online reviews.

We have focused on extracting BAFs from text. Other works extract different types of argument graphs, for other application areas (e.g parliamentary debates [172]). We plan to test and/or adapt our approach for this and other settings. Future work also includes experimenting whether first determining arguments based on their argumentative structure, e.g. as in [85, 120], may be useful to single out chunks of text to be fed into RbAM.

We proposed a hybrid system that combines DL for RbAM and argumentative reasoning to evaluate deception of online opinions and reviews by introducing argumentative features that capture semantic information from reviews represented as BAFs. We showed experimentally, using Random Forests, for reviews about hotels and restaurants, that including the argumentative features yields better results in classifier performance.

Much of the previous work on detecting deceptive reviews focus on detecting either reviews [72, 140, 175] or deceptive spammers [116, 136]. Other works focus on detecting single review spammers [116] and group review spammers [136], while others look at identifying reviews written by the same person but under different names [173]. Different forms of ML have been used in the literature to detect deceptive behaviour, notably unsupervised [137], semi-supervised [72] and supervised [36, 68, 115, 139, 140, 175] techniques, in particular Naïve Bayes, SVMs, and Random Forests. DL techniques have also been used in detecting deceptive reviews, notably Convolutional Neural Networks [17, 204] and LSTMs [17]. Given the small size of the datasets and the type of the datasets (i.e. textual only), we have used Random Forests, focusing on detecting deceptive reviews.

Our experiments indicate that there is promise in integrating DL for determining argumentative relations with argumentative reasoning. These findings are also

backed by the results, varying from 1% to 3%, for determining the truthfulness of a review using a subset of the dataset. Further experimentation is needed to investigate whether the use of argumentative features extracted from BAFs obtained using a DL model can bring further performance improvements for detecting deceptive reviews. We plan to test whether DL and argumentative reasoning perform better than standard supervised ML techniques in other settings, besides detecting deceptive reviews. The exploration of other notions of strength is also left for future work as well as the proposal of a method that computes, rather than uses given, base scores for arguments.

The works most similar in spirit to ours are the ones by Ferilli et al. [66], Poddar et al. [153], and Pazienza et al. [147]. In [66], a summarisation framework based on Abstract Argumentation was proposed. This summarisation can be seen as extracting the most relevant information (arguments) from reviews whereas the focus of [153] is on identifying supporting opinions in reviews. We mine Argumentation Frameworks that can be seen as summaries of reviews and we focus on support and attack relations. Bipolar Weighted Argumentation Frameworks were introduced in [147] where attack and support relations are assigned weights, negative for attacks and positive for supports. In contrast to this work where the graph is evaluated using acceptability semantics, we assign strengths to arguments and use quantitative gradual semantics to evaluate the graph.

6. Mining Abstract Argumentation Frameworks by AA-CBR

In this chapter we deploy AA-CBR to mine AAFs in support of two binary classification tasks: firstly with a dataset with categorical features and secondly with a textual dataset. We give the motivation for this work in Section 6.1. We present our pipeline for extracting AAFs from data using AA-CBR in Section 6.2. We run a preliminary experiment with a dataset of 8K instances of poisonous and edible mushrooms where features are categorical in Section 6.3. We then customise our method on text in Section 6.4. In particular, we upgrade the attack definition in AA-CBR (Section 6.4.2) to experiment with two textual datasets of 1K short sentences each that address the sentiment polarity classification (positive or negative). We then port explanations for AA-CBR to explain predictions for classification tasks on text. In particular, we illustrate two types of explanations (counterfactual and dialectical) that can be drawn from AAFs mined from the textual datasets in Section 6.4.4. We summarise the research in this chapter in Section 6.5.

6.1. Motivation

Data-centric AI has recently received a great deal of attention and has shown success in several domains (e.g. see [110] for DL). ML techniques are able to achieve very good results at the task they are trained on but their decisions are hard to explain (see [5] for Artificial Neural Networks (ANNs)). Several solutions have been proposed to explain ML classifications, for example model-agnostic methods [166, 167]. In NLP tasks, these explanations amount to highlighting the words that contribute the most to the classification task [6, 106].

To address the lack of transparency of classification algorithms, we propose to use AA-CBR [48, 49] in binary classification problems. Indeed, AA-CBR, originally developed for reasoning with cases, has been advocated as having explanatory power. In order to use AA-CBR for explaining classifications, we first need to

adapt it to perform classifications. We treat labelled examples in datasets as cases and unseen examples as new cases. Given how AA-CBR is defined, we can perform binary classifications with binary features only (i.e. a training instance either has the feature or does not). Our proposed method mines AAFs from data and uses them in making predictions and justifying these predictions.

We experiment with two types of datasets, one categorical describing poisonous and edible mushrooms, and one textual concerned with sentiment polarity. Both are binary classification problems, as required for the application of AA-CBR.

We select the dataset of poisonous/edible mushrooms as it was the largest available categorical dataset for binary classification. The choice of categorical features is driven by the fact that each feature can be expanded to become binary as required by AA-CBR. We can then convert categorical features to a binary representation by assigning each feature to its alternative values. For example, for a feature describing colour with three possible values: red, yellow, and blue, we would convert it to three binary features describing each of the attributes. Thus, instead of a single feature describing the colour red, we now have a vector of three values (representing each colour) with 1 for red and 0 for the other two values.

Then we show how this method can be ported, by upgrading AA-CBR, to text. The second experiment is using AA-CBR in an NLP task, namely Sentiment Analysis, for explanations. Sentiment Analysis aims to analyse people's opinions or sentiments towards topics, items, etc. and their attributes [121, 122, 203] and is a recurrent theme in this thesis from the point of view of applications (see Sections 5.2 and 5.3) but also for the similarities it shares with AM. We show that our AA-CBR method outperforms Decision Trees from which explanations of classifications can also be extracted (by means of rules). In addition, we provide examples of explanations for classifications drawn from the mined AAFs.

6.2. Pipeline for predicting with AA-CBR

In this section we describe our methodology for prediction by means of AA-CBR, summarised in Figure 6.1.

Consider a *training dataset* consisting of a (possibly large) set of examples, (referred to as $\mathcal{L}$ in the remainder), each characterised by a set of *features* from a given set $\mathbb{F}_{\mathcal{L}}$ and an *outcome* from a set $\mathbb{O}$. There are exactly two outcomes in $\mathbb{O}$, namely prediction is a binary classification problem. The set $\mathbb{F}_{\mathcal{L}}$ may be large, especially if features are assignments of alternative values to attributes (for

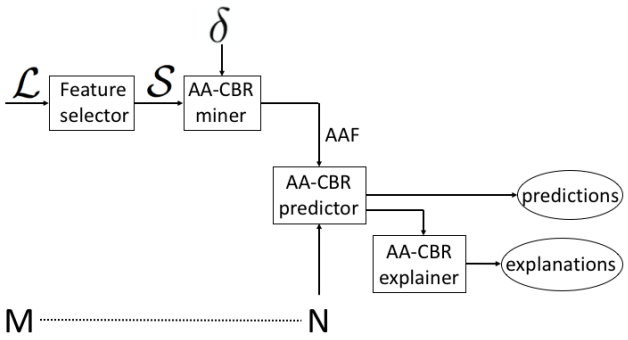


Figure 6.1.: An overview of how AA-CBR is used for classification. The AAF is used to predict the oucome of (any) M via its trimmed version N.

example the attribute 'colour' may take many different values). Consider also an example M whose outcome we want to predict.

We then carry out four stages, described below, alongside the other components of Figure 6.1 ($\mathcal{S}$, δ, and N).

6.2.1. Stage 1: Feature selector

Given that $\mathbb{F}_{\mathcal{L}}$ may be very large and in addition AA-CBR requires features to be binary and the mapping of categorical features to a binary representation can lead to a very large number of features, we need a way to select the salient features in classification before we can apply variants of AA-CBR.

Thus, the first stage in the pipeline is concerned with dimensionality reduction. Typically, dimensionality reduction refers to selecting a subset of the original feature set (selection) or mapping the initial data in V-dimensional space onto a U-dimensional space with $U < V$ (extraction). We focus on feature selection as we are interested in keeping (subsets) of the original features. We will see in Sections 6.3 and 6.4 different feature selectors depending on the size of the datasets and the application.

The ranking is then used to select a subset of (highest-ranked) features. Thus, applying the feature selector results in a *trimmed dataset* (referred to as $\mathcal{S}$ in the remainder), which may be much smaller than the original $\mathcal{L}$ and consisting of examples with much fewer features (from a *trimmed set of features* $\mathbb{F} \subseteq \mathbb{F}_{\mathcal{L}}$). Let the *trimming function* $\mathcal{T} : \wp(\mathbb{F}_{\mathcal{L}}) \mapsto \wp(\mathbb{F})$ be $\mathcal{T}(X) = X \cap \mathbb{F}$, for any $X \subseteq \mathbb{F}_{\mathcal{L}}$. Formally, $\mathcal{S} = \{(Y, o) | (X, o) \in \mathcal{L}, Y = \mathcal{T}(X)\}$. We enforce that the feature selection, leading to $\mathbb{F}$, is such that $\mathcal{S}$ is *coherent* (namely, with no two trimmed

examples in $\mathcal{S}$ having the same features but different outcomes) and thus faithful of the original $\mathcal{L}$. Thus the feature selector guarantees that $\mathcal{S}$ can be used to form an AAF by applying AA-CBR in later stages. Here, we show what this first stage amounts to by using the following example.

Example 6.2.1. Let examples in $\mathcal{L}$ be characterised by 5 attributes $a_1, \ldots, a_5$ and an outcome amongst o_1, o_2. Suppose each attribute may take one of 4 distinct, discrete values, say $v^1_{a_i}, \ldots, v^4_{a_i}$ for a_i. Then $\mathbb{F}_{\mathcal{L}} = \{a_1{=}v^1_{a_1}, \ldots, a_5{=}v^4_{a_5}\}$ consists of 20 features (binary attribute-value pairs). Suppose that we select 4 features, namely $\mathbb{F}$ consists of 4 elements. Concretely, say

$$\mathbb{F} = \{a_1 = v^1_{a_1}, a_2 = v^4_{a_2}, a_3 = v^2_{a_3}, a_4 = v^4_{a_4}\} = \{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}$$

If $\mathcal{L} = \{(\{\mathsf{a}, a_1 = v^2_{a_1}, a_5 = v^1_{a_5}\}, o_1), (\{\mathsf{a}, a_1 = v^3_{a_1}\}, o_1)\}$ then $\mathcal{S} = \{(\{\mathsf{a}\}, o_1)\}$. Note that if $\mathcal{L}$ had also included $(\{\mathsf{a}, a_5 = v^1_{a_5}\}, o_2)$, then we would not have selected $\mathbb{F}$, as this would have resulted in an incoherent $\mathcal{S} = \{(\{\mathsf{a}\}, o_1), (\{\mathsf{a}\}, o_2)\}$.

In the remainder we will refer to elements of $\mathcal{L}$ as *original examples*, and to elements of $\mathcal{S}$ simply as *examples*.

6.2.2. Stage 2: AA-CBR miner

The next stage in the pipeline is the AA-CBR miner which takes the set of examples in $\mathcal{S}$ (that can also be seen as cases) and returns an AAF characterised as $(Args, \mathcal{L}^-)$. We assume that a default δ can be naturally identified in $\mathbb{O}$, as the prediction that can be legitimately made in the absence of information about features. The AA-CBR miner returns the AAF $(\mathcal{S} \cup \{(\{\}, \delta)\}, \mathcal{L}^-)$ (see AA-CBR Step 1 in Section 2.3.1.1). In the AAF, the set of arguments consists of the trimmed examples as well as the *default* argument, and an argument attacks another if they have different outcomes and the former is a 'concise' superset of the latter.

Note that the AAF returned by the AA-CBR miner can be seen as a model of the trimmed dataset, and implicitly of the original training dataset. This model identifies conflicts between (original) examples, that need to be resolved every time that a prediction is to be made.

6.2.3. Stage 3: AA-CBR predictor

Having obtained an AAF, we can now classify new examples consisting of a set M of features in $\mathbb{F}_{\mathcal{L}}$ as shown in Figure 6.1. M characterises an unseen example

with unknown outcome, and predicts an outcome from $\mathbb{O}$ as follows. We use the earlier *trimming function* to obtain $N = \mathcal{T}(M)$. Note that trimming different sets of features may result into the same set. Considering Example 6.2.1, $\mathcal{T}(\{\mathsf{a}, a_1 = v^2_{a_1}, a_5 = v^1_{a_5}\}) = \mathcal{T}(\{\mathsf{a}, a_1 = v^3_{a_1}\}) = \{\mathsf{a}\}$.

The prediction is made by adding to the AAF a new argument for N, attacking every argument in the AAF with features irrelevant to N, and determining whether the default argument is dialectically acceptable in the extended AAF, using the grounded semantics [55]. Thus, the AA-CBR predictor takes $N = \mathcal{T}(M)$ in input, alongside $(Args, \mathcal{L}^-)$ given by the AA-CBR miner, and computes the outcome of $((Args, \mathcal{L}^-), N)$ (see Step 2 in Section 2.3.1.1).

Example 6.2.2. Assuming the *default* is o_1 and $\mathcal{S} = \{(\{\mathsf{a}\}, o_2), (\{\mathsf{a}, \mathsf{b}\}, o_1)\}$, then $\mathcal{L}^- = \{((\{\mathsf{a}\}, o_2), (\{\}, o_1)), ((\{\mathsf{a}, \mathsf{b}\}, o_1), (\{\mathsf{a}\}, o_2))\}$. Given $(N, ?) = (\{\mathsf{a}, \mathsf{d}\}, ?)$ then the outcome for N is o_2 (i.e. the *non-default*) as $\mathbb{G} = \{(\{\mathsf{a}\}, o_2), (\{\mathsf{a}, \mathsf{d}\}, ?)\}$ and $(\{\}, o_1) \notin \mathbb{G}$.

6.2.4. Stage 4: AA-CBR explainer

Justifications for outcomes are given in terms of dispute trees as in AA-CBR (see Step 3 in Section 2.3.1.1). The explanation for the outcome of N in Example 6.2.2 is derived from the following (maximal) dispute tree:

$$
\begin{array}{c}
[P : (\{\}, o_1)] \\
| \\
[O : (\{\mathsf{a}\}, o_2)] \\
| \\
[P : (\{\mathsf{a}, \mathsf{b}\}, o_1)] \\
| \\
[O : (\{\mathsf{a}, \mathsf{d}\}, ?)]
\end{array}
$$

We will port explanations of AA-CBR to explain predictions for classification tasks on text. In particular, we will provide two types of explanations: counterfactual and dialectical.

Definition 6.2.1. Let $\langle Args, \mathcal{L}^- \rangle$ be the mined AAF. A *counterfactual explanation* for a new case N with outcome $\bar{o}$, given $((N, ?), (f, o)) \in \mathcal{L}^-,\ o, \bar{o} \in \{\delta, \bar{\delta}\}$, is as follows:

- if N contained $f \setminus N$ instead of $N \setminus f$, then the outcome would have been o.

Definition 6.2.2. Let $\langle Args, \mathcal{L}^{-} \rangle$ be the mined AAF and $f_1 \ldots f_n$ sets of features. A *dialectical explanation from a maximal dispute tree* is obtained from a branch as follows:

$$
\begin{array}{c}
[P : (\{\}, \delta)] \\
| \\
\ldots \\
| \\
[P : (f_{n-1}, \delta)] \\
| \\
[O : (N, ?)]
\end{array}
$$

P: 'δ';
$\ldots$
P: (f_{n-1}, δ) attacks the above argument;
O: $f_{n-1} \setminus N$ is/are irrelevant (as absent from N)

Note that if $n = 1$ (the proponent is always the first one in the dialogue) then there is a single argument from P, namely the outcome δ. Otherwise, O's first argument (that attacks P's argument represented by the default) is as follows: "$(f_1, \overline{\delta})$ attacks the outcome δ".

Definition 6.2.3. Let $\langle Args, \mathcal{L}^{-} \rangle$ be the mined AAF and $f_1 \ldots f_n$ sets of features. A *dialectical explanation from an admissible dispute tree* is obtained from a branch as follows:

$$
\begin{array}{c}
[P : (\{\}, \delta)] \\
| \\
[O : (f_1, \overline{\delta})] \\
| \\
\ldots \\
| \\
[O : (f_{n-1}, \overline{\delta})] \\
| \\
[P : (N, ?)]
\end{array}
$$

P: 'δ';
O: $(f_1, \overline{\delta})$ attacks the outcome 'δ';
...
O: $(f_{n-1}, \overline{\delta})$ attacks the above argument;
P: $f_{n-1} \setminus N$ is/are irrelevant (as absent from N)

Note that if $n = 2$ (the opponent is always the second one in the dialogue) then there is a single argument from O, namely $(f_1, \overline{\delta})$.

In Example 6.2.2, the counterfactual explanation may be as follows:

- If N contained b instead of d, then the outcome would have been o_1.

If $(N', ?) = (\{\mathsf{a}, \mathsf{b}\}, ?)$ (thus N' is N with b instead of d) then $(\{\}, o_1) \in \mathbb{G}$ where $\mathbb{G} = \{(\{\}, o_1), (\{\mathsf{a}, \mathsf{b}\}, o_1), (\{\mathsf{a}, \mathsf{b}\}, ?)\}$ and the outcome for N' is o_1 as it would be matching exactly an encountered example.

The dialectical explanation that shows the debate between P and O can unfold with the following arguments:

P: 'o_1';
O: $(\{\mathsf{a}\}, o_2)$ attacks the outcome 'o_1';
P: $(\{\mathsf{a}, \mathsf{b}\}, o_1)$ attacks the above argument;
O: b is irrelevant (as absent from N).

P starts by arguing for outcome o_1 saying that, in the absence of any information, the outcome should be o_1. O then argues against it by putting forward an argument with outcome o_2. P then puts forward a more specific example that gives outcome o_1 with some features that are in N (i.e. a). However, the new case attacks this argument, thus this argument is discarded from having an influence on the outcome. Hence, the outcome is deemed to be o_2.

6.3. Predicting and explaining from categorical data with AA-CBR

The first experiment uses the pipeline in Section 6.2 to provide explanations for a categorical dataset, as a preliminary step towards textual datasets.

We choose the mushroom dataset[1] from the UCI Machine Learning Repository [52] as data to be analysed. The dataset contains 8124 examples of gilled mushrooms classified as edible or poisonous. Each example is characterised by 22

[1] archive.ics.uci.edu/ml/datasets/Mushroom

categorical attributes that can take a number of different values, leading to 126 binary features. Thus, our starting point consists of 8124 examples with 126 binary features.

6.3.1. Stage 1: Feature selection

Due to the size of the dataset being quite large, we use an autoencoder, a type of ANN, as our feature selector. ANNs have been widely applied both in classification tasks and in dimensionality reduction, e.g. as in [189]. ANN-based feature selection methods use multilayer perceptrons to determine which features are redundant [75] as well as autoencoders [88, 90, 197]. These are unsupervised learning models based on ANNs which take a set of features as input and aim, through training, to reconstruct the inputs [90, 61].

Our proposed autoencoder is shown in Figure 6.2.

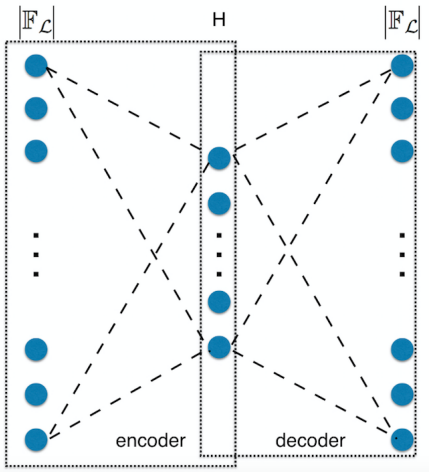


Figure 6.2.: Autoencoder architecture: $|\mathbb{F}_{\mathcal{L}}|$ binary features are used to train the autoencoder to obtain a **code** (hidden layer **h** of size $H < |\mathbb{F}_{\mathcal{L}}|$) that best captures the input $|\mathbb{F}_{\mathcal{L}}|$ features.

The autoencoder has one hidden layer with (for $X \subseteq \mathbb{F}_{\mathcal{L}}$):

1. an encoder function $f(X) = \sigma(XW^{(1)})$
2. a decoder function $\sigma(f(X)W^{(2)})$

where $W^{(1)}, W^{(2)}$ are the weight parameters in the encoder and decoder, respectively.

In order to select $\mathbb{F}$, we average the weights in $W^{(1)}$ for each input and select the top F factors. F can be chosen in many alternative ways, either iteratively (starting

from a small number of features, until a coherent S is obtained) or empirically (as we will see in the experiments in Section 6.3.2).

6.3.2. Stages 2 and 3: Empirical evaluation

We then deploy AA-CBR miner and predictor as given in Section 6.2. In order to verify that AA-CBR is useful for explanations, we first need to verify that it predicts well.

We conduct experiments with the mushroom dataset showing that our method significantly outperforms ANNs, while being less sensitive to the size of the training dataset, as well as Decision Trees. In our experiments we choose several subsets of this dataset as $\mathcal{L}$, but, for all choices, $|\mathbb{F}_{\mathcal{L}}| = 126$. We have $|\mathbb{F}| = 22$, for $\mathbb{F}$ the trimmed set of features, $\mathbb{O} = \{\text{edible, poisonous}\}$ and $\delta = \text{edible}$. This means that in the absence of any information (i.e. features) about a mushroom, it can be deemed edible, as represented by $(\{\}, \delta)$. However, as soon as a mushroom with features is encountered, it being edible has to be justified by countering all the relevant examples of poisonous mushrooms.

We use sigmoid as activation function in the autoencoder and binary cross entropy as loss function. We experiment with different sizes for the hidden layer in the autoencoder, $H \in \{22, 30\}$ (see Figure 6.2).

We have experimented with $|\mathbb{F}| \in \{10, 22, 30, 50\}$ as these were less than the original number of features, i.e. 126, with tanh and ReLU as activations functions and with various optimizers, but report results for the best performing combinations of parameters obtained using grid search.

In Table 6.1 we report 5-fold cross-validation results, using weighted averages for each metric, for a stand-alone ANN, for a combination of Autoencoder+ANN, and for our AA-CBR based method.

Hidden layer size 22	**Precision**	**Recall**	$\mathbf{F_1}$
AA-CBR based method	**0.97**	**0.96**	**0.958**
Autoencoder + ANN	0.938	0.894	0.878
ANN	0.934	0.888	0.87
Hidden layer size 30			
AA-CBR based method	**0.97**	**0.962**	**0.962**
Autoencoder + ANN	0.932	0.886	0.86
ANN	0.936	0.896	0.88

Table 6.1.: 5-fold cross-validation results

The chosen ANN has one hidden layer and uses sigmoid as activation function and softmax to make predictions. The hyper-parameters were optimised using the Adam method [100] with learning rate 0.001. For Autoencoder+ANN, we use the learnt weights $W^{(1)}$ from the encoder, which we do not optimise during training, and softmax for classification. In both cases, we trained for 50 epochs or until the performance on the development set stopped improving. In the case of AA-CBR based method, we use the learnt weights from the encoder to select the top 22 features. As shown in Table 6.1, our method performs better than the two ANN approaches with differences in F_1 up to 8% when using a hidden layer size of 22, and up to 10% when using a size of 30 for the hidden layer.

We also conducted experiments to test whether our method can better cope with smaller datasets than the Autoencoder+ANN method (arguably the better performing of the two end-to-end ANN methods). Hence we run experiments on 6000 randomly drawn examples and 5000 randomly drawn examples, respectively, and tested on the remaining examples in the starting dataset. We repeated the experiments 5 times and report the average performances in Table 6.2. Here as well we use the learnt weights $W^{(1)}$ from the encoder in Autoencoder+ANN and softmax for prediction.

In this set of experiments, we also used for comparison Decision Trees that also use an autoencoder for feature selection. For our method and Decision Trees alike, we use the learnt weights from the encoder to select the top 22 features as $\mathbb{F}$ and give $\mathcal{S}$ as discussed in Section 6.2. We used information gain for Decision Trees.

The experiments on reduced datasets show that Autoencoder+ANN is less performing than our method and Decision Trees. Our approach performs better than Decision Trees throughout all experiments, with improvements in F_1 up to 20% with training set size of 6000 examples, and up to 12% with training set size of 5000 examples.

Training set size: 6000, Testing set size: 2124			
Hidden layer size 22	**Precision**	**Recall**	$\mathbf{F_1}$
AA-CBR based method	**0.978**	**0.976**	**0.976**
Autoencoder + ANN	0.802	0.642	0.61
Decision Tree	0.858	0.774	0.762
Hidden layer size 30			
AA-CBR based method	**0.966**	**0.964**	**0.966**
Autoencoder + ANN	0.802	0.638	0.604
Decision Tree	0.852	0.772	0.766
Training set size: 5000, Testing set size: 3124			
Hidden layer size 22	**Precision**	**Recall**	$\mathbf{F_1}$
AA-CBR based method	**0.954**	**0.954**	**0.954**
Autoencoder + ANN	0.84	0.76	0.75
Decision Tree	0.876	0.828	0.826
Hidden layer size 30			
AA-CBR based method	**0.97**	**0.97**	**0.97**
Autoencoder + ANN	0.84	0.756	0.748
Decision Tree	0.886	0.844	0.844

Table 6.2.: Average of 5 runs of training on a reduced dataset and testing on the remaining examples.

6.3.3. Stage 4: Illustration of AA-CBR explanations

Consider the following instances of mushrooms (with outcome *edible* represented as $+$ and *poisonous* as $-$) in $\mathcal{S}$ after the feature selector ($\mathbb{F} = \{$gill-attachment_free (GAF), ring-number_one (RNO), veil-color_white (VCW), odor_none (ON), habitat_woods (HW)$\}$:

- $(\{\text{GAF}\}, -)$
- $(\{\text{GAF, RNO}\}, +)$
- $(\{\text{GAF, VCW}\}, +)$
- $(\{\text{GAF, RNO, VCW}\}, -)$
- $(\{\text{GAF, RNO, VCW, ON}\}, +)$

and a new instance, whose type, edible or poisonous, we aim to determine and explain $(N, ?) = (\{\text{GAF, RNO, VCW, HW}\}, ?)$ given the AAF in Figure 6.3.

With the default $+$, as $(\{\}, +) \notin \mathbb{G}$, for $\mathbb{G}$ the grounded extension of this AAF ($\mathbb{G} = \{(\{\text{GAF}\}, -), (\{\text{GAF, RNO, VCW}\}, -), (\{\text{GAF, RNO, VCW, HW}\}, ?)\}$),

the outcome of $(N, ?)$ is $-$.

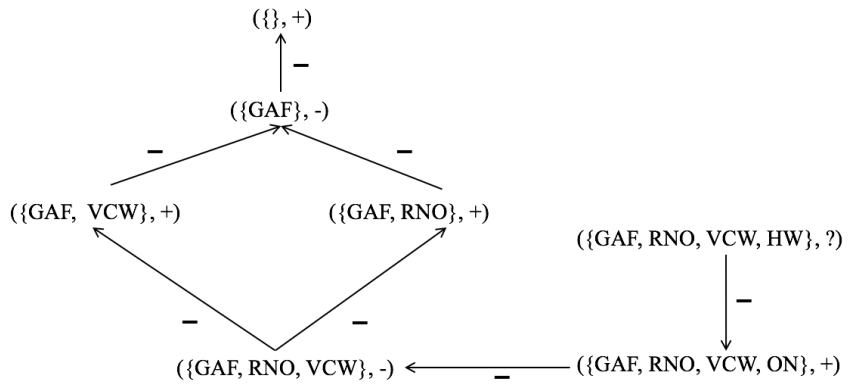


Figure 6.3.: Example of AAF for categorical data.

We obtain two maximal dispute trees that justify the outcome:

$$
\begin{array}{c}
[P : (\{\}, +)] \\
| \\
[O : (\{\text{GAF}\}, -)] \\
| \\
[P : (\{\text{GAF, RNO}\}, +)] \\
| \\
[O : (\{\text{GAF, RNO, VCW}\}, -)] \\
| \\
[P : (\{\text{GAF, RNO, VCW, ON}\}, +)] \\
| \\
[O : (\{\text{GAF, RNO, VCW, HW}\}, ?)]
\end{array}
\qquad
\begin{array}{c}
[P : (\{\}, +)] \\
| \\
[O : (\{\text{GAF}\}, -)] \\
| \\
[P : (\{\text{GAF, VCW}\}, +)] \\
| \\
[O : (\{\text{GAF, RNO, VCW}\}, -)] \\
| \\
[P : (\{\text{GAF, RNO, VCW, ON}\}, +)] \\
| \\
[O : (\{\text{GAF, RNO, VCW, HW}\}, ?)]
\end{array}
$$

We can obtain two types of explanation from this AAF: counterfactual and dialectical (as a debate between proponent P and opponent O).

For the instance that we aim to classify, the *counterfactual explanation* may be as follows:

- If N contained *ON* instead of *HW*, then the outcome would have been $+$.

The *dialectical explanation* that shows the debate between P and O can unfold with the following arguments:

P: '+';
O: $(\{GAF\}, -)$ attacks the outcome '+';
P: $(\{GAF, RNO\}, +)$ attacks the above argument;
O: $(\{GAF, RNO, VCW\}, -)$ attacks the above argument;
P: $(\{GAF, RNO, VCW, ON\}, +)$ attacks the above argument;
O: *"ON"* is irrelevant (as absent from N).

Thus, as P cannot argue against O, $+$ cannot be upheld dialectically, thus explaining the prediction $-$. Note that here we focused on the left tree. A different *dialectical explanation* can be obtained from the right tree as follows:

P: '+';
O: $(\{GAF\}, -)$ attacks the outcome '+';
P: $(\{GAF, VCW\}, +)$ attacks the above argument;
O: $(\{GAF, RNO, VCW\}, -)$ attacks the above argument;
P: $(\{GAF, RNO, VCW, ON\}, +)$ attacks the above argument;
O: *"ON"* is irrelevant (as absent from N).

6.4. Predicting and explaining from textual data with AA-CBR

Given the positive results on categorical data, our focus in this section is on binary classification of textual data to determine sentiment polarity. We select the texts[2] used in evaluation in [103] that extracted and manually labeled 1000 sentences from Amazon reviews about cell phones and accessories category used in [127] and 1000 sentences from the IMDb movie review sentiment dataset used in [124]. These datasets consist of short sentences that can be clearly distinguished by humans as being positive or negative.

For each of the two websites, there are 500 positive and 500 negative sentences, selected so that they have a clear positive or negative polarity. Below are examples from the dataset:

Amazon negative: I advise EVERYONE DO NOT BE FOOLED!
Amazon positive: So Far So Good!
Amazon negative: The commercials are the most misleading.
Amazon positive: I have yet to run this new battery below two bars and that's three days without charging.

[2] archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences

IMDb negative: It had some average acting from the main person, and it was a low budget as you clearly can see.
IMDb positive: It's practically perfect in all of them a true masterpiece in a sea of faux "masterpieces".
IMDb negative: All in all, a great disappointment.
IMDb positive: This is definitely a cult classic well worth viewing and sharing with others.

As we now operate with text, instead of having categorical features where each attribute can have several values, we represent $\mathbb{F}_{\mathcal{L}}$, i.e. the vocabulary obtained from the lemmas in the dataset $\mathcal{L}$, by means of a one-hot-encoding, where 1 indicates the presence of words in the text, and 0 the absence. In addition, we make two main changes compared with the categorical dataset: we use a different feature selector to accommodate the smaller size of the dataset and we define a new notion of attack for AA-CBR that is applicable to text.

6.4.1. Stage 1: Feature selection

Since the dataset $\mathcal{L}$ is much smaller than the dataset from Section 6.3, the first change we make with respect to the previous experiment is concerned with the choice of the feature selector. An autoencoder (i.e. a neural model) is not fit to be used on a dataset with 1000 instances, thus, to select the most relevant features, we use an ensemble technique, namely Random Forests [27].

In text, different terms can have the same meaning. However, some of these terms may occur more often than others, and hence the less used terms may be overlooked during feature selection. For example, "nice" and "beautiful" can be used interchangeably, but only one of them may appear in the training set. This way, the least used one may be discarded and thus information contributing to classification may be lost. To address this issue, we pre-process the data first to cluster related terms. In our earlier example, "nice" and "beautiful" can be replaced by a single word, e.g. "nice".

6.4.1.1. Clustering related terms

We calculate the term relatedness in order to cluster them efficiently. This semantic similarity leads to a reduced set of features, all related words being replaced by

one of the words from the computed cluster. Thus, while the semantics remains unchanged, the polarity of texts can still be determined, even in the presence of words that have not been encountered before but are related to the ones in the training set.

Computing the semantic relatedness between two words requires that both terms have the same sentiment polarity (e.g. "good" will be tested against "great" which has a positive polarity but not against "bad" which has a negative polarity) and the same POS tag (e.g. "good" will be tested against "amazing" which is an adjective but not against "work" which can be a verb or a noun depending on context).

To compute the semantic relatedness we use the semantic network ConceptNet[3] and select the pairs that have the relatedness score greater than 0.3. Table 6.3 shows examples of pairs of words and their relatedness. Using our imposed threshold, we obtain a single cluster {great, superb, amazing, wonderful, awesome, terrific} and replace each of these words encountered in texts with the word *great*.

Word 1	**Word 2**	**Relatedness score**
great	funny	-0.01
great	fine	0.007
great	interesting	0.115
great	good	0.184
great	important	0.283
great	superb	0.487
great	amazing	0.572
great	wonderful	0.577
great	awesome	0.598
great	terrific	0.653
art	work	0.132
art	fan	0.041
art	treasure	0.084

Table 6.3.: Examples of word relatedness.

Using our example words from Table 6.3, from $|\mathbb{F}_{\mathcal{L}}| = 15$ we get $|\mathbb{F}_{\mathcal{L}}^{C}| = 10$ using the cluster we previously obtained.

[3] conceptnet.io

6.4.1.2. Random Forests for Feature selection

We rely upon an ensemble-based method for classification, namely Random Forests, for capturing the most salient features from the training dataset, resulting in the *trimmed dataset* $\mathcal{S}$. The ensemble method uses TF-IDF features (which are commonly used features for text) of the training dataset and fits 150 random trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and to control overfitting. In the remainer, TF-IDF$_i$ stands for word i in the given set of "documents" (i.e. texts).

As an alternative, we also experiment with the following score for word i which represents a combination of TF-IDF and the polarity of synsets based on the POS tag:

$$\text{TF-IDF-SYN}_i = \text{TF-IDF}_i \cdot \max\left(\frac{\sum_{j \in S_i} \text{pos_score}_j}{|S_i|}, \frac{\sum_{j \in S_i} \text{neg_score}_j}{|S_i|}\right)$$

where $S_i = \text{synsets}(i)$.

In the remainder, we will refer to the weight of word i computed using TF-IDF as $w_{1i} = \text{TF-IDF}_i$ and to the weight of word i computed using TF-IDF and synsets as $w_{2i} = \text{TF-IDF-SYN}_i$. Also, w will stand for any of w_{1i} and w_{2i} when either can be used.

We experiment with different number of features that we select: $\mathbb{F} = \mathbb{F}^C_{\mathcal{L}}$, $|\mathbb{F}| = X$, for $\mathbb{F}$ the trimmed set of features where X is a threshold experimentally set (see Section 6.4.3).

6.4.2. Upgraded AA-CBR: Changing the attack definition

The new definition of attack in AA-CBR is defined in as follows (see Section 2.3.1.1 for the original definition):

- for $(X, o_X), (Y, o_Y) \in \mathcal{S} \cup \{(\{\}, \delta)\}$, it holds that $((X, o_X), (Y, o_Y)) \in \mathcal{L}^-$ iff
  1. $o_X \neq o_Y$, and (different outcomes)
  2. $\sum_{x_i \in X} w(x_i) > \sum_{y_i \in Y} w(y_i) \wedge |X| > |Y|$, and (new specificity $\subset$)
  3. $\nexists (Z, o_X) \in CB$ with $Y \subsetneq Z \subsetneq X$. (concision wrt 2.)

The new specificity replaces $Y \subsetneq X$ in the original definition. We change the definition of attack on the basis that features are equipped with a weight (w) indicating their importance in classification. The concision property is based on the new specificity property.

Intuitively, the new property states that a case with more features attacks another case as it provides more information describing the sentiment polarity. In particular, it forces that a case with more but weaker (in terms of weight) features should attack a case with fewer but stronger (in terms of weight) features. For example, assuming the selected features are adjectives, the (positive) sentence *The movie was amazing, with a very good story and a great cast* attacks the (negative) sentence *Awful film* as it has more information. This way, we can identify attacks between arguments that would otherwise be ignored in the original AA-CBR due to its specificity requirement.

6.4.3. Stages 2 and 3: Empirical evaluation

We show empirically, using the evaluation dataset of [103] that AA-CBR outperforms Decision Trees. We have chosen Decision Trees (DT) as they are also explainable (in the form of rules), unlike Random Forests for example. We will see in Section 6.4.4 the types of explanations we can extract from the AAFs mined.

We experiment with various parameters for AA-CBR and report the best results using 5 fold cross-validation (CV) on the IMDB dataset in Table 6.4 and on the Amazon dataset in Table 6.5. For DT, we used gini impurity criterion and TF-IDF features. For AA-CBR, we experiment with various numbers of selected features as well as other parameters such as the default used (0 for negative polarity and 1 for positive polarity), whether we used synsets when selecting features, clusters to group similar features, and the upgraded version of AA-CBR.

The average 5 fold CV F_1 results with AA-CBR using all features for IMDB is 72% and for Amazon 76.2%. For IMDB, DT and AA-CBR trained on the selected features do not generally perform better, but the best results using AA-CBR are better by 1.8%. For Amazon, DT with selected features does not improve the results when using all features, but AA-CBR performs better than DT with all features, with improvements of up to 3.6%. In all cases when using selected features, AA-CBR performs better than DT with improvements of 3% for IMDB (500 features) and 4.3% for Amazon (500 features).

The best results are obtained when using the upgraded version of AA-CBR, thus considering the weight of feature importance when constructing the framework, no synsets when selecting features, and using clusters. The difference with respect to parameters is given by the default selected: negative in the case of IMDB and positive in the case of Amazon.

Using the upgraded AA-CBR generally performs better than the original version for both datasets. Synsets are an important aspect for Amazon, and a similar pattern can be seen for clusters.

IMDB						
#F	**S**	**D**	**C**	**U**	**DT**	**AA-CBR**
200	Y	1	N	N	69.4	70.7
200	Y	0	N	Y	69.4	71.9
200	N	0	N	Y	70.7	71.9
200	N	0	Y	Y	72.5	73.8
500	Y	0	N	N	67.1	69.4
500	Y	0	N	Y	67.1	70.1
800	Y	0	N	Y	67	69.8

Table 6.4.: Experimental results with various parameters: **#F**(eatures), **S**(ynsets), **D**(efault) chosen, the use of **C**(lusters), **U**(pgraded) version of AA-CBR, obtained using 5 fold CV.

Amazon						
#F	**S**	**D**	**C**	**U**	**DT**	**AA-CBR**
200	N	1	Y	N	75.8	77.1
200	N	0	N	Y	74.1	77
200	N	1	N	Y	74.1	77.6
200	N	1	Y	Y	77	79.8
500	N	1	Y	N	75.3	76.9
500	N	1	Y	Y	75.4	79.7
800	N	1	Y	Y	76	79.7

Table 6.5.: Experimental results with various parameters: **#F**(eatures), **S**(ynsets), **D**(efault) chosen, the use of **C**(lusters), **U**(pgraded) version of AA-CBR, obtained using 5 fold CV.

6.4.4. Stage 4: Illustration of AA-CBR explanations

In this section we describe the types of explanations that can be extracted from the AAFs mined using AA-CBR focusing on the textual data. In addition to predicting the class, counterfactual and dialectical explanations can be obtained from the AAF generated by (all variants of) AA-CBR. We illustrate these two forms of

explanations with examples.

For the first example, let us consider the following sentences with polarity in brackets:

The film was really good. $(+)$
The story was not very good. $(-)$

and a new instance *This movie was very good actually* whose sentiment polarity we aim to determine and explain.

Assuming "really" and "very" are in the same cluster and $\mathbb{F} = \{\text{good}, \text{not}, \text{very/really}\}$ are the most important features, we obtain $CB = \{(\{\text{really, good}\}, +), (\{\text{not, very, good}\}, -)\}$, the new case $(N, ?) = (\{\text{very, good}\}, ?)$, and, with default $-$, the AAF in Figure 6.4. As $(\{\}, -) \notin \mathbb{G}$, for $\mathbb{G}$ the grounded extension of this AAF ($\mathbb{G} = \{(\{\text{really, good}\}, +), (\{\text{very, good}\}, ?)\}$), the outcome of $(N, ?)$ is $+$.

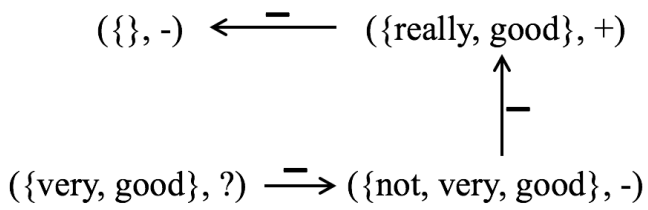


Figure 6.4.: First example of AAF for Sentiment Analysis.

We obtain the following maximal dispute tree that justifies the outcome:

$$[P : (\{\}, -)]$$
$$|$$
$$[O : (\{\text{really, good}\}, +)]$$
$$|$$
$$[P : (\{\text{not, very, good}\}, -)]$$
$$|$$
$$[O : (\{\text{very, good}\}, ?)]$$

We can obtain two types of explanations from this AAF: counterfactual (hinging upon word occurrence) and dialectical (as a debate between proponent P and opponent O).

For the instance that we aim to classify, the *counterfactual explanation* may be as follows:

- If N contained the word *not*, then it would have been negative.

The *dialectical explanation* that shows the debate between P and O can unfold with the following arguments:

P: '$-$';
O: $(\{$(really, good)$\}, +)$, attacks the outcome '$-$';
P: $(\{$(not, very, good)$\}, -)$ attacks the above argument;
O: *"not"* is irrelevant (as absent from N).

As P cannot argue against O, $-$ cannot be upheld dialectically, thus explaining the prediction $+$.

For the second example, let us consider the following sentences with polarity in brackets:

The film was good. $(+)$
The story was not good. $(-)$
I thought it would be good, but in the end it was terrible. $(-)$
The story was not just good, it was wonderful. $(+)$

and a new instance *The movie was not good, it was terrible* whose sentiment polarity we aim to determine and explain.

Assuming $\mathbb{F} = \{\text{good}, \text{not}, \text{wonderful}, \text{terrible}\}$ are the most important features, we obtain $CB = \{(\{\text{good}\}, +), (\{\text{not, good}\}, -), (\{\text{good, terrible}\}, -), (\{\text{not, good, wonderful}\}, +)\}$, the new case $(N, ?) = (\{\text{not, good, terrible}\}, ?)$, and, with default $-$, the AAF in Figure 6.5. As $(\{\}, -) \in \mathbb{G}$, for $\mathbb{G}$ the grounded extension of this AAF ($\mathbb{G} = \{(\{\}, -), (\{\text{good, terrible}\}, -), (\{\text{not, good}\}, -), (\{\text{not, good, terrible}\}, ?)\}$), the outcome of $(N, ?)$ is $-$.

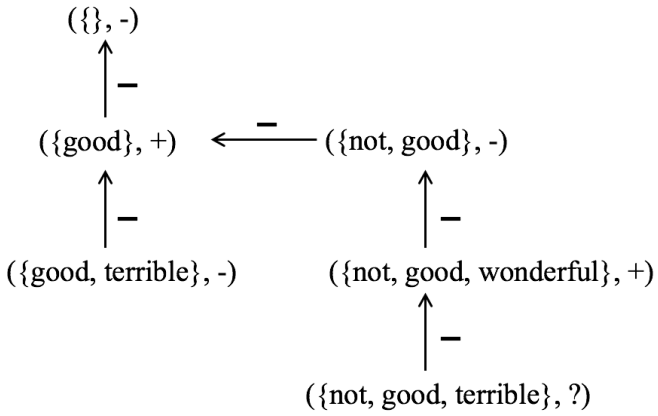


Figure 6.5.: Second example of AAF for Sentiment Analysis.

We obtain the following admissible dispute tree that justifies the outcome:

$$[P : (\{\}, -)]$$
$$|$$
$$[O : (\{\text{good}\}, +)]$$
$$|$$
$$[P : (\{\text{not, good}\}, -)]$$
$$|$$
$$[O : (\{\text{not, good, wonderful}\}, +)]$$
$$|$$
$$[P : (\{\text{not, good, terrible}\}, ?)]$$

For the instance that we aim to classify, the *counterfactual explanation* may be as follows:

- If N contained the word *wonderful* instead of *terrible*, then it would have been positive.

The *dialectical explanation* that shows the debate between P and O can unfold with the following arguments:

P: '$-$';
O: $(\{(\text{good})\}, +$ attacks the outcome '$-$';
P: $(\{(\text{not, good})\}, -)$ attacks the above argument;
O: $(\{(\text{not, good, wonderful})\}, +)$ attacks the above argument;
P: *"wonderful"* is irrelevant (as absent from N).

Thus, as P attacked O's arguments and O has no other arguments to put forward,

it explained the prediction is $-$.

Note that other types of counterfactual explanations can be extracted for dispute trees. In the earlier example, we can obtain another admissible dispute tree:

$$[P : (\{\}, -)]$$
$$|$$
$$[O : (\{\text{good}\}, +)]$$
$$|$$
$$[P : (\{\text{good, terrible}\}, -)]$$

from which a *counterfactual explanation* may be as follows:

- Even if N did not have *not*, the outcome would still be $-$.

6.5. Summary and Discussion

In this chapter we have adapted AA-CBR [48, 49], an existing Abstract Argumentation (AA)-driven Case-Based Reasoning (CBR)-inspired model, for supporting explanations of binary classification problems. With AA-CBR, we mine AAFs from textual data and explain predictions for new instances. We illustrate two types of explanations: counterfactual and dialectical.

We ported the explanations from AA-CBR to explaining predictions for classification tasks. We proposed the use of AA-CBR designed to operate on binary features to predict outcomes from one domain, namely knowledge representation, on categorical data for binary classification as categorical data can be mapped to binary. The portability of (explanations drawn from) AA-CBR was further explored by porting this approach to text.

We experimented with two datasets: a larger dataset of 8K instances consisting of categorical features that determine whether a mushroom is poisonous or edible and two smaller textual datasets of 1K instances each that describe the sentiment polarity. For the categorical dataset, we proposed a method that uses ANNs enhanced with argumentation to (1) select features by means of an autoencoder and (2) to construct an AAF from the training instances that can be used in classification. We showed that our method outperforms ANNs, while being able to provide explanations for classifications.

We then tested the portability of explanations to a different type of dataset, namely textual. For the text-based datasets, we upgraded AA-CBR by proposing

a new definition of attack, and used Random Forests to select features. We have made several changes to accommodate the richness of text. We grouped features based on word relatedness, allowing for words that do no appear in the training set to still contribute to determining the polarity, semantics remaining unchanged. We showed that (upgraded) AA-CBR performs better than Decision Trees, an ML algorithm which is able to explain its classification (by means of rules extracted from the tree) and showed our types of explanations for classification, namely counterfactual and dialectical.

We foresee numerous directions for future work. In terms of methodology and datasets, we plan to experiment with other (larger) datasets (e.g. the full Amazon [127] and IMDB [124]) and other techniques for feature selection or engineering. For the one-hot representation, we plan to test whether using n-grams can improve results (i.e. taking the order of words into account). Further work is required for testing the generalisability of our approach to other NLP domains, not limited to Sentiment Analysis. We also plan to explore the use of embeddings, rather than a one-hot encoding representation of texts. Further modifications need to be applied to AA-CBR to address the continuous space, for example, changing the attack definition as we have done for texts.

Our comparison with Decision Trees was done experimentally: we compared the performance rather than the explanations that can be extracted from either method. Indeed, explanations from Decision Trees can be extracted in the form of rules, whereas our explanations are dialectical and counterfactual, thus their type differs. Both our proposed approach and Decision Trees are interpretable as long as the dialectical tree and decision tree are not deep. In the presence of tens of arguments in our case or features in rules in the case of Decision Trees, the explanations may become complicated and hence affect interpretability. The scalability of our proposed method is left for future work. In the case of text however, clustering similar terms greatly reduces the number of features, thus limiting the depth of the explanations we provide. This is suitable for the Sentiment Analysis task we have considered and may also be suitable for specific RbAM datasets as, for each of these tasks, there is a (limited) number of words that influence the classification (i.e. 'good' and variations for Sentiment Analysis, 'agree' and variations for RbAM).

We have provided and illustrated preliminary definitions of counterfactual and dialectical explanations. We plan to study other types of explanations extracted from dispute trees. We plan to further explore counterfactual explanations which

are a well established type of explanation in social sciences when humans reason about the state of an event that would have resulted from an event that did not occur [134]. As counterfactual explanations are a natural way for humans to reason, the author claims that they should be considered to be incorporated in Explainable AI. Indeed, several works have made use of counterfactual explanations to explain predictions given by Decision Trees [176, 177], predictions given by Neural Networks [192], and explaining why an image does not belong to a counterfactual class [89]. This represents a step forward into providing insight into the algorithmic decision making, thus providing transparency and explainability required by EU General Data Protection Regulation (GDPR).

Finally, we plan to conduct user studies to determine whether the explanations currently extracted can be easily understood by users. For dialectical explanations, we plan to compare their quality to those of other methods, e.g. [166, 167].

7. Conclusion

We have made several contributions to the field of Argument Mining (AM), and in particular to Relation-based Argument Mining (RbAM) [37], including the introduction of Argumentation Frameworks into the realm of explainable AI on textual data. We give a summary of our contributions to this field in Section 7.1 and look forward to future work in Section 7.2 where we weigh on the problems in this field and how our research has provided foundations for these matters.

Our research was driven by the following problems and questions in the field: *1)* can we overcome the current problem of AM solutions that heavily rely on the datasets used in experimentation which makes them difficult to transfer to other domains as well as making them difficult to compare amongst proposed techniques? *2)* can we mine Argumentation Frameworks (AFs), e.g. Abstract Argumentation Frameworks (AAFs) [55] and Bipolar Argumentation Frameworks (BAFs) [40], from text using RbAM or other methods, such as AA-CBR, which is a method that empowers explanations? *3)* can we use these mined AFs efficiently in real world settings from which users would benefit?

7.1. Summary of Contributions

This thesis contributed to the field of AM in several important ways, summarised below. We have made three substantial contributions concerned with portability: *i)* portability of RbAM models, *ii)* portability of RbAM: from downstream task to mining BAFs for applications, *iii)* portability of AA-CBR to address the lack of transparency of classification tasks. We have successfully deployed RbAM in a spectrum of applications: to mine BAFs that we compare with the ones obtained by annotators, on Rotten Tomatoes, a movie review aggregation website that lacks explainability, and on product reviews to detect the deceptive reviews.

Current approaches in RbAM are treated as a stand-alone classification task addressed by Machine Learning (ML) techniques with models trained and tested on data from the same, or similar, domains [9, 33, 178, 180]. Despite the satisfying

results reported in particular domains (education [149, 178], legal [145, 184], politics [128], and web-based content [21, 23, 37, 38]), the models are hardly portable from one application domain to another as they heavily rely on the peculiar features of the datasets used in the experimental setting. Thus, the context-dependency of the learnt models poses an obstacle to the deployment of these models "in the wild".

We identified two realisations of RbAM due to their respective advantages: 2 class RbAM and 3 class RbAM. 2 class RbAM is concerned with determining a class amongst *attack* and *support*. Indeed, the majority of datasets contain only these two types of relations. We proposed several neural models that perform homogeneously over all the datasets proposed in the literature for the relation prediction task in AM. Thus, the contribution of our proposal is to bestow the AM community with a set of strong cross-dataset baselines to compare with in order to demonstrate how well a relation prediction method performs. 3 class RbAM is concerned with determining a class amongst *attack*, *support*, and *neither attack nor support*. We proposed a neural model that determines the argumentative relation, if any, between two texts. The advantage of this realisation is that the model can be used in real world applications where we first need to discard the neither support nor attack relations. Our model is also comparable to human annotators in the task of identifying BAFs from two texts. The advantage of models comparable to annotators is that it can lead to the automatic creation of (AM) corpora, highly needed in experimentation.

In addition to proposing efficient models that address the 3 class RbAM task, we considered the application of RbAM in the online setting in which RbAM can be seen as a prerequisite step for constructing BAFs. We focused on review aggregation sites as they have become a popular method for quality control for users. These sites are simple in the sense that they provide an overall score of reviews in addition to providing the reviews. We proposed the use of BAFs extracted from textual data by means of RbAM with quantitative gradual semantics to provide the backbone for explanations in order to overcome this simplicity. The explanations we proposed are drawn from BAFs mined from text, which can be seen as summaries of reviews. The first setting we considered was the movie domain, and in particular Rotten Tomatoes website. We showed how Natural Language Processing (NLP) and quantitative gradual semantics can be used as a novel approach to remedy the lack of reasoning and explainability of Rotten Tomatoes that aggregates critics' reviews to obtain an overall percentage of how good a movie

is. We illustrated an argumentative dialogical agent (ADA) that extracts a BAF for any chosen movie and provides a dialectical strength which is comparable with the movie score while at the same time being able to provide dialogical explanations of why a movie obtained its strength via interactions between the user and ADA. The second setting we considered was the hotel/restaurant review domain. We showed how RbAM and quantitative gradual semantics can be used to mine a temporal-based topic-dependent BAF that can then be used in explaining the reasoning within the reviews as well as providing strengths of aspects identified in reviews. We showed that the mined BAFs contribute a novel argumentative feature that, when used in combination with other features in standard supervised classifiers, outperforms the latter in the task of detecting whether hotel/restaurant reviews are deceptive. As BAFs are well studied in the Argumentation literature but are difficult to manually annotate in the presence of many arguments, we have shown how to use RbAM to mine them and to benefit applications.

Explaining ML has attracted a lot of interest recently, from explaining black box models [82, 166, 167] to explaining classifications in NLP [6, 106]. Argumentation has also been used to generate explanations in AI, with a focus on explaining decisions [3], classifications [4], and recommendations [28]. We ported the explanations of AA-CBR [48, 49], an Abstract Argumentation (AA)-driven Case-Based Reasoning (CBR) model, to explanations of classification of categorical data, and then to classification of text sentiment by proposing an upgraded version of AA-CBR to accommodate the richness of text. We used AA-CBR, originally designed for the task of deciding and explaining outcomes of new cases based on past cases, to mine AAFs from data and to use them for classification while also providing explanations. We illustrated two types of explanations that can be extracted from AAFs, counterfactual and dialectical, and showed that our method is competitive in predicting when compared to Decision Trees (also explainable by means of rules).

7.2. Future work

Each chapter of this thesis has contributed to the field of AM not only for what has been achieved in this work but also the potential avenues for future research which have been highlighted. Here we discuss directions for future research with an emphasis on portability and explainability.

7.2.1. Portability

The first line of future work we foresee is concerned with the generalisability of our approach to other NLP domains. We plan to deploy our methodologies for mining BAFs on other aggregation websites e.g. Amazon for products and Metacritic for movies and music.

For AA-CBR, we plan to deploy our method on the full Amazon [127] and IMDB [124] datasets, followed by experimentation with other datasets in NLP classification tasks. In addition, we plan to test whether using n-grams (i.e. taking the order of words into account) can improve the results, before exploring the use of embeddings. Further modifications need to be brought to AA-CBR to address the continuous space, for example, changing the attack definition.

Our models for 2 class RbAM perform homogeneously over all datasets consisting of attack and support relation. We are interested to see if we can use these models as starting point to achieve results comparable with the state-of-the-art on the respective datasets.

The field of (Rb)AM is, similarly to most ML tasks, data silos. The solutions proposed are largely focused on building systems that rely on discriminative features from the dataset of analysis, datasets that are manually annotated for specific tasks. In order to advance the current solutions, we need systems that learn better and that can be applied more widely, as opposed to what is achieved on individual datasets and tasks (e.g. Sentiment Analysis in the movie domain and Sentiment Analysis on Twitter). One way to address this is by transfer learning [142, 198] where the aim is to step up from supervised learning methods that capture the salient features of the dataset used for the experimental setting but that, in most cases, fail to generalise.

Transfer learning is one direction of future work we plan to explore. In the RbAM downstream task, we will investigate transfer learning by using the model trained on one dataset as the starting point for models on the remaining datasets. Thus, the model will be trained twice, firstly on the dataset of choice, and secondly, on the new dataset by keeping the trained model's weights and using them in the new training task. In addition, we plan to experiment whether including features related to the specific domain used during training can bring further improvements and thus compete with the state-of-the-art on the respective datasets.

One issue with AM, shared with other NLP tasks, is that words have different meanings depending on the context. For example, if we consider tech items, the

word *compact* will likely carry a positive meaning when travelling and hence indicate a supporting argument or an argument for acquiring such items. However, the word *compact*, when associated with a room will carry a negative meaning and hence indicate an attacking argument or an argument against renting or buying that particular room.

One way to addres this issue (and that relates to transfer learning) is the choice of word embeddings. Recently, contextualized word embeddings were introduced: Embeddings from Language Models (ELMo) [152] and Bidirectional Encoder Representations from Transformers (BERT) [51], that analyse the entire sentence before assigning an embedding to each word. For example, for the sentence *You were right to turn right at the light*, ELMo and BERT would generate different vectors for the two words *right*: the first being closer to words like *correct, accurate* and the second being closer to words such as *left, forward*, thus being context sensitive. The main difference between GloVE and these new embeddings is that GloVE does not take the word order into account during training, whereas ELMo and BERT do. ELMo uses LSTMs and BERT uses a transformer model [188]. While for GloVE vectors we do not need the model in order to use the embeddings, for the contextualied word embeddings we require the pre-trained language models that we can then fine tune using the datasets of the downstream task. These contextualized word embeddings have increased the (successful) applications of transfer learning in NLP, a path we plan to explore.

Creating a model that achieves comparable results to annotators could lead to the deployment of the model on unlabelled text to create new corpora, highly needed in experimentation. Improving our models proposed for the RbAM task could thus lead to larger datasets. While human annotators might still be required to check the labels assigned to unseen texts, the burden of creation of new datasets is, at least, reduced. Automatic corpus extension and lexicon creator have been previously employed with satisfactory results [70, 125].

7.2.2. Explainability

Our explanations were obtained from the mined AFs. We envisage other ways of mining AFs than the ones considered in this thesis. One direction we would like to explore is concerned with graph embeddings [34, 79], thus learning the graphs directly from data, rather than having a two stage pipeline: first learning the relations and then constructing the graph. These can lead to different (types of) AFs

extracted from text. For example, in [160], an argumentation-driven recommender system is defined to provide explanations for recommendations, based on movie scores solely, extracted from user-tailored Tripolar AFs.

For AA-CBR, the use of embeddings may lead to further modifications to the attack definition. We will explore whether keeping word embeddings or combining the word embeddings (e.g. averaging or concatenation of vectors) to obtain a single vector representation for the text has an impact on classification performance and how these changes impact the current (types of) explanations. If a merging technique is used to represent instances, we will need to recover the original features in order to provide explanations that can be understood by users. The design of a framework that addresses these challenges is left for future work.

Our explanations were drawn from the BAFs mined from text. Different methods for mining BAFs may lead to other (templates of) explanations that can be generated. For mining BAFs, future work includes experimenting with different methods for extracting arguments. In particular, we plan to experiment whether first determining arguments based on their argumentative structure, e.g. as in [85, 120], may be useful to single out chunks of text to be fed into RbAM, as well as to identify the arguments relevant for the issue considered [190] and to define a way to combine them rather than assuming that arguments are represented by single phrases.

Conversational AI with applications to recommendations [45, 183, 202] is another line of future work that we would like to explore. We aim for the conversation to be transparent in the sense that the system will provide an explanation in addition to its input. For example, if a user asks to book a hotel in a city, the system could, after filtering hotels in that city, provide a reply as follows: *I recommend you hotel X. It is in the centre of the city and you like to be in the centre and close to tourist attractions*. In the movie setting, the system could reply: *You should watch X. It is directed by D and you have seen many movies directed by D*. There are other ways in which explanations in recommendations can be explored. Starting from one domain, we can make recommendations to a related item from another domain, thus "transfer" recommendations, and use the common aspect as an explanation. Such a system would, based on user's preferences in one domain, recommend items from other domains. For example, when a user gives a stellar review to a book from the "Sherlock Holmes" series by Sir Arthur Conan Doyle, the system could recommend either to watch the films, the TV series, or the escape room based on Sherlock Holmes. The key aspect in this case is rep-

resented by different domains being tied together. Knowledge graphs [81, 196] could be used in this case to link domains that have common aspects or aspects closely related that could be considered common. One possible recommendation with an explanation put forward by the system could be: *You gave a stellar review to the book "Sherlock Holmes". You might like to explore the TV series as they are based on "Sherlock Holmes"*. Having an AI engaging in a conversation with a user and providing arguments for or against the ideas the user puts forward concerned with a topic of discussion as well as providing recommendations of items from review aggregation websites based on a user's preferences would contribute to this medium of interactive conversations.

In all our experiments, we have used existing semantics such as [162]. It would be interesting to see whether we can define other argumentation semantics that are better suited for the applications we have considered. Similarly, defining other notions of strength is left for future work, as well as defining a different method that computes, rather than uses given, base score for arguments. These changes will affect the way BAFs are evaluated and thus have an impact on the explanations we provide, e.g. leading to other types or templates of explanations than the ones we have illustrated in this thesis.

Finally, we plan to conduct user studies to determine whether the explanations extracted can be easily understood by users. For AA-CBR we plan to conduct user studies to compare the quality of explanations to the quality of those of other methods, e.g. [166]. As we have used BAFs to provide the underlying structure for generating explanations for users, we plan to run user studies to decide on the best way to provide the explanations, e.g. visual or linguistic.

Bibliography

[1] Hojjat Aghakhani, Aravind Machiry, Shirin Nilizadeh, Christopher Kruegel, and Giovanni Vigna. “Detecting Deceptive Reviews Using Generative Adversarial Networks”. In: *2018 IEEE Security and Privacy Workshops*. 2018, pp. 89–95.

[2] Leila Amgoud and Jonathan Ben-Naim. “Evaluation of Arguments in Weighted Bipolar Graphs”. In: *Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 14th European Conference, ECSQARU*. 2017, pp. 25–35.

[3] Leila Amgoud and Henri Prade. “Using Arguments for Making and Explaining Decisions”. In: *Artificial Intelligence* 173.3-4 (2009), pp. 413–436.

[4] Leila Amgoud and Mathieu Serrurier. “Agents that argue and explain classifications”. In: *Autonomous Agents and Multi-Agent Systems* 16.2 (2007), pp. 187–209.

[5] Robert Andrews, Joachim Diederich, and Alan B. Tickle. “Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks”. In: *Knowledge-Based Systems* 8.6 (Dec. 1995), pp. 373–389.

[6] Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. “Explaining Recurrent Neural Network Predictions in Sentiment Analysis”. In: *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP*. 2017, pp. 159–168.

[7] Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Ricardo Simari, Matthias Thimm, and Serena Villata. “Towards Artificial Argumentation”. In: *AI Magazine* 38.3 (2017), pp. 25–36.

[8] Randall Balestriero. “Neural Decision Trees”. In: *CoRR* abs/1702.07360 (2017). arXiv: 1702.07360.

[9] Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. “Stance Classification of Context-Dependent Claims”. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. 2017, pp. 251–261.

[10] Marco Baroni and Alessandro Lenci. “Distributional Memory: A General Framework for Corpus-Based Semantics.” In: *Computational Linguistics* 36.4 (2010), pp. 673–721.

[11] Pietro Baroni, Antonio Rago, and Francesca Toni. “How Many Properties Do We Need for Gradual Argumentation?” In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 2018, pp. 1736–1743.

[12] Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza. “Automatic evaluation of design alternatives with quantitative argumentation”. In: *Argument & Computation* 6.1 (2015), pp. 24–49.

[13] Trevor J. M. Bench-Capon and Paul E. Dunne. “Argumentation in artificial intelligence”. In: *Artificial Intelligence* 171.10-15 (2007), pp. 619–641.

[14] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. “Learning Long-term Dependencies with Gradient Descent is Difficult”. In: *Transactions on Neural Networks* 5.2 (Mar. 1994), pp. 157–166.

[15] Philippe Besnard, Alejandro Javier García, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Ricardo Simari, and Francesca Toni. “Introduction to Structured Argumentation”. In: *Argument & Computation* 5.1 (2014), pp. 1–4.

[16] Floris Bex, Sanjay Modgil, Henry Prakken, and Chris Reed. “On logical specifications of the Argument Interchange Format”. In: *Journal of Logic and Computation* 23.5 (2013), pp. 951–989.

[17] Rupal Bhargava, Anushka Baoni, and Yashvardhan Sharma. “Composite Sequential Modeling for Identifying Fake Reviews”. In: *Journal of Intelligent Systems* (Apr. 2018).

[18] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 2009.

[19] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. “Latent Dirichlet Allocation”. In: *Journal of Machine Learning Research* 3 (Mar. 2003), pp. 993–1022.

[20] Guido Boella, Dov M. Gabbay, Leendert W. N. van der Torre, and Serena Villata. “Support in Abstract Argumentation”. In: *Computational Models of Argument: Proceedings of COMMA*. 2010, pp. 111–122.

[21] Filip Boltužić and Jan Šnajder. “Back up your Stance: Recognizing Arguments in Online Discussions”. In: *Proceedings of the First Workshop on Argumentation Mining*. 2014, pp. 49–58.

[22] Elise Bonzon, Jérôme Delobelle, Sébastien Konieczny, and Nicolas Maudet. “Argumentation Ranking Semantics Based on Propagation”. In: *Computational Models of Argument - Proceedings of COMMA*. 2016, pp. 139–150.

[23] Tom Bosc, Elena Cabrio, and Serena Villata. “Tweeties Squabbling: Positive and Negative Results in Applying Argument Mining on Social Media”. In: *Computational Models of Argument COMMA*. 2016, pp. 21–32.

[24] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A Training Algorithm for Optimal Margin Classifiers”. In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. 1992, pp. 144–152.

[25] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. “A Fast Unified Model for Parsing and Sentence Understanding”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL*. 2016.

[26] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. “A large annotated corpus for learning natural language inference”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2015.

[27] Leo Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32.

[28] C. E. Briguez, M. C. Budán, C. A. D. Deagustini, A. G. Maguitman, M. Capobianco, and G. R. Simari. “Argument-based mixed recommenders and their application to movie suggestion”. In: *Expert Systems with Applications* 41.14 (2014), pp. 6467–6482.

[29] Stefano Bromuri, Visara Urovi, Maxime Morge, Kostas Stathis, and Francesca Toni. “A Multi-agent System for Service Discovery, Selection and Negotiation”. In: *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*. AAMAS. 2009, pp. 1395–1396.

[30] Elena Cabrio and Serena Villata. “Detecting Bipolar Semantic Relations among Natural Language Arguments with Textual Entailment: a Study”. In: *Joint Symposium on Semantic Processing (JSSP)*. Nov. 2013.

[31] Elena Cabrio and Serena Villata. “Five Years of Argument Mining: a Data-driven Analysis”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*. 2018, pp. 5427–5433.

[32] Elena Cabrio and Serena Villata. “Natural Language Arguments: A Combined Approach”. In: *ECAI*. Vol. 242. 2012, pp. 205–210.

[33] Elena Cabrio and Serena Villata. “Towards a Benchmark of Natural Language Arguments”. In: *CoRR* abs/1405.0941 (2014). URL: http://arxiv.org/abs/1405.0941.

[34] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. “A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications”. In: *IEEE Transactions on Knowledge and Data Engineering* 30.9 (2018), pp. 1616–1637.

[35] Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. “e-SNLI: Natural Language Inference with Natural Language Explanations”. In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS*. 2018, pp. 9560–9572.

[36] Emerson F. Cardoso, Renato Moraes Silva, and Tiago A. Almeida. “Towards automatic filtering of fake reviews”. In: *Neurocomputing* 309 (2018), pp. 106–116.

[37] Lucas Carstens and Francesca Toni. “Towards relation based Argumentation Mining”. In: *Proceedings of the 2nd Workshop on Argumentation Mining*. 2015, pp. 29–34.

[38] Lucas Carstens and Francesca Toni. “Using Argumentation to Improve Classification in Natural Language Problems”. In: *ACM Transactions on Internet Technology* 17.3 (2017), 30:1–30:23.

[39] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. “Coalitions of arguments: A tool for handling bipolar argumentation frameworks”. In: *International Journal of Intelligent Systems* 25.1 (2010), pp. 83–109.

[40] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. “On the Acceptability of Arguments in Bipolar Argumentation Frameworks”. In: *Proceedings of the 8th European conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. 2005, pp. 378–389.

[41] Federico Cerutti, Alexis Palmer, Ariel Rosenfeld, Jan Snajder, and Francesca Toni. “A Pilot Study in Using Argumentation Frameworks for Online Debates”. In: *Proceedings of the First International Workshop on Systems and Algorithms for Formal Argumentation (SAFA)*. Vol. 1672. 2016, pp. 63–74.

[42] Carlos Iván Chesñevar and Guillermo Ricardo Simari. “A Lattice-based Approach to Computing Warranted Beliefs in Skeptical Argumentation Frameworks”. In: *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. IJCAI’07. 2007, pp. 280–285.

[43] Carlos Iván Chesñevar, Jarred McGinnis, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Ricardo Simari, Matthew South,

Gerard Vreeswijk, and Steven Willmott. “Towards an argument interchange format”. In: *Knowledge Engineering Review* 21.4 (2006), pp. 293–316.

[44] Kyunghyun Cho, Bart van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Oct. 2014, pp. 1724–1734.

[45] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. “Towards Conversational Recommender Systems”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 815–824.

[46] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling”. In: *CoRR* abs/1412.3555 (2014). URL: http://arxiv.org/abs/1412.3555.

[47] Michael Crawford, TaghiM. Khoshgoftaar, JosephD. Prusa, AaronN. Richter, and Hamzah Al Najada. “Survey of review spam detection using machine learning techniques”. In: *Journal of Big Data* 2.1 (2015), pp. 1–24.

[48] Kristijonas Čyras, Ken Satoh, and Francesca Toni. “Abstract Argumentation for Case-Based Reasoning”. In: *Principles of Knowledge Representation and Reasoning, 15th International Conference*. 2016, pp. 549–552.

[49] Kristijonas Čyras, Ken Satoh, and Francesca Toni. “Explanation for Case-Based Reasoning via Abstract Argumentation”. In: *6th International Conference on Computational Models of Argument*. 2016, pp. 243–254.

[50] Tom De Smedt and Walter Daelemans. “Pattern for Python”. In: *Journal of Machine Learning Research* 13 (2012), pp. 2031–2035.

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR* abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805.

[52] Dua Dheeru and Efi Karra Taniskidou. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[53] Hai Ha Do, P. W. C. Prasad, Angelika Maag, and Abeer Alsadoon. “Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review”. In: *Expert Systems with Applications* 118 (2019), pp. 272–299.

[54] P. M. Dung, P. Mancarella, and F. Toni. “Computing Ideal Sceptical Argumentation”. In: *Artificial Intelligence* 171.10-15 (July 2007), pp. 642–674.

[55] Phan Minh Dung. “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games”. In: *Artificial Intelligence* 77.2 (1995), pp. 321 –357.

[56] Phan Minh Dung, Robert Kowalski, and Francesca Toni. “Dialectic Proof Procedures for Assumption-Based, Admissible Argumentation”. In: *Artificial Intelligence* 170.2 (2006), pp. 114–159.

[57] Phan Minh Dung, Robert A. Kowalski, and Francesca Toni. “Assumption-Based Argumentation”. In: *Argumentation in Artificial Intelligence*. 2009, pp. 199–218.

[58] Mihai Dusmanu, Elena Cabrio, and Serena Villata. “Argument Mining on Twitter: Arguments, Facts and Sources”. In: *The 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*. Sept. 2017.

[59] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. “Neural End-to-End Learning for Computational Argumentation Mining”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 11–22.

[60] Jeffrey L. Elman. “Finding structure in time”. In: *Cognitive Science* 14.2 (1990), pp. 179–211.

[61] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. “Why Does Unsupervised Pre-training Help Deep Learning?” In: *Journal of Machine Learning Research* 11 (2010), pp. 625–660.

[62] Andrea Esuli and Fabrizio Sebastiani. “SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining”. In: *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC’06*. 2006, pp. 417–422.

[63] Xiuyi Fan and Francesca Toni. “On Computing Explanations in Argumentation”. In: *Proceedings of AAAI*. 2015, pp. 1496–1502.

[64] Xiuyi Fan and Francesca Toni. “On Explanations for Non-Acceptable Arguments”. In: *Theory and Applications of Formal Argumentation - Third International Workshop, TAFA*. 2015, pp. 112–127.

[65] Vanessa Wei Feng and Graeme Hirst. “Classifying arguments by scheme”. In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*. 2011, pp. 987–996.

[66] Stefano Ferilli, Andrea Pazienza, Sergio Angelastro, and Alessandro Suglia. “A Similarity-Based Abstract Argumentation Approach to Extractive Text Summarization”. In: *AI*IA 2017 Advances in Artificial Intelligence - XVIth International Conference of the Italian Association for Artificial Intelligence*. 2017, pp. 87–100.

[67] J.L. Fleiss et al. “Measuring nominal scale agreement among many raters”. In: *Psychological Bulletin* 76.5 (1971), pp. 378–382.

[68] Julien Fontanarava, Gabriella Pasi, and Marco Viviani. “Feature Analysis for Fake Review Detection through Supervised Classification”. In: *2017 IEEE International Conference on Data Science and Advanced Analytics, DSAA*. 2017, pp. 658–666.

[69] John Fox, David Glasspool, Dan Grecu, Sanjay Modgil, Matthew South, and Vivek Patkar. “Argumentation-Based Inference and Decision Making–A Medical Perspective”. In: *IEEE Intelligent Systems* 22.6 (2007), pp. 34–41.

[70] Valerij Fredriksen, Brage Ekroll Jahren, and Björn Gambäck. “Utilizing Large Twitter Corpora to Create Sentiment Lexica”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC*. 2018, pp. 2829–2836.

[71] Christie M. Fuller, David P. Biros, Douglas P. Twitchell, Judee K. Burgoon, and Mark Adkins. “An Analysis of Text-Based Deception Detection Tools”. In: *12th Americas Conference on Information Systems*. 2006, p. 418.

[72] Donato Hernández Fusilier, Manuel Montes-y-Gómez, Paolo Rosso, and Rafael Guzmán-Cabrera. “Detecting positive and negative deceptive opinions using PU-learning”. In: *Information Processing & Management* 51.4 (2015), pp. 433–443.

[73] Alejandro Javier García and Guillermo Ricardo Simari. “Defeasible Logic Programming: An Argumentative Approach”. In: *Theory and Practice of Logic Programming* 4.1-2 (2004), pp. 95–138.

[74] Alejandro Javier García, Carlos Iván Chesñevar, Nicolás D. Rotstein, and Guillermo Ricardo Simari. “Formalizing dialectical explanation support for argument-based reasoning in knowledge-based systems”. In: *Expert Systems with Applications* 40.8 (2013), pp. 3233–3247.

[75] Eduardo Gasca, José Salvador Sánchez, and R. Alonso. “Eliminating redundancy and irrelevance using a new MLP-based feature selection method.” In: *Pattern Recognition* 39.2 (2006), pp. 313–315.

[76] Fatih Gedikli, Dietmar Jannach, and Mouzhi Ge. “How should I explain? A comparison of different explanation types for recommender systems”. In: *International Journal of Human-Computer Studies* 72.4 (2014), pp. 367–382.

[77] Aryo Pradipta Gema, Suhendro Winton, Theodorus David, Derwin Suhartono, Muhsin Shodiq, and Wikaria Gazali. “It Takes Two To Tango: Modification of Siamese Long Short Term Memory Network with Attention Mechanism in Recognizing Argumentative Relations in Persuasive Essay”. In: *Procedia Computer Science* 116 (2017), pp. 449–459.

[78] Theodosios Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. “Argument Extraction from News, Blogs, and Social Media”. In: *Artificial Intelligence: Methods and Applications - 8th Hellenic Conference on AI, SETN*. 2014, pp. 287–299.

[79] Palash Goyal and Emilio Ferrara. “Graph embedding techniques, applications, and performance: A survey”. In: *Knowledge-Based Systems* 151 (2018), pp. 78–94.

[80] Kathrin Grosse, María Paula González, Carlos Iván Chesñevar, and Ana Gabriela Maguitman. “Integrating argumentation and sentiment analysis for mining opinions from Twitter”. In: *AI Communications* 28.3 (2015), pp. 387–401.

[81] Niannian Guan, Dandan Song, and Lejian Liao. “Knowledge graph embedding with concepts”. In: *Knowledge-Based Systems* 164 (2019), pp. 38–44.

[82] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. “A Survey of Methods for Explaining Black Box Models”. In: *ACM Computing Surveys* 51.5 (2019), 93:1–93:42.

[83] I. Habernal and I. Gurevych. “Argumentation Mining in User-generated Web Discourse”. In: *Computational Linguistics* 43.1 (2017), pp. 125–179.

[84] Ivan Habernal, Judith Eckle-Kohler, and Iryna Gurevych. “Argumentation Mining on the Web from Information Seeking Perspective”. In: *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*. 2014.

[85] Ivan Habernal and Iryna Gurevych. “Argumentation Mining in User-Generated Web Discourse”. In: *Computational Linguistics* 43.1 (2017), pp. 125–179.

[86] Ivan Habernal and Iryna Gurevych. “What makes a convincing argument? Empirical analysis and detecting attributes of convincingness in Web argumentation”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 2016, pp. 1214–1223.

[87] Emmanuel Hadoux, Anthony Hunter, and Jean-Baptiste Corrégé. “Strategic Dialogical Argumentation Using Multi-criteria Decision Making with Application to Epistemic and Emotional Aspects of

Arguments". In: *Foundations of Information and Knowledge Systems - 10th International Symposium, FoIKS*. 2018, pp. 207–224.

[88] Kai Han, Chao Li, and Xin Shi. "Autoencoder Feature Selector". In: *CoRR* abs/1710.08310 (2017). URL: http://arxiv.org/abs/1710.08310.

[89] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. "Generating Counterfactual Explanations with Natural Language". In: *CoRR* abs/1806.09809 (2018). URL: http://arxiv.org/abs/1806.09809.

[90] Geoffrey Hinton and Ruslan Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006), pp. 504 –507.

[91] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780.

[92] Xinyu Hua and Lu Wang. "Understanding and Detecting Diverse Supporting Arguments on Controversial Issues". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL*. 2017, pp. 203–208.

[93] Drew A. Hudson and Christopher D. Manning. "Compositional Attention Networks for Machine Reasoning". In: *CoRR* abs/1803.03067 (2018). URL: http://arxiv.org/abs/1803.03067.

[94] Clayton J. Hutto and Eric Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM*. 2014.

[95] Rahwan Iyad and Chris Reed. "The Argument Interchange Format". In: *Argumentation in Artificial Intelligence*. 2009, pp. 383–402.

[96] Sarthak Jain and Byron C. Wallace. "Attention is not Explanation". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT* (2019). URL: http://arxiv.org/abs/1902.10186. To appear.

[97] Nitin Jindal and Bing Liu. "Analyzing and Detecting Review Spam". In: *Proceedings of the 7th IEEE International Conference on Data Mining*. 2007, pp. 547–552.

[98] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. "FastText.zip: Compressing text classification models". In: *CoRR* (2016). URL: http://arxiv.org/abs/1612.03651.

[99] Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. “An Empirical Exploration of Recurrent Network Architectures”. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML*. 2015, pp. 2342–2350.

[100] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR*. 2015.

[101] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulò. “Deep Neural Decision Forests”. In: *Proceedings of IJCAI*. 2016, pp. 4190–4194.

[102] Yuta Koreeda, Toshihiko Yanase, Kohsuke Yanai, Misa Sato, and Yoshiki Niwa. “Neural Attention Model for Classification of Sentences that Support Promoting/Suppressing Relationship”. In: *Proceedings of the Third Workshop on Argument Mining*. 2016.

[103] Dimitrios Kotzias, Misha Denil, Nando de Freitas, and Padhraic Smyth. “From Group to Individual Labels Using Deep Features”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2015, pp. 597–606.

[104] Robert A. Kowalski and Francesca Toni. “Abstract Argumentation”. In: *Artificial Intelligence and Law* 4.3-4 (Sept. 1996), pp. 275–296.

[105] Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology (second edition)*. 2004.

[106] Vivian Lai and Chenhao Tan. “On Human Predictions with Explanations and Predictions of Machine Learning Models: A Case Study on Deception Detection”. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT*. 2019, pp. 29–38.

[107] John Lawrence and Chris Reed. “Argument Mining Using Argumentation Scheme Structures”. In: *Computational Models of Argument - Proceedings of COMMA*. 2016, pp. 379–390.

[108] John Lawrence and Chris Reed. “Combining Argument Mining Techniques”. In: *Proceedings of the 2nd Workshop on Argumentation Mining*. 2015, pp. 127–136.

[109] John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. “AIFdb: Infrastructure for the Argument Web”. In: *Computational Models of Argument - Proceedings of COMMA*. Vol. 245. 2012, pp. 515–516.

[110] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444.

[111] Daniel D. Lee and H. Sebastian Seung. “Learning the parts of objects by nonnegative matrix factorization”. In: *Nature* 401 (1999), pp. 788–791.

[112] Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. “Rationalizing Neural Predictions”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 2016, pp. 107–117.

[113] Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. “Context Dependent Claim Detection”. In: *COLING 2014, 25th International Conference on Computational Linguistics*. 2014, pp. 1489–1500.

[114] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. “Learning to Identify Review Spam”. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*. 2011, pp. 2488–2493.

[115] Jiwei Li, Myle Ott, Claire Cardie, and Eduard H. Hovy. “Towards a General Rule for Identifying Deceptive Opinion Spam”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL*. 2014, pp. 1566–1576.

[116] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. “Detecting Product Review Spammers Using Rating Behaviors”. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. 2010, pp. 939–948.

[117] Marco Lippi and Paolo Torroni. “Argument Mining: A Machine Learning Perspective”. In: *The 2015 International Workshop on Theory and Applications of Formal Argument*. 2015.

[118] Marco Lippi and Paolo Torroni. “Argumentation Mining: State of the Art and Emerging Trends”. In: *ACM Transactions on Internet Technology* 16.2 (2016), p. 10.

[119] Marco Lippi and Paolo Torroni. “Context-Independent Claim Detection for Argument Mining”. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*. 2015, pp. 185–191.

[120] Marco Lippi and Paolo Torroni. “MARGOT: A web server for argumentation mining”. In: *Expert Systems with Applications* 65 (2016), pp. 292 –303.

[121] Bing Liu. *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.

[122] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. 2012.

[123] Pengfei Liu, Xipeng Qiu, Yaqian Zhou, Jifan Chen, and Xuanjing Huang. “Modelling Interaction of Sentence Pair with Coupled-LSTMs”. In:

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP. 2016, pp. 1703–1712.

[124] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. "Learning Word Vectors for Sentiment Analysis". In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 142–150.

[125] Elena Manishina, Bassam Jabaian, Stéphane Huet, and Fabrice Lefèvre. "Automatic Corpus Extension for Data-driven Natural Language Generation". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC*. 2016, pp. 3624–3631.

[126] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. "Building a Large Annotated Corpus of English: The Penn Treebank". In: *Computational Linguistics* 19.2 (June 1993), pp. 313–330.

[127] Julian J. McAuley and Jure Leskovec. "Hidden factors and hidden topics: understanding rating dimensions with review text". In: *Seventh ACM Conference on Recommender Systems, RecSys*. 2013, pp. 165–172.

[128] Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. "Never Retreat, Never Retract: Argumentation Analysis for Political Speeches". In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 2018, pp. 4889–4896.

[129] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. "Advances in Pre-Training Distributed Word Representations". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*. 2018.

[130] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 3111–3119.

[131] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR*. 2013.

[132] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. "Recurrent neural network based language model". In: *INTERSPEECH*. 2010, pp. 1045–1048.

[133] George A. Miller. "WordNet: A Lexical Database for English". In: *Communications of the ACM* 38 (1995), pp. 39–41.

[134] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267 (2019), pp. 1–38.

[135] S Modgil, F Toni, F Bex, I Bratko, CI Chesñevar, W Dvoák, MA Falappa, X Fan, SA Gaggl, AJ Garcia, MP Gonzalez, TF Gordon, J Leite, M Moina, C Reed, GR Simari, S Szeider, P Torroni, and S Woltran. "The added value of argumentation". In: *Agreement Technologies*. 2013, pp. 357–403.

[136] Arjun Mukherjee, Bing Liu, and Natalie Glance. "Spotting Fake Reviewer Groups in Consumer Reviews". In: *Proceedings of the 21st International Conference on World Wide Web*. 2012, pp. 191–200.

[137] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malú Castellanos, and Riddhiman Ghosh. "Spotting opinion spammers using behavioral footprints". In: *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*. 2013, pp. 632–640.

[138] Vlad Niculae, Joonsuk Park, and Claire Cardie. "Argument Mining with Structured SVMs and RNNs". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, pp. 985–995.

[139] Myle Ott, Claire Cardie, and Jeffrey T. Hancock. "Negative Deceptive Opinion Spam". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013.

[140] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. "Finding Deceptive Opinion Spam by Any Stretch of the Imagination". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 309–319.

[141] Raquel Mochales Palau and Marie-Francine Moens. "Argumentation mining". In: *Artificial Intelligence and Law* 19.1 (2011), pp. 1–22.

[142] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and data engineering* 22.10 (2010), pp. 1345–1359.

[143] Bo Pang and Lillian Lee. "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts". In: *Proceedings of the Association for Computational Linguistics (ACL)*. 2004, pp. 271–278.

[144] Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. "A Decomposable Attention Model for Natural Language Inference". In: *EMNLP*. 2016, pp. 2249–2255.

[145] Joonsuk Park and Claire Cardie. "A Corpus of eRulemaking User Comments for Measuring Evaluability of Arguments". In: *Proceedings of*

the Eleventh International Conference on Language Resources and Evaluation, LREC. 2018.

[146] Theodore Patkos, Antonis Bikakis, and Giorgos Flouris. "A Multi-Aspect Evaluation Framework for Comments on the Social Web". In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR*. 2016, pp. 593–596.

[147] Andrea Pazienza, Stefano Ferilli, and Floriana Esposito. "Constructing and Evaluating Bipolar Weighted Argumentation Frameworks for Online Debating Systems". In: *Proceedings of the 1st Workshop on Advances In Argumentation In Artificial Intelligence co-located with XVI International Conference of the Italian Association for Artificial Intelligence, AI^3@AI*IA*. 2017, pp. 111–125.

[148] Andreas Peldszus and Manfred Stede. "From Argument Diagrams to Argumentation Mining in Texts: A Survey". In: *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* 7.1 (2013), pp. 1–31.

[149] Andreas Peldszus and Manfred Stede. "Joint prediction in MST-style discourse parsing for argumentation mining". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 2015, pp. 938–948.

[150] Andreas Peldszus and Manfred Stede. "Ranking the annotators: An agreement study on argumentation structure". In: *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. 2013, pp. 196–204.

[151] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 1532–1543.

[152] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. 2018, pp. 2227–2237.

[153] Lahari Poddar, Wynne Hsu, and Mong-Li Lee. "Author-aware Aspect Topic Sentiment Model to Retrieve Supporting Opinions from Reviews". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 2017, pp. 483–492.

[154] Nina Pörner, Hinrich Schütze, and Benjamin Roth. "Evaluating neural network explanation methods using hybrid documents and morphosyntactic agreement". In: *Proceedings of the 56th Annual Meeting*

of the Association for Computational Linguistics, ACL. 2018, pp. 340–350.

[155] M. F. Porter. “An algorithm for suffix stripping”. In: *Program* 14.3 (1980), pp. 130–137.

[156] M. F. Porter. “Snowball: A language for stemming algorithms”. In: (2001). URL: http://snowball.tartarus.org/texts/introduction.html.

[157] Peter Potash, Alexey Romanov, and Anna Rumshisky. “Here’s My Point: Joint Pointer Architecture for Argument Mining”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 2017, pp. 1364–1373.

[158] Henry Prakken. “Coherence and Flexibility in Dialogue Games for Argumentation”. In: *Journal of Logic and Computation* 15.6 (2005), pp. 1009–1040.

[159] Henry Prakken. “Formal systems for persuasion dialogue”. In: *Knowledge Engineering Review* 21.2 (2006), pp. 163–188.

[160] Antonio Rago, Oana Cocarascu, and Francesca Toni. “Argumentation-Based Recommendations: Fantastic Explanations and How to Find Them”. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*. 2018, pp. 1949–1955.

[161] Antonio Rago and Francesca Toni. “Quantitative Argumentation Debates with Votes for Opinion Polling”. In: *PRIMA*. 2017, pp. 369–385.

[162] Antonio Rago, Francesca Toni, Marco Aurisicchio, and Pietro Baroni. “Discontinuity-Free Decision Support with Quantitative Argumentation Debates”. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR*. 2016, pp. 63–73.

[163] Toqir Ahmad Rana and Yu-N Cheah. “Aspect extraction in sentiment analysis: comparative analysis and survey”. In: *Artificial Intelligence Review* 46.4 (2016), pp. 459–483.

[164] Chris Reed, Simon Wells, Joseph Devereux, and Glenn Rowe. “AIF+: Dialogue in the Argument Interchange Format”. In: *Computational Models of Argument: Proceedings of COMMA*. 2008, pp. 311–323.

[165] Chris Reed, Simon Wells, Katarzyna Budzynska, and Joseph Devereux. “Building arguments with argumentation: the role of illocutionary force in computational models of argument”. In: *Computational Models of Argument: Proceedings of COMMA*. 2010, pp. 415–426.

[166] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. “Anchors: High-Precision Model-Agnostic Explanations”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[167] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, pp. 1135–1144.

[168] Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomás Kociský, and Phil Blunsom. "Reasoning about Entailment with Neural Attention". In: *4th International Conference on Learning Representations, ICLR*. 2016.

[169] Nicolás D. Rotstein, Martín O. Moguillansky, and Guillermo R. Simari. "Dialectical Abstract Argumentation: A Characterization of the Marking Criterion". In: *Proceedings of the 21st International Jont Conference on Artifical Intelligence*. IJCAI'09. 2009, pp. 898–903.

[170] Jitendra Kumar Rout, Smriti Singh, Sanjay Kumar Jena, and Sambit Bakshi. "Deceptive review detection using labeled and unlabeled data". In: *Multimedia Tools and Applications* 76.3 (2017), pp. 3187–3211.

[171] Patrick Saint-Dizier. "Argument Mining: the Bottleneck of Knowledge and Language Resources". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC*. 2016.

[172] Zaher Salah, Frans Coenen, and Davide Grossi. "Extracting Debate Graphs from Parliamentary Transcripts: A Study Directed at UK House of Commons Debates". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*. 2013, pp. 121–130.

[173] Vlad Sandulescu and Martin Ester. "Detecting Singleton Review Spammers Using Semantic Similarity". In: *Proceedings of the 24th International Conference on World Wide Web*. 2015, pp. 971–976.

[174] Mike Schuster and K. Kuldip Paliwal. "Bidirectional Recurrent Neural Networks". In: *Transactions on Signal Processing* 45.11 (Nov. 1997), pp. 2673–2681.

[175] Somayeh Shojaee, Masrah Azrifah Azmi Murad, Azreen bin Azman, Nurfadhlina Mohd Sharef, and Samaneh Nadali. "Detecting deceptive reviews using lexical and syntactic features". In: *13th International Conference on Intellient Systems Design and Applications, ISDA*. 2013, pp. 53–58.

[176] Kacper Sokol and Peter A. Flach. "Conversational Explanations of Machine Learning Predictions Through Class-contrastive Counterfactual Statements". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*. 2018, pp. 5785–5786.

[177] Kacper Sokol and Peter A. Flach. "Counterfactual Explanations of Machine Learning Predictions: Opportunities and Challenges for AI

Safety". In: *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence*. 2019.

[178] Christian Stab and Iryna Gurevych. "Parsing Argumentation Structures in Persuasive Essays". In: *Computational Linguistics* 43.3 (2017), pp. 619–659.

[179] Christian Stab, Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. "Argumentation Mining in Persuasive Essays and Scientific Articles from the Discourse Structure Perspective". In: *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing*. 2014.

[180] Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. "Cross-topic Argument Mining from Heterogeneous Sources". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Oct. 2018.

[181] Manfred Stede and Jodi Schneider. *Argumentation Mining*. Vol. 40. Synthesis Lectures in Human Language Technology. 2018.

[182] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. "Exploring Topic Coherence over Many Models and Many Topics". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 2012, pp. 952–961.

[183] Yueming Sun and Yi Zhang. "Conversational Recommender System". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval,SIGIR*. 2018, pp. 235–244.

[184] Milagro Teruel, Cristian Cardellino, Fernando Cardellino, Laura Alonso Alemany, and Serena Villata. "Increasing Argument Annotation Reproducibility by Using Inter-annotator Agreement to Improve Guidelines". In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC*. 2018.

[185] Simone Teufel, Advaith Siddharthan, and Colin R. Batchelor. "Towards Domain-Independent Argumentative Zoning: Evidence from Chemistry and Computational Linguistics". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 2009, pp. 1493–1502.

[186] Francesca Toni. "A tutorial on assumption-based argumentation". In: *Argument & Computation* 5.1 (2014), pp. 89–117.

[187] Stephen E. Toulmin. *The Uses of Argument*. 2003.

[188] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information*

Processing Systems 30: Annual Conference on Neural Information Processing Systems. 2017, pp. 6000–6010.

[189] A. Verikas and M. Bacauskiene. "Feature Selection with Neural Networks". In: *Pattern Recogn. Lett.* 23.11 (2002), pp. 1323–1335.

[190] Henning Wachsmuth, Benno Stein, and Yamen Ajjour. ""PageRank" for Argument Relevance". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 17)*. Apr. 2017, pp. 1116–1126.

[191] Henning Wachsmuth, Martin Trenkmann, Benno Stein, and Gregor Engels. "Modeling Review Argumentation for Robust Sentiment Analysis". In: *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*. 2014, pp. 553–564.

[192] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. "Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR". In: *CoRR* abs/1711.00399 (2017). URL: http://arxiv.org/abs/1711.00399.

[193] Marilyn A. Walker, Jean E. Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. "A Corpus for Research on Deliberation and Debate". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC*. 2012, pp. 812–817.

[194] Douglas Walton, Chris Reed, and Fabrizio Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.

[195] Douglas N. Walton. *Argumentation Schemes for Presumptive Reasoning*. L. Erlbaum Associates, 1996.

[196] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. "Knowledge Graph Embedding: A Survey of Approaches and Applications". In: *IEEE Transactions on Knowledge and Data Engineering* 29.12 (2017), pp. 2724–2743.

[197] Shuyang Wang, Zhengming Ding, and Yun Fu. "Feature Selection Guided Auto-Encoder". In: *Proceedings of AAAI*. 2017, pp. 2725–2731.

[198] Karl R. Weiss, Taghi M. Khoshgoftaar, and Dingding Wang. "A survey of transfer learning". In: *Journal of Big Data* 3 (2016), p. 9.

[199] Adam Wyner, Jodi Schneider, Katie Atkinson, and Trevor J. M. Bench-Capon. "Semi-Automated Argumentative Analysis of Online Product Reviews". In: *Computational Models of Argument - Proceedings of COMMA*. 2012, pp. 43–50.

[200] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. "Hierarchical Attention Networks for Document

Classification". In: *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 1480–1489.

[201] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. "ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs". In: *TACL* 4 (2016), pp. 259–272.

[202] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. "Towards Conversational Search and Recommendation: System Ask, User Respond". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM*. 2018, pp. 177–186.

[203] Jun Zhao, Kang Liu, and Liheng Xu. "Sentiment Analysis: Mining Opinions, Sentiments, and Emotions". In: *Computational Linguistics* 42.3 (2016), pp. 595–598.

[204] Siyuan Zhao, Zhiwei Xu, Limin Liu, and Mengjie Guo. "Towards Accurate Deceptive Opinion Spam Detection based on Word Order-preserving CNN". In: *CoRR* abs/1711.09181 (2017). URL: http://arxiv.org/abs/1711.09181.

[205] Weizhong Zhao, James J. Chen, Roger Perkins, Zhichao Liu, Weigong Ge, Yijun Ding, and Wen Zou. "A heuristic approach to determine an appropriate number of topics in topic modeling". In: *BMC Bioinformatics*. 2015.

[206] Qiaoting Zhong, Xiuyi Fan, Xudong Luo, and Francesca Toni. "An explainable multi-attribute decision model based on argumentation". In: *Expert Systems with Applications* 117 (2019), pp. 42–61.

[207] Lina Zhou, Judee K. Burgoon, Jay F. Nunamaker, and Doug Twitchell. "Automating Linguistics-Based Cues for Detecting Deception in Text-Based Asynchronous Computer-Mediated Communications". In: *Group Decision and Negotiation* 13.1 (2004), pp. 81–106.

[208] Zhi-Hua Zhou and Ji Feng. "Deep Forest: Towards An Alternative to Deep Neural Networks". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*. 2017, pp. 3553–3559.

Appendices

A. 2 class Relation-based Argument Mining

A.1. 100 words for sequence length

A.1.1. Main training dataset: *aif*

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
AE FT	F_1 A		0.41	0.57	0.60	0.54	0.60	-	0.69	0.63		0.577
	F_1 S		0.36	0.20	0.22	0.27	0.36	0.36	0.12	0.35		0.280
AE G	F_1 A		0.41	0.55	0.58	0.53	0.57	-	0.69	0.62		0.564
	F_1 S		0.43	0.33	0.26	0.43	0.40	0.42	0.14	0.44		0.356
A G	F_1 A	0.57	0.42	0.59		0.53	0.60	-	0.69	0.60		0.571
	F_1 S	0.38	0.41	0.54		0.52	0.25	0.48	0.13	0.48		0.399
M G	F_1 A	0.49	0.40	0.50		0.50	0.53	-	0.57	0.48		0.496
	F_1 S	0.49	0.63	0.60		0.50	0.44	0.67	0.38	0.55		0.533
A G	F_1 A	0.59	0.41	0.60	0.57	0.53		-	0.70	0.62		0.574
	F_1 S	0.34	0.30	0.51	0.34	0.49		0.41	0.08	0.47		0.368
C FT	F_1 A	0.56	0.41	0.55	0.50	0.55		-	0.60	0.55		0.531
	F_1 S	0.42	0.48	0.53	0.54	0.34		0.49	0.40	0.50		0.463
C G	F_1 A	0.38	0.26	0.48	0.26	0.48		-	0.43	0.41		0.386
	F_1 S	0.57	0.71	0.65	0.67	0.54		0.81	0.52	0.57		0.630
M FT	F_1 A	0.52	0.40	0.58	0.50	0.52		-	0.58	0.52		0.517
	F_1 S	0.47	0.50	0.61	0.54	0.54		0.60	0.42	0.53		0.526
M G	F_1 A	0.49	0.40	0.52	0.42	0.49		-	0.53	0.48		0.476
	F_1 S	0.51	0.64	0.61	0.60	0.52		0.69	0.43	0.56		0.570

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
AE FT	F_1 A		0.40	0.51	0.57	0.47	0.57	-	0.66	0.62		0.543
	F_1 S		0.48	0.35	0.35	0.39	0.44	0.52	0.25	0.45		0.404
AE G	F_1 A		0.41	0.47	0.57	0.46	0.56	-	0.65	0.62		0.534
	F_1 S		0.53	0.36	0.37	0.40	0.48	0.55	0.27	0.50		0.433
C FT	F_1 A	0.44	0.34	0.50		0.53	0.50	-	0.39	0.46		0.451
	F_1 S	0.54	0.65	0.61		0.51	0.49	0.71	0.54	0.56		0.576
C G	F_1 A	0.39	0.31	0.50		0.48	0.49	-	0.46	0.40		0.433
	F_1 S	0.56	0.70	0.64		0.53	0.49	0.79	0.49	0.56		0.595
M FT	F_1 A	0.59	0.41	0.58		0.55	0.59	-	0.68	0.62		0.574
	F_1 S	0.33	0.31	0.51		0.33	0.27	0.37	0.14	0.45		0.339
M G	F_1 A	0.49	0.41	0.51		0.50	0.52	-	0.57	0.49		0.499
	F_1 S	0.49	0.63	0.60		0.52	0.45	0.69	0.37	0.54		0.536
C FT	F_1 A	0.43	0.34	0.51	0.37	0.51		-	0.35	0.41		0.417
	F_1 S	0.56	0.69	0.65	0.65	0.55		0.75	0.55	0.58		0.623
C G	F_1 A	0.42	0.33	0.52	0.35	0.48		-	0.49	0.43		0.431
	F_1 S	0.54	0.68	0.63	0.66	0.53		0.75	0.46	0.55		0.600
M G	F_1 A	0.46	0.34	0.47	0.37	0.48		-	0.45	0.42		0.427
	F_1 S	0.54	0.67	0.62	0.62	0.56		0.76	0.49	0.56		0.603
AE FT	F_1 A	0.58	0.40	0.51	0.60	0.44	0.60	-		0.64		0.539
	F_1 S	0.26	0.26	0.38	0.25	0.27	0.35	0.33		0.36		0.308
AE G	F_1 A	0.57	0.41	0.49	0.60	0.43	0.58	-		0.62		0.529
	F_1 S	0.30	0.31	0.41	0.29	0.36	0.39	0.39		0.38		0.354

A.1.2. Main training dataset: *essay*

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.30	0.38	0.37	0.43	0.42	-	0.59	0.30	0.45	0.405
	F_1 S		0.70	0.58	0.57	0.66	0.55	0.83	0.42	0.54	0.50	0.594
A G	F_1 A		0.28	0.34	0.41	0.34	0.41	-	0.54	0.25	0.41	0.373
	F_1 S		0.73	0.65	0.58	0.66	0.57	0.83	0.50	0.59	0.52	0.626
AE FT	F_1 A		0.41	0.13	0.53	0.15	0.45	-	0.57	0.02	0.25	0.314
	F_1 S		0.55	0.70	0.43	0.74	0.58	0.75	0.47	0.64	0.44	0.589
AE G	F_1 A		0.38	0.11	0.49	0.13	0.41	-	0.54	0.03	0.24	0.291
	F_1 S		0.61	0.71	0.54	0.74	0.60	0.80	0.50	0.65	0.46	0.623
C FT	F_1 A		0.37	0.48	0.51	0.50	0.53	-	0.68	0.51	0.56	0.518
	F_1 S		0.56	0.50	0.49	0.58	0.49	0.67	0.23	0.45	0.43	0.489
C G	F_1 A		0.29	0.39	0.43	0.31	0.42	-	0.45	0.41	0.40	0.388
	F_1 S		0.70	0.66	0.58	0.71	0.56	0.78	0.54	0.53	0.51	0.619
M FT	F_1 A		0.36	0.42	0.45	0.46	0.47	-	0.71	0.50	0.46	0.479
	F_1 S		0.70	0.57	0.59	0.66	0.55	0.73	0.05	0.52	0.47	0.538
M G	F_1 A		0.33	0.36	0.35	0.36	0.38	-	0.67	0.43	0.40	0.410
	F_1 S		0.74	0.66	0.62	0.68	0.58	0.83	0.26	0.52	0.53	0.602

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.27	0.31	0.32		0.36	-	0.42	0.21	0.36	0.321
	F_1 S		0.74	0.65	0.63		0.59	0.89	0.54	0.61	0.54	0.649
A G	F_1 A		0.17	0.32	0.36		0.31	-	0.43	0.25	0.30	0.306
	F_1 S		0.76	0.65	0.65		0.60	0.90	0.55	0.63	0.56	0.663
AE FT	F_1 A		0.40	0.15	0.55		0.50	-	0.60	0.02	0.30	0.360
	F_1 S		0.47	0.68	0.38		0.54	0.63	0.43	0.64	0.42	0.524
AE G	F_1 A		0.43	0.19	0.55		0.53	-	0.63	0.05	0.32	0.386
	F_1 S		0.42	0.69	0.41		0.49	0.59	0.39	0.64	0.40	0.504
C FT	F_1 A		0.43	0.55	0.56		0.58	-	0.70	0.59	0.63	0.577
	F_1 S		0.48	0.45	0.40		0.41	0.57	0.13	0.33	0.36	0.391
C G	F_1 A		0.37	0.42	0.51		0.50	-	0.60	0.45	0.48	0.476
	F_1 S		0.69	0.64	0.53		0.51	0.70	0.44	0.50	0.46	0.559

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.32		0.34	0.34	0.38	-	0.51	0.28	0.38	0.364
	F_1 S		0.73		0.61	0.69	0.58	0.86	0.50	0.57	0.52	0.633
A G	F_1 A		0.28		0.41	0.27	0.35	-	0.46	0.28	0.33	0.340
	F_1 S		0.77		0.61	0.69	0.59	0.89	0.54	0.62	0.55	0.658
AE FT	F_1 A		0.42		0.53	0.18	0.47	-	0.57	0.04	0.27	0.354
	F_1 S		0.54		0.44	0.74	0.57	0.71	0.47	0.65	0.44	0.570
AE G	F_1 A		0.40		0.50	0.15	0.41	-	0.54	0.04	0.24	0.326
	F_1 S		0.61		0.52	0.75	0.60	0.78	0.50	0.65	0.46	0.609
C FT	F_1 A		0.36		0.45	0.47	0.52	-	0.65	0.46	0.52	0.490
	F_1 S		0.64		0.55	0.67	0.53	0.72	0.34	0.48	0.47	0.550
C G	F_1 A		0.31		0.39	0.24	0.37	-	0.44	0.37	0.35	0.353
	F_1 S		0.75		0.62	0.72	0.58	0.83	0.55	0.56	0.53	0.643
M FT	F_1 A		0.46		0.55	0.52	0.59	-	0.71	0.59	0.63	0.579
	F_1 S		0.49		0.42	0.47	0.33	0.47	0.01	0.30	0.26	0.344
M G	F_1 A		0.39		0.44	0.35	0.44	-	0.68	0.38	0.40	0.440
	F_1 S		0.72		0.57	0.67	0.56	0.77	0.23	0.53	0.49	0.568

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.43	0.56		0.53	0.59	-	0.66	0.58	0.63	0.569
	F_1 S		0.47	0.41		0.52	0.36	0.49	0.28	0.29	0.32	0.393
A G	F_1 A		0.42	0.51		0.46	0.55	-	0.62	0.59	0.58	0.533
	F_1 S		0.55	0.49		0.66	0.45	0.61	0.38	0.37	0.42	0.491
AE FT	F_1 A		0.44	0.26		0.33	0.54	-	0.61	0.08	0.35	0.373
	F_1 S		0.35	0.67		0.72	0.48	0.53	0.41	0.64	0.38	0.523
AE G	F_1 A		0.45	0.29		0.26	0.55	-	0.60	0.08	0.35	0.369
	F_1 S		0.38	0.68		0.73	0.47	0.51	0.43	0.64	0.38	0.528
C FT	F_1 A		0.45	0.58		0.52	0.63	-	0.71	0.64	0.68	0.601
	F_1 S		0.24	0.30		0.41	0.20	0.30	0.05	0.18	0.17	0.231
C G	F_1 A		0.45	0.53		0.50	0.61	-	0.69	0.60	0.62	0.571
	F_1 S		0.41	0.52		0.57	0.27	0.36	0.16	0.24	0.25	0.348
M FT	F_1 A		0.44	0.57		0.55	0.63	-	0.71	0.62	0.67	0.599
	F_1 S		0.27	0.23		0.24	0.21	0.31	0.01	0.18	0.15	0.200

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.26	0.28	0.28	0.24	0.33		0.44	0.18	0.32	0.291
	F_1 S		0.77	0.66	0.64	0.70	0.60		0.52	0.62	0.54	0.631
A G	F_1 A		0.28	0.33	0.37	0.27	0.38		0.48	0.34	0.40	0.356
	F_1 S		0.76	0.63	0.63	0.72	0.56		0.53	0.59	0.54	0.620
AE FT	F_1 A		0.45	0.20	0.55	0.19	0.53		0.60	0.03	0.32	0.359
	F_1 S		0.41	0.68	0.33	0.73	0.51		0.43	0.64	0.39	0.515
AE G	F_1 A		0.43	0.24	0.56	0.32	0.56		0.62	0.07	0.35	0.394
	F_1 S		0.37	0.69	0.36	0.73	0.49		0.42	0.65	0.39	0.513
C FT	F_1 A		0.40	0.53	0.56	0.55	0.58		0.70	0.61	0.63	0.570
	F_1 S		0.42	0.43	0.38	0.54	0.40		0.14	0.32	0.35	0.373
C G	F_1 A		0.45	0.56	0.61	0.53	0.64		0.71	0.65	0.68	0.604
	F_1 S		0.30	0.29	0.22	0.44	0.20		0.08	0.13	0.17	0.229

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.42	0.54	0.52	0.49		-	0.66	0.58	0.62	0.547
	F_1 S		0.52	0.49	0.46	0.61		0.56	0.27	0.41	0.36	0.460
A G	F_1 A		0.43	0.53	0.53	0.45		-	0.65	0.60	0.61	0.543
	F_1 S		0.52	0.52	0.46	0.68		0.54	0.31	0.37	0.35	0.469
AE FT	F_1 A		0.45	0.29	0.59	0.31		-	0.64	0.09	0.36	0.390
	F_1 S		0.26	0.64	0.18	0.71		0.44	0.39	0.64	0.37	0.454
AE G	F_1 A		0.41	0.22	0.55	0.08		-	0.56	0.05	0.26	0.304
	F_1 S		0.54	0.71	0.39	0.74		0.68	0.48	0.65	0.45	0.580
C FT	F_1 A		0.43	0.54	0.60	0.52		-	0.70	0.59	0.66	0.577
	F_1 S		0.36	0.36	0.27	0.54		0.40	0.10	0.29	0.26	0.323
C G	F_1 A		0.45	0.51	0.59	0.48		-	0.69	0.60	0.62	0.563
	F_1 S		0.41	0.48	0.29	0.64		0.36	0.15	0.25	0.24	0.353

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A			0.31	0.28	0.27	0.32	-	0.47	0.19	0.30	0.306
	F_1 S			0.67	0.63	0.72	0.61	0.91	0.52	0.61	0.55	0.653
A G	F_1 A			0.33	0.38	0.25	0.33	-	0.47	0.17	0.31	0.320
	F_1 S			0.66	0.62	0.70	0.60	0.89	0.53	0.62	0.55	0.646
AE FT	F_1 A			0.15	0.52	0.26	0.50	-	0.59	0.02	0.30	0.334
	F_1 S			0.69	0.46	0.73	0.55	0.67	0.46	0.64	0.42	0.578
AE G	F_1 A			0.12	0.49	0.18	0.47	-	0.56	0.03	0.27	0.303
	F_1 S			0.71	0.53	0.74	0.56	0.70	0.48	0.64	0.44	0.600
C FT	F_1 A			0.53	0.51	0.51	0.55	-	0.69	0.54	0.59	0.560
	F_1 S			0.52	0.49	0.57	0.47	0.63	0.22	0.40	0.43	0.466
C G	F_1 A			0.39	0.41	0.25	0.39	-	0.47	0.38	0.36	0.379
	F_1 S			0.70	0.61	0.70	0.58	0.83	0.53	0.57	0.53	0.631
M FT	F_1 A			0.57	0.59	0.55	0.63	-	0.71	0.64	0.68	0.624
	F_1 S			0.26	0.30	0.32	0.21	0.31	0.00	0.19	0.14	0.216
M G	F_1 A			0.42	0.40	0.38	0.44	-	0.65	0.48	0.46	0.461
	F_1 S			0.64	0.60	0.64	0.55	0.79	0.36	0.50	0.51	0.574

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.18	0.17	0.24	0.15	0.27	-	0.29		0.24	0.220
	F_1 S		0.79	0.69	0.66	0.73	0.63	0.95	0.59		0.57	0.701
A G	F_1 A		0.25	0.31	0.43	0.30	0.35	-	0.43		0.38	0.350
	F_1 S		0.73	0.64	0.63	0.70	0.58	0.85	0.55		0.54	0.653
AE FT	F_1 A		0.44	0.18	0.55	0.25	0.53	-	0.60		0.31	0.409
	F_1 S		0.43	0.70	0.41	0.73	0.53	0.62	0.44		0.40	0.533
AE G	F_1 A		0.43	0.24	0.54	0.26	0.52	-	0.61		0.33	0.419
	F_1 S		0.46	0.69	0.43	0.73	0.53	0.61	0.42		0.41	0.535
C FT	F_1 A		0.43	0.55	0.58	0.56	0.60	-	0.70		0.65	0.581
	F_1 S		0.44	0.38	0.35	0.46	0.35	0.49	0.10		0.31	0.360
C G	F_1 A		0.35	0.43	0.48	0.31	0.45	-	0.58		0.43	0.433
	F_1 S		0.71	0.68	0.58	0.70	0.54	0.77	0.47		0.50	0.619
M FT	F_1 A		0.45	0.58	0.60	0.54	0.64	-	0.71		0.69	0.601
	F_1 S		0.24	0.16	0.25	0.20	0.17	0.26	0.01		0.11	0.175
M G	F_1 A		0.46	0.58	0.60	0.55	0.65	-	0.71		0.69	0.606
	F_1 S		0.33	0.38	0.33	0.16	0.21	0.25	0.01		0.14	0.226

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.34	0.41	0.37	0.38	0.46	-		0.34	0.49	0.399
	F_1 S		0.73	0.64	0.61	0.75	0.55	0.78		0.58	0.52	0.645
A G	F_1 A		0.34	0.33	0.35	0.15	0.42	-		0.39	0.45	0.347
	F_1 S		0.70	0.63	0.62	0.75	0.56	0.79		0.60	0.53	0.648
AE FT	F_1 A		0.43	0.51	0.61	0.57	0.61	-		0.49	0.59	0.544
	F_1 S		0.21	0.29	0.06	0.19	0.30	0.33		0.50	0.26	0.268
AE G	F_1 A		0.44	0.54	0.61	0.57	0.62	-		0.51	0.61	0.557
	F_1 S		0.20	0.27	0.04	0.13	0.26	0.32		0.45	0.24	0.239
C FT	F_1 A		0.44	0.58	0.59	0.56	0.64	-		0.61	0.70	0.589
	F_1 S		0.27	0.45	0.31	0.51	0.24	0.30		0.35	0.25	0.335
C G	F_1 A		0.37	0.30	0.45	0.20	0.53	-		0.27	0.44	0.366
	F_1 S		0.67	0.66	0.57	0.73	0.51	0.72		0.59	0.48	0.616
M FT	F_1 A		0.45	0.45	0.50	0.37	0.55	-		0.46	0.50	0.469
	F_1 S		0.57	0.56	0.49	0.68	0.47	0.58		0.52	0.38	0.531
M G	F_1 A		0.43	0.36	0.46	0.37	0.54	-		0.33	0.50	0.427
	F_1 S		0.66	0.64	0.59	0.70	0.49	0.64		0.54	0.44	0.588

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.37	0.39	0.39	0.41	0.44	-	0.60	0.37	0.45	0.428
	F_1 S		0.72	0.60	0.59	0.67	0.56	0.82	0.41	0.57	0.50	0.604
A G	F_1 A		0.30	0.42	0.40	0.38	0.43	-	0.53	0.35	0.48	0.411
	F_1 S		0.73	0.61	0.60	0.68	0.55	0.82	0.49	0.56	0.53	0.619
AE FT	F_1 A		0.40	0.12	0.49	0.12	0.42	-	0.56	0.00	0.23	0.293
	F_1 S		0.62	0.70	0.58	0.74	0.60	0.81	0.50	0.65	0.46	0.629
AE G	F_1 A		0.38	0.11	0.52	0.05	0.40	-	0.51	0.01	0.22	0.275
	F_1 S		0.63	0.71	0.52	0.73	0.60	0.82	0.52	0.65	0.47	0.628
C FT	F_1 A		0.29	0.42	0.42	0.45	0.47	-	0.65	0.47	0.49	0.458
	F_1 S		0.67	0.57	0.58	0.66	0.56	0.77	0.34	0.46	0.49	0.567
C G	F_1 A		0.28	0.41	0.41	0.29	0.39	-	0.43	0.41	0.37	0.374
	F_1 S		0.70	0.67	0.60	0.72	0.58	0.81	0.55	0.57	0.51	0.634
M FT	F_1 A		0.42	0.50	0.49	0.48	0.55	-	0.71	0.55	0.57	0.534
	F_1 S		0.58	0.54	0.52	0.61	0.47	0.61	0.04	0.41	0.37	0.461
M G	F_1 A		0.37	0.43	0.43	0.40	0.46	-	0.71	-	0.46	0.466
	F_1 S		0.71	0.64	0.61	0.70	0.55	0.78	0.11	0.78	0.51	0.599

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.37	0.39	0.39	0.41	0.44	-	0.60	0.37	0.45	0.428
	F_1 S		0.72	0.60	0.59	0.67	0.56	0.82	0.41	0.57	0.50	0.604
A G	F_1 A		0.30	0.42	0.40	0.38	0.43	-	0.53	0.35	0.48	0.411
	F_1 S		0.73	0.61	0.60	0.68	0.55	0.82	0.49	0.56	0.53	0.619
AE FT	F_1 A		0.40	0.12	0.49	0.12	0.42	-	0.56	0.00	0.23	0.293
	F_1 S		0.62	0.70	0.58	0.74	0.60	0.81	0.50	0.65	0.46	0.629
AE G	F_1 A		0.38	0.11	0.52	0.05	0.40	-	0.51	0.01	0.22	0.275
	F_1 S		0.63	0.71	0.52	0.73	0.60	0.82	0.52	0.65	0.47	0.628
C FT	F_1 A		0.29	0.42	0.42	0.45	0.47	-	0.65	0.47	0.49	0.458
	F_1 S		0.67	0.57	0.58	0.66	0.56	0.77	0.34	0.46	0.49	0.567
C G	F_1 A		0.28	0.41	0.41	0.29	0.39	-	0.43	0.41	0.37	0.374
	F_1 S		0.70	0.67	0.60	0.72	0.58	0.81	0.55	0.57	0.51	0.634
M FT	F_1 A		0.42	0.50	0.49	0.48	0.55	-	0.71	0.55	0.57	0.534
	F_1 S		0.58	0.54	0.52	0.61	0.47	0.61	0.04	0.41	0.37	0.461
M G	F_1 A		0.37	0.43	0.43	0.40	0.46	-	0.71	-	0.46	0.466
	F_1 S		0.71	0.64	0.61	0.70	0.55	0.78	0.11	0.78	0.51	0.599

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.26		0.30	0.25	0.34	-	0.50	0.24	0.34	0.319
	F_1 S		0.75		0.63	0.71	0.60	0.90	0.49	0.59	0.54	0.651
A G	F_1 A		0.26		0.30	0.31	0.34	-	0.38	0.29	0.36	0.320
	F_1 S		0.76		0.66	0.70	0.60	0.87	0.55	0.60	0.56	0.663
AE FT	F_1 A		0.42		0.53	0.15	0.46	-	0.58	0.04	0.26	0.349
	F_1 S		0.54		0.47	0.73	0.57	0.72	0.46	0.64	0.44	0.571
AE G	F_1 A		0.42		0.54	0.15	0.51	-	0.59	0.03	0.28	0.360
	F_1 S		0.48		0.44	0.73	0.54	0.66	0.46	0.64	0.42	0.546
C FT	F_1 A		0.37		0.50	0.49	0.54	-	0.68	0.51	0.57	0.523
	F_1 S		0.59		0.50	0.59	0.50	0.68	0.24	0.43	0.43	0.495
C G	F_1 A		0.33		0.44	0.29	0.39	-	0.47	0.36	0.35	0.376
	F_1 S		0.73		0.61	0.71	0.58	0.81	0.53	0.58	0.52	0.634
M FT	F_1 A		0.45		0.57	0.52	0.62	-	0.71	0.64	0.67	0.597
	F_1 S		0.41		0.40	0.40	0.30	0.38	0.00	0.23	0.24	0.295
M G	F_1 A		0.41		0.41	0.31	0.44	-	0.66	0.39	0.38	0.429
	F_1 S		0.75		0.60	0.65	0.55	0.79	0.33	0.54	0.49	0.588

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.45	0.55		0.53	0.60	-	0.68	0.61	0.64	0.580
	F_1 S		0.45	0.36		0.47	0.33	0.47	0.21	0.29	0.30	0.360
A G	F_1 A		0.43	0.58		0.52	0.62	-	0.67	0.64	0.67	0.590
	F_1 S		0.38	0.44		0.49	0.28	0.36	0.23	0.19	0.26	0.329
AE FT	F_1 A		0.44	0.28		0.36	0.57	-	0.64	0.08	0.37	0.391
	F_1 S		0.33	0.66		0.72	0.46	0.47	0.37	0.64	0.37	0.503
AE G	F_1 A		0.42	0.17		0.21	0.54	-	0.63	0.03	0.33	0.333
	F_1 S		0.33	0.67		0.71	0.48	0.52	0.41	0.64	0.38	0.518
C G	F_1 A		0.45	0.55		0.51	0.62	-	0.69	0.60	0.63	0.579
	F_1 S		0.42	0.46		0.58	0.27	0.36	0.17	0.28	0.26	0.350
M FT	F_1 A		0.45	0.55		0.52	0.64	-	0.71	0.64	0.67	0.597
	F_1 S		0.30	0.25		0.38	0.22	0.31	0.01	0.18	0.16	0.226

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.28	0.31	0.29	0.32	0.36		0.45	0.22	0.32	0.319
	F_1 S		0.76	0.64	0.64	0.72	0.60		0.53	0.60	0.54	0.629
A G	F_1 A		0.40	0.51	0.52	0.45	0.54		0.60	0.56	0.59	0.521
	F_1 S		0.61	0.57	0.52	0.61	0.45		0.42	0.42	0.43	0.504
AE FT	F_1 A		0.46	0.24	0.57	0.17	0.55		0.63	0.03	0.32	0.371
	F_1 S		0.39	0.69	0.31	0.71	0.51		0.42	0.65	0.39	0.509
AE G	F_1 A		0.42	0.16	0.55	0.12	0.52		0.61	0.04	0.30	0.340
	F_1 S		0.37	0.69	0.33	0.72	0.54		0.45	0.65	0.41	0.520
C G	F_1 A		0.44	0.52	0.56	0.45	0.56		0.65	0.55	0.57	0.538
	F_1 S		0.59	0.58	0.45	0.62	0.43		0.34	0.42	0.38	0.476

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.41	0.54	0.54	0.53		-	0.68	0.62	0.65	0.567
	F_1 S		0.45	0.42	0.40	0.54		0.49	0.21	0.34	0.32	0.396
A G	F_1 A		0.39	0.48	0.49	0.43		-	0.56	0.57	0.57	0.499
	F_1 S		0.61	0.54	0.55	0.70		0.66	0.46	0.43	0.43	0.548
AE FT	F_1 A		0.43	0.22	0.59	0.23		-	0.64	0.09	0.34	0.363
	F_1 S		0.32	0.65	0.20	0.73		0.47	0.40	0.64	0.38	0.474
AE G	F_1 A		0.46	0.29	0.58	0.28		-	0.62	0.08	0.33	0.377
	F_1 S		0.36	0.67	0.24	0.74		0.52	0.43	0.64	0.38	0.498
C FT	F_1 A		0.44	0.54	0.58	0.50		-	0.71	0.60	0.66	0.576
	F_1 S		0.41	0.40	0.30	0.56		0.40	0.06	0.29	0.28	0.338
C G	F_1 A		0.44	0.47	0.56	0.44		-	0.69	0.54	0.57	0.530
	F_1 S		0.51	0.52	0.38	0.66		0.42	0.18	0.35	0.29	0.414
M FT	F_1 A		0.45	0.53	0.58	0.53		-	0.71	0.63	0.67	0.586
	F_1 S		0.34	0.30	0.29	0.51		0.31	0.01	0.22	0.20	0.273

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A			0.38	0.35	0.41	0.41	-	0.51	0.35	0.42	0.404
	F_1 S			0.64	0.60	0.67	0.57	0.83	0.50	0.55	0.52	0.610
A G	F_1 A			0.44	0.40	0.41	0.39	-	0.43	0.41	0.43	0.416
	F_1 S			0.64	0.62	0.67	0.56	0.83	0.54	0.55	0.53	0.618
AE FT	F_1 A			0.16	0.54	0.15	0.50	-	0.59	0.05	0.28	0.324
	F_1 S			0.70	0.42	0.73	0.55	0.67	0.47	0.65	0.42	0.576
C FT	F_1 A			0.55	0.56	0.55	0.59	-	0.70	0.58	0.64	0.596
	F_1 S			0.45	0.42	0.46	0.40	0.54	0.14	0.31	0.36	0.385
C G	F_1 A			0.43	0.46	0.32	0.45	-	0.52	0.45	0.44	0.439
	F_1 S			0.64	0.57	0.69	0.54	0.76	0.49	0.53	0.49	0.589
M G	F_1 A			0.39	0.42	0.30	0.45	-	0.67	0.40	0.40	0.433
	F_1 S			0.66	0.60	0.67	0.55	0.76	0.31	0.53	0.50	0.573

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.22	0.35	0.30	0.28	0.34	-	0.36		0.33	0.311
	F_1 S		0.76	0.69	0.64	0.70	0.61	0.88	0.56		0.54	0.673
A G	F_1 A		0.28	0.34	0.38	0.36	0.35	-	0.36		0.36	0.347
	F_1 S		0.75	0.68	0.65	0.70	0.59	0.87	0.57		0.55	0.670
AE FT	F_1 A		0.43	0.23	0.57	0.32	0.55	-	0.64		0.34	0.440
	F_1 S		0.37	0.69	0.31	0.70	0.51	0.57	0.40		0.39	0.493
AE G	F_1 A		0.42	0.17	0.54	0.17	0.51	-	0.61		0.30	0.389
	F_1 S		0.50	0.69	0.46	0.72	0.54	0.65	0.44		0.41	0.551
C G	F_1 A		0.37	0.44	0.53	0.35	0.50	-	0.59		0.48	0.466
	F_1 S		0.66	0.64	0.51	0.69	0.50	0.70	0.45		0.45	0.575

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.20	0.31	0.35	0.27	0.38	-		0.19	0.42	0.303
	F_1 S		0.71	0.65	0.63	0.73	0.59	0.85		0.61	0.55	0.665
A G	F_1 A		0.33	0.42	0.41	0.14	0.46	-		0.36	0.47	0.370
	F_1 S		0.68	0.63	0.59	0.74	0.54	0.77		0.59	0.51	0.631
AE FT	F_1 A		0.42	0.46	0.61	0.55	0.60	-		0.38	0.55	0.510
	F_1 S		0.21	0.44	0.10	0.35	0.36	0.40		0.57	0.30	0.341
AE G	F_1 A		0.44	0.54	0.60	0.56	0.60	-		0.45	0.57	0.537
	F_1 S		0.23	0.38	0.12	0.23	0.35	0.38		0.52	0.29	0.313
C FT	F_1 A		0.38	0.24	0.41	0.18	0.47	-		0.26	0.44	0.340
	F_1 S		0.70	0.69	0.61	0.75	0.57	0.76		0.62	0.52	0.653
C G	F_1 A		0.43	0.40	0.54	0.36	0.57	-		0.41	0.54	0.464
	F_1 S		0.60	0.65	0.50	0.69	0.41	0.57		0.55	0.43	0.550
M FT	F_1 A		0.42	0.51	0.51	0.44	0.58	-		0.56	0.58	0.514
	F_1 S		0.52	0.47	0.52	0.48	0.44	0.58		0.41	0.36	0.473
M G	F_1 A		0.42	0.48	0.50	0.36	0.58	-		0.46	0.57	0.481
	F_1 S		0.60	0.62	0.57	0.69	0.44	0.59		0.46	0.43	0.550

A.1.3. Main training dataset: *ibm*

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A G	F_1 A	0.41	0.42	0.43		0.47	0.53	-	0.53	0.62	0.55	0.495
	F_1 S	0.55	0.55	0.56		0.56	0.46	0.48	0.49	0.38	0.40	0.492
A FT	F_1 A	0.54	0.43	0.54			0.58	-	0.62	0.64	0.68	0.576
	F_1 S	0.45	0.42	0.42			0.38	0.38	0.34	0.24	0.36	0.374
A FT	F_1 A	0.61	0.44	0.59		0.56	0.62		0.65	0.67	0.70	0.605
	F_1 S	0.34	0.31	0.38		0.11	0.24		0.31	0.22	0.26	0.271
A G	F_1 A	0.58	0.41	0.59		0.55	0.63		0.65	0.69	0.70	0.600
	F_1 S	0.34	0.25	0.30		0.08	0.20		0.34	0.08	0.24	0.229
A FT	F_1 A	0.62	0.43	0.60		0.56		-	0.68	0.68	0.70	0.610
	F_1 S	0.35	0.27	0.30		0.10		0.27	0.22	0.09	0.25	0.231
A G	F_1 A	0.61	0.44	0.61		0.55		-	0.66	0.68	0.67	0.603
	F_1 S	0.29	0.25	0.33		0.14		0.20	0.30	0.09	0.27	0.234
C G	F_1 A	0.43	0.30	0.45		0.41		-	0.28	0.52	0.52	0.416
	F_1 S	0.60	0.69	0.65		0.54		0.72	0.59	0.46	0.55	0.600
A FT	F_1 A	0.61	0.42	0.60		0.54	0.62	-	0.69		0.70	0.597
	F_1 S	0.24	0.26	0.30		0.10	0.21	0.25	0.22		0.21	0.224
A G	F_1 A	0.55	0.40	0.56		0.53	0.61	-		0.65	0.69	0.570
	F_1 S	0.49	0.47	0.50		0.25	0.33	0.41		0.27	0.37	0.386
C FT	F_1 A	0.64	0.46	0.56		0.55	0.64	-		0.58	0.71	0.591
	F_1 S	0.31	0.29	0.40		0.12	0.07	0.27		0.31	0.18	0.244
C G	F_1 A	0.61	0.46	0.54		0.55	0.65	-		0.50	0.69	0.571
	F_1 S	0.41	0.37	0.47		0.11	0.10	0.31		0.42	0.20	0.299

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.37	0.50		0.49	0.48	-	0.63	0.46	0.49	0.489
	F_1 S		0.64	0.61		0.58	0.50	0.74	0.36	0.54	0.45	0.553
A G	F_1 A		0.37	0.58		0.53	0.53	-	0.61	0.59	0.55	0.537
	F_1 S		0.61	0.60		0.42	0.50	0.72	0.43	0.38	0.47	0.516
AE FT	F_1 A		0.42	0.14		0.21	0.47	-	0.63	0.01	0.27	0.307
	F_1 S		0.58	0.68		0.71	0.58	0.73	0.42	0.65	0.42	0.596
AE G	F_1 A		0.40	0.13		0.24	0.45	-	0.62	0.02	0.27	0.304
	F_1 S		0.57	0.68		0.72	0.57	0.70	0.41	0.65	0.42	0.590
C FT	F_1 A		0.40	0.50		0.52	0.58	-	0.62	0.60	0.61	0.547
	F_1 S		0.47	0.48		0.46	0.37	0.53	0.40	0.31	0.30	0.415
C G	F_1 A		0.40	0.38		0.46	0.53	-	0.56	0.41	0.47	0.459
	F_1 S		0.63	0.66		0.68	0.50	0.67	0.49	0.52	0.42	0.571
MG	F_1 A		0.44	0.47		0.50	0.54	-	0.67	0.53	0.52	0.524
	F_1 S		0.61	0.60		0.48	0.47	0.63	0.30	0.45	0.37	0.489

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A G	F_1 A	0.45	0.41	0.47		0.51	0.54	-	0.55	0.64	0.58	0.519
	F_1 S	0.53	0.50	0.53		0.51	0.43	0.42	0.48	0.31	0.38	0.454
A FT	F_1 A	0.53	0.43	0.54			0.59	-	0.62	0.63	0.68	0.574
	F_1 S	0.46	0.44	0.49			0.39	0.39	0.35	0.28	0.37	0.396
A FT	F_1 A	0.50	0.38	0.51		0.51	0.54		0.55	0.59	0.59	0.521
	F_1 S	0.51	0.56	0.59		0.43	0.46		0.46	0.38	0.43	0.478
A G	F_1 A	0.58	0.41	0.62		0.55	0.63		0.64	0.68	0.69	0.600
	F_1 S	0.39	0.31	0.40		0.10	0.26		0.38	0.08	0.28	0.275
A G	F_1 A	0.59	0.44	0.59		0.55		-	0.63	0.68	0.67	0.593
	F_1 S	0.34	0.36	0.33		0.19		0.26	0.37	0.11	0.35	0.289
A G	F_1 A	0.57	0.39	0.55		0.51	0.61	-		0.55	0.63	0.544
	F_1 S	0.48	0.49	0.58		0.38	0.36	0.44		0.33	0.36	0.428
C FT	F_1 A	0.63	0.45	0.57		0.53	0.64	-		0.58	0.69	0.584
	F_1 S	0.40	0.38	0.49		0.14	0.14	0.37		0.37	0.26	0.319
M G	F_1 A	0.60	0.44	0.62		0.56	0.64	-		0.61	0.69	0.594
	F_1 S	0.48	0.31	0.50		0.07	0.12	0.20		0.32	0.23	0.279

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.41	0.55		0.51	0.56	-	0.66	0.56	0.62	0.553
	F_1 S		0.55	0.50		0.40	0.37	0.57	0.30	0.42	0.37	0.435
A G	F_1 A		0.36	0.54		0.50	0.51	-	0.59	0.59	0.55	0.520
	F_1 S		0.67	0.63		0.49	0.51	0.74	0.47	0.41	0.49	0.551
AE FT	F_1 A		0.42	0.12		0.28	0.44	-	0.64	0.01	0.27	0.311
	F_1 S		0.56	0.67		0.70	0.57	0.72	0.40	0.65	0.42	0.586
AE G	F_1 A		0.43	0.21		0.37	0.50	-	0.65	0.03	0.30	0.356
	F_1 S		0.49	0.66		0.69	0.54	0.62	0.39	0.64	0.40	0.554
C FT	F_1 A		0.43	0.57		0.56	0.63	-	0.70	0.67	0.68	0.606
	F_1 S		0.43	0.32		0.21	0.19	0.34	0.15	0.16	0.13	0.241
C G	F_1 A		0.45	0.53		0.53	0.61	-	0.68	0.52	0.61	0.561
	F_1 S		0.45	0.57		0.45	0.29	0.46	0.25	0.37	0.27	0.389
M FT	F_1 A		0.38	0.52		0.52	0.50	-	0.55	0.51	0.49	0.496
	F_1 S		0.63	0.62		0.45	0.51	0.69	0.49	0.48	0.43	0.538
MG	F_1 A		0.47	0.60		0.54	0.62	-	0.70	0.61	0.63	0.596
	F_1 S		0.41	0.46		0.31	0.35	0.44	0.11	0.27	0.23	0.323

A.1.4. Main training dataset: *web*

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A	0.63	0.42	0.56	0.58			-	0.51	0.63	0.64	0.567
	F_1 S	0.30	0.32	0.26	0.16			0.28	0.42	0.22	0.28	0.280
A G	F_1 A	0.62	0.43	0.55	0.59			-	0.67	0.67	0.63	0.594
	F_1 S	0.31	0.36	0.31	0.25			0.30	0.19	0.07	0.31	0.263
A G	F_1 A	0.61	0.40		0.56	0.49		-	0.47	0.63	0.62	0.540
	F_1 S	0.35	0.36		0.33	0.46		0.29	0.49	0.26	0.34	0.360
A G	F_1 A	0.61	0.41	0.60		0.48		-	0.66	0.66	0.61	0.576
	F_1 S	0.32	0.28	0.27		0.41		0.29	0.26	0.12	0.35	0.288
A FT	F_1 A	0.49	0.36	0.43	0.48	0.26			0.56	0.38	0.36	0.415
	F_1 S	0.53	0.63	0.49	0.43	0.72			0.36	0.58	0.50	0.530
A G	F_1 A	0.56	0.43	0.49	0.51	0.42			0.51	0.61	0.52	0.506
	F_1 S	0.45	0.47	0.45	0.45	0.61			0.44	0.47	0.46	0.475
C G	F_1 A	0.63	0.42	0.60	0.58	0.46			0.63	0.65	0.59	0.570
	F_1 S	0.27	0.29	0.28	0.24	0.45			0.35	0.25	0.30	0.304
A G	F_1 A	0.61	0.42	0.57	0.59	0.43		-	0.67		0.59	0.554
	F_1 S	0.32	0.38	0.28	0.28	0.46		0.36	0.17		0.35	0.325
A FT	F_1 A	0.61	0.41	0.57	0.56	0.49		-		0.63	0.63	0.557
	F_1 S	0.38	0.37	0.30	0.27	0.52		0.34		0.34	0.35	0.359
A G	F_1 A	0.64	0.42	0.60	0.59	0.50		-		0.65	0.65	0.579
	F_1 S	0.27	0.30	0.27	0.26	0.26		0.22		0.27	0.32	0.271
C G	F_1 A	0.63	0.45	0.58	0.60	0.55		-		0.57	0.68	0.580
	F_1 S	0.36	0.38	0.33	0.22	0.20		0.20		0.38	0.23	0.288

SYN	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.40	0.49	0.45	0.44		-	0.59	0.36	0.55	0.469
	F_1 S		0.70	0.59	0.58	0.67		0.73	0.42	0.53	0.49	0.589
A G	F_1 A		0.36	0.48	0.43	0.39		-	0.52	0.45	0.51	0.449
	F_1 S		0.75	0.66	0.62	0.68		0.79	0.52	0.56	0.54	0.640
AE FT	F_1 A		0.46	0.31	0.57	0.44		-	0.61	0.11	0.41	0.416
	F_1 S		0.39	0.63	0.26	0.66		0.55	0.44	0.64	0.39	0.495
AE G	F_1 A		0.45	0.34	0.56	0.46		-	0.62	0.12	0.42	0.424
	F_1 S		0.38	0.64	0.28	0.65		0.51	0.41	0.62	0.37	0.483
C FT	F_1 A		0.42	0.55	0.59	0.47		-	0.69	0.61	0.67	0.571
	F_1 S		0.41	0.38	0.22	0.60		0.38	0.18	0.27	0.24	0.335
C G	F_1 A		0.31	0.38	0.47	0.21		-	0.38	0.24	0.45	0.349
	F_1 S		0.74	0.70	0.55	0.73		0.74	0.57	0.57	0.50	0.638
M FT	F_1 A		0.40	0.49	0.46	0.46		-	0.70	0.43	0.52	0.494
	F_1 S		0.64	0.59	0.55	0.72		0.70	0.13	0.48	0.46	0.534
MG	F_1 A		0.45	0.60	0.60	0.51		-	0.71	0.61	0.65	0.590
	F_1 S		0.29	0.24	0.24	0.55		0.28	0.04	0.30	0.20	0.268

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A G	F_1 A	0.61	0.44	0.54	0.58			-	0.65	0.68	0.61	0.587
	F_1 S	0.31	0.36	0.34	0.27			0.25	0.27	0.15	0.32	0.284
A G	F_1 A	0.64	0.43	0.61		0.52		-	0.65	0.65	0.64	0.591
	F_1 S	0.28	0.26	0.30		0.31		0.34	0.33	0.12	0.36	0.288
A FT	F_1 A	0.49	0.34	0.42	0.49	0.31			0.51	0.41	0.37	0.418
	F_1 S	0.51	0.61	0.51	0.44	0.69			0.42	0.54	0.49	0.526
A G	F_1 A	0.53	0.37	0.41	0.48	0.29			0.47	0.46	0.44	0.431
	F_1 S	0.53	0.53	0.55	0.46	0.63			0.49	0.60	0.50	0.536
C G	F_1 A	0.65	0.44	0.60	0.57	0.47			0.65	0.61	0.58	0.571
	F_1 S	0.28	0.26	0.29	0.21	0.41			0.40	0.25	0.29	0.299
A FT	F_1 A	0.64	0.44	0.57	0.59	0.53		-	0.55		0.66	0.569
	F_1 S	0.24	0.32	0.17	0.17	0.30		0.23	0.42		0.28	0.266
A G	F_1 A	0.63	0.44	0.55	0.59	0.46		-	0.66		0.62	0.564
	F_1 S	0.29	0.35	0.25	0.25	0.43		0.21	0.23		0.30	0.289

ALL	dataset	essay	micro	db	ibm	com	web	cdcp	ukp	nk	aif	Avg
A FT	F_1 A		0.40	0.43	0.43	0.37		-	0.55	0.26	0.50	0.420
	F_1 S		0.72	0.64	0.58	0.72		0.77	0.47	0.59	0.51	0.625
A G	F_1 A		0.43	0.54	0.49	0.46		-	0.59	0.63	0.63	0.539
	F_1 S		0.68	0.55	0.57	0.56		0.65	0.46	0.38	0.47	0.540
AE FT	F_1 A		0.44	0.39	0.56	0.49		-	0.63	0.21	0.47	0.456
	F_1 S		0.38	0.59	0.31	0.57		0.52	0.41	0.62	0.36	0.470
AE G	F_1 A		0.46	0.43	0.57	0.44		-	0.64	0.25	0.50	0.470
	F_1 S		0.29	0.63	0.23	0.60		0.42	0.43	0.59	0.33	0.440
C FT	F_1 A		0.40	0.51	0.54	0.47		-	0.68	0.51	0.62	0.533
	F_1 S		0.56	0.50	0.42	0.68		0.59	0.27	0.41	0.42	0.481
C G	F_1 A		0.44	0.47	0.54	0.43		-	0.61	0.48	0.59	0.509
	F_1 S		0.60	0.59	0.44	0.63		0.48	0.45	0.46	0.37	0.503
M FT	F_1 A		0.43	0.53	0.59	0.53		-	0.71	0.57	0.64	0.571
	F_1 S		0.37	0.35	0.39	0.60		0.40	0.01	0.29	0.26	0.334
MG	F_1 A		0.46	0.48	0.56	0.39		-	0.70	0.44	0.57	0.514
	F_1 S		0.52	0.53	0.44	0.62		0.50	0.14	0.54	0.37	0.458

B. 3 class Relation-based Argument Mining

Algorithm	Dataset	Average F_1 CV	standard deviation
GRU	*web*	43.32	0.3
GRU	*web*, *com*, *ukp*, *nk*	47.84	0.4
Attention GRU	*web*	43.24	0.2
Attention GRU	*web*, *com*, *ukp*, *nk*	48.9	0.8

Table B.1.: Results using 5x5 fold CV.