UNIVERSITY OF LIVERPOOL

Aggregating and Analysing Opinions for Argument-based Relations

Thesis submitted in accordance with the requirements of
the University of Liverpool for the degree of Doctor in Philosophy by

Pavithra Rajendran

June 2019

Contents

Illustrations

List of Figures

List of Tables

Preface

This thesis is primarily my own work. The sources of other materials are identifed.

Abstract

Computational argumentation is a widely studied research area that has developed many formal models of argumentation. Argumentation is itself a multi-disciplinary field that branches from several research domains such as pragmatics, philosophy and logic. The idea of automatically extracting arguments and their relations from social media texts gives rise to a new research domain known as argument mining. The advancement in the field of natural language processing and machine learning has been helpful for extracting argument structures from informal texts. However, work on argument mining itself suffers from several drawbacks with one main problem related to lack of annotated corpora that can deal with argumentative texts. Another main drawback that can be related is the heterogeneous nature of data which prevents the use of a generalised annotation corpus for identifying argument structures.

This thesis studies the intersection of computational argumentation and natural language processing to understand and process natural language arguments by developing an argumentation process which consists of identifying arguments and their relations and evaluating these arguments. Arguments are identified as structured or abstract arguments using linguistic attributes such as sentiment, stance and topic. The linguistic expression of the content towards a particular topic as a stance gives a pattern among opinions leading to two types of opinions - explicit opinions and implicit opinions. A binary classification approach is proposed for automatically classifying opinions as explicit or implicit opinions based on the way the stance is expressed. A set of hotel reviews is selected and the opinions are annotated by human annotators. The dataset is developed further by using different semi-supervised and weakly supervised approaches that automatically labels a large unlabeled dataset. This automatically labelled dataset is evaluated for deep learning models with the best performance of an LSTM model on the annotated dataset giving an accuracy of 84%. The second step of the argumentation process uses this classification of opinions to identify different types of relations that occur among these opinions such that it leads to constructing argument structures supporting a particular conclusion. Linguistic attributes such as sentiment and topic, along with the stance classification and domain-based knowledge are used for proposing a distant-supervision based approach that relates opinions as premises leading to a conclusion. The relation among the premises is similar to the entailment relation present in textual entailment and hence it is termed “support-based entailment” relation. Another relation that is identified is the rephrase relation, in which, two opinions have similar argument meaning and wherein one can replace another without changing the meaning. These relations are useful for constructing argument structures as well as to identify enthymemes from arguments where an enthymeme is an argument with certain information missing.

The different steps of the argumentation process for processing arguments in opinionated texts are carried out by considering opinions as abstract arguments. These arguments are built into bipolar argumentation graphs where a set of arguments are related to the support and attack relation. Different existing computational argumentation methods to compute the strength of these arguments are investigated in combination with natural language processing methods. The support relation in these graphs is used to convert them into coalitions of arguments, where a set of arguments support each other directly or indirectly and no attack relation exists within the coalition. These arguments are evaluated by investigating different ways of choosing coalitions, computing their strength and using them to support arguments as a whole. The evaluation process of these arguments is empirically evaluated for an NLP based task, which is to predict the overall sentiment of a review.

The thesis explores a series of steps to identify and understand arguments present in opinionated texts by considering how natural language arguments fit within the argumentation process. It is shown that there exist different types of relations among these arguments if they are considered as structured arguments and that such relations help in identifying enthymemes from arguments. These relations are similar to existing relations in natural language processing but the latter has several drawbacks as they were not designed to detect argument-based relations. It is also shown that existing computational argumentation frameworks that are not developed for natural language arguments can be adopted for real-world tasks.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisors, Prof. Danushka Bollegala and Prof. Simon Parsons for offering this PhD opportunity. I would also like to thank the Department of Computer Science for the scholarship to pursue my doctoral studies. It was challenging when I started my studies with my personal circumstances and would not have been possible without the tremendous support and guidance from my supervisors, despite their busy schedule they were always supportive and flexible. I thank them for their valuable time and expertise which helped in gaining knowledge and training to pursue this research.

I thank Prof. Simon Parsons for the opportunity to learn about computational argumentation and for his continuous guidance as a supervisor after moving from Liverpool. I also thank Prof. Danushka Bollegala for becoming my first supervisor and for the opportunity to learn about natural language processing. Simon and Danushka have always encouraged and motivated me at every step and continuously reviewed my work throughout these four years. Their advice and mentorship to help me understand how to carry out research and write publications as well as writing this thesis are beyond expectations. I'm very fortunate to work with my supervisors who always cared about me both personally and academically and were open for any questions and provide feedback on my work.

A special thanks to my advisors, Prof. Katie Atkinson and Prof. Frans Coenen for all their valuable feedback.

I thank my parents and my brother for their support and love. My gratitude to my parents for helping me pursue my Masters degree without which this would not have been possible. I am indebted to my loving mother for her unconditional support and I am greatly overwhelmed by her support and guidance.

Finally, my dearest husband Karthik who encouraged me in pursuing my career after marriage. I am deeply indebted to him for his unconditional support right from quitting his job and coming to Liverpool, helping me in taking care of our little one especially when I had to attend conferences, encouraging me during tough times and for his understanding, patience and tolerance during the four years of my PhD.

I dedicate my thesis to my loving and caring daughter Navya, for her love and kindness has emotionally strengthened me day by day.

Chapter 1

Introduction

In everyday life, we find that humans argue a lot and by doing so develop their communication skills and knowledge. Mercier and Sperber [3] argues that argumentation is fundamental to human reasoning. The art of reasoning is a means of developing our intellectual capability and for making better decisions. Since ancient times, the idea of representing arguments in everyday reasoning has been studied widely by philosophers and rhetoric theorists, which further developed into informal logic studied by Johnson and Blair [4], Fogelin [5], Walton [6] and several others. The idea of making machines to acquire this argumentative behaviour has been studied as a branch of Artificial Intelligence (AI), also known as computational argumentation, with its roots in philosophy and logic and, in computer science has been mainly developed for logic-based reasoning with significant earlier work such as Simari and Loui [7], Pollock [8] and Dung [9]. It has been adapted for decision-making [10], handling uncertainty in a knowledge base [11], non-monotonic reasoning [12] and multi-agent systems [13] but not limited to these. Amgoud et al. [14] describe argumentation as a process that is carried out in different steps for making a decision and drawing inferential conclusions. The following steps describe the process.

1. Identifying or constructing arguments in favour or against a decision.

2. Identifying the relations among the arguments

3. Compute the strength of the arguments

4. Evaluate the acceptability of the arguments

5. Comparing the decisions based on the accepted arguments.

However, studies in computational argumentation have successfully modelled formal models of reasoning that are not specifically constructed to work on natural language texts. A recent work [15] discusses why a traditional logical based argumentation process fails to represent arguments that are present in natural language texts and dialogues. The authors argue that there exist a few drawbacks in using abstract argumentation and logical argumentation for representing natural language arguments and hence propose a better formalism, in which, an argument is represented as a formula in the form $\mathcal{R}(y) : (-)\mathcal{C}(x)$ where $\mathcal{R}()$ is a function representing the reasoning as premises and $\mathcal{C}()$ is a function representing the conclusion. The difference between the definitions of an argument in existing literature and the work proposed by Amgoud et al. [15] is that, Amgoud et al. [15] represents an argument as: (1) an argument without any notion of derivation, (2) an argument can be an enthymeme, (3) an argument can contain reasons that are hypothetical and (4) an argument can have non-deductive links between the reasoning and conclusion.

The above work is an example of how the idea of evaluating natural language content has attracted the attention of theoretical argumentation researchers. This is mainly because of the emergence of a new research area known as argument mining [16–18], that aims to integrate the theoretical aspects of argumentation theory with natural language processing techniques from computational linguistics, for mining arguments and the relations between them present within natural language texts. In earlier work related to argument mining, the investigation has focused on legal content [19, 20], scientific articles [21] and other formal content. The gap between traditional argumentation and argumentation for natural language texts is bridged by making use of natural language techniques.

Research in the field of natural language processing aims to understand the context and structure of a language and helps in extracting useful linguistic information from natural language texts. Different topics are explored in this domain such as text parsing, semantic representations, machine translation etc. The advent of internet technologies and the vast amount of social media data available over the internet has broadened the field of natural language processing with recent work in argument mining attempting to discover the underlying arguments that are present in unstructured data available over the internet. Lippi and Torroni present a detailed survey of the existing work [22] in argument mining and the kind of machine learning techniques that have been used so far [23]. In particular, they explain that existing work in argument mining follows a pipeline of steps for identifying arguments and their relations among natural language texts and these pipeline of steps is termed as the argument mining pipeline and is as follows:

1. Identifying whether a sentence is an argumentative or not.
2. Identifying the different components of an argument.
3. Argument structure prediction that relates the different components of an argument.

This thesis addresses research questions that contribute to the argumentation community by understanding how arguments occur in natural language texts, in particular opinions, and how these arguments are related. Within the natural language processing community, opinion mining has been widely studied in which different techniques are explored for identifying texts as opinions and whether an opinion is positive, negative or subjective. This thesis addresses research questions that contribute to the natural language community, in particular, opinion mining, by understanding the reasoning behind opinions and how this can help in the process of decision making.

A reason behind this work is the emergence of online e-commerce markets which are becoming progressively competitive. These rely on online reviews that are comprised largely of opinionated texts, which have become an important factor for increasing the market sales and trust among the customers [24]. People decide to purchase a product or service based on the reviews posted and this decision is influenced by several factors, such as (1) the impact of positive and negative reviews and (2) the timeline in which the negative reviews have been posted.

Online reviews are accompanied by an overall star rating that gives an overall sentiment of whether a reviewer likes the product/service or not. These overall ratings are usually in the range of 1 to 5 where a rating of 1 or 2 are rated for negative reviews and a rating of 4 or 5 are rated for positive reviews. Reviews with a rating of 3 are usually neutral reviews. Although these ratings resemble the overall conclusion of reviews, it does not provide any further reasoning or justification as to why the decision to award the number of stars was made. In this work, I address the question of how to extract the justification for the overall star rating from the text of the review. I focus on two kinds of overall summary of a review: (1) the reviewer likes the product/service and (2) the reviewer does not like the product/service and I examine whether an argumentative analysis can assist in identifying the strength and weakness of the decision.

In natural language processing, sentiment analysis and opinion mining have been an ongoing research area for many years [25] and is still a topic of interest. The Oxford dictionary [1] defines an opinion as "*a view or judgement formed about something, not necessarily based on fact or knowledge*". These opinions are better understood by extracting linguistic attributes such as (1) *sentiment* (2) *topic* and (3) *stance* which have been useful for sentiment analysis tasks. Sentiment analysis identifies the opinion based on the polarity expressed in the context as positive, negative or objective [25]. Online reviews contain opinions in which these polarities are also expressed in context with a topic, also known as aspect-based sentiment analysis [26]. Another growing research topic in NLP is the identification of stance which refers to whether a speaker is for or against a topic. Opinions are a different kind of text, in which aspect-based sentiment analysis is similar to the notion of stance. But, according to linguistics, stance [27] has a different meaning where it is an "expression of judgement, attitude towards a topic in the context". This kind of definition is relevant to opinions present in online reviews and is followed in this work. An example to illustrate these attributes is given below.

[1] https://www.oxforddictionaries.com/

Review 1

> *Overall star rating: 1.0*
>
> *Made to feel unwelcome! hotel room itself was beautiful and very clean, however, this is probably the worst service we have ever experienced!*

In the above example, the opinion "*hotel room itself was beautiful and very clean, however, this is probably the worst service we have ever experienced!*" has a negative sentiment and talks about the topics hotel, room and service. This kind of opinion is argumentative because it provides a justification for the opinion.

By identifying similar opinions in other reviews with the same overall star rating and relating them can help in reasoning why a certain conclusion is supported by a set of reviews. These opinions are also structured differently based on the stance expressed. For example, an opinion " *I do not recommend the hotel*" explicitly states the dislike of a reviewer and so I consider it to be an explicit opinion whereas an opinion "*the hotel is a bit old and not clean*" implicitly expresses the dislike of a reviewer by expressing negative attitude about the hotel's appearance but does not explicitly express the fact that the reviewer does not recommend it.

The linguistic properties explained above give an opportunity to explore different types of argument structures. Computational argumentation deals with arguments in two main ways: abstract argumentation and structured argumentation. The outcome of this thesis provides the argument mining community with a novel argument mining pipeline that explores arguments as both abstract and structured, studies computation of the strength of the arguments and investigates the adaptability of existing computational methods for a natural language processing task relevant to reviews.

Abstract arguments do not have any internal structural representation whereas structured argumentation deals with the internal structure of texts such that, a simple structured argument consists of a set of premises leading to a conclusion. A simple premise-conclusion model consists of a premise or reasoning that can inferentially support a conclusion. A premise-conclusion [28] model cannot be adapted directly for natural language texts as it may not contain logical inferences for connecting the premises and conclusion. Freeman's [29] different types of arguments structures are constructed using several premises for a given conclusion without any logical inference between the premises and conclusion. According to Freeman, a serial argument structure is a series of premises in which one supports the other in a linked fashion and together supporting a conclusion and a linked argument structure is a set of independent premises that as a group support a conclusion. In this thesis, the different types of opinionated texts present in a set of reviews is studied for identifying Freeman-based argument structures.

Many abstract models of reasoning have been developed in computational argumentation, that is based on the initial work by Dung known as the abstract argumentation framework. In Dung's [9] framework, a given set of arguments are related using the "*attack*" relation. The notion of support as an independent relation is introduced in an extension of the abstract argumentation framework known as bipolar argumentation framework [14], which relates arguments using both attack and support relation. This type of framework is further converted into meta-arguments known as coalitions of arguments [14], in which arguments supporting each other directly or indirectly are grouped together. These existing abstract argument frameworks help in understanding the relation between different arguments. Hence, in this thesis, the Freeman-based argument structures are converted into abstract bipolar argumentation graphs and computation of the strength of the arguments is studied. Further, because we find a lot of similar arguments about a topic present in reviews, the bipolar argumentation graphs are converted into coalitions of arguments. This gives us an opportunity to study how the strength of the coalitions can be computed. In this work, different ways of computing the strength are proposed which is then used to evaluate the overall sentiment of a review. In doing this, the usefulness of traditional abstract argumentation for the overall sentiment prediction task is studied.

1.1 Thesis structure

The structure of the thesis is based on exploring different steps of the proposed argument mining pipeline which is given below:

INPUT A set of reviews for some product/service.

1. Identifying opinions as argumentative based on their linguistic properties: sentiment, stance and topic.
2. Identifying explicit and implicit opinions based on how stance is expressed in the opinions.
3. Identifying relations among arguments for constructing Freeman-style [29] serial and linked argument structures in favour/against a decision.
4. Computing the strength of the opinions by identifying attack and support relation among opinions as arguments.
5. Aggregating opinions as coalitions of arguments and assessing their strength for the overall sentiment prediction task.

OUTPUT This gives the overall opinion of a set of reviews for the product/service.

The chapters are as follows:

- Chapter 2 is a literature review that covers the related work in argumentation, argument mining and natural language processing.
- Chapter 3 gives a brief description of the different steps of the argument mining pipeline, why these steps are considered and how each step is connected to each other.

- Chapters 4, 5 and 6 discuss the different steps of the argument mining pipeline and the different research questions that are answered in these chapters are discussed in Chapter 3.

- Chapter 7 discusses the conclusion of the thesis, open issues present in the current work and on the scope for future work.

1.2 List of publications

Following is the list of publications related to this thesis:

1. Pavithra Rajendran, Danushka Bollegala and Simon Parsons, Contextual stance classification of opinions: A step towards enthymeme reconstruction in online reviews,3rd Workshop on Argument Mining at Annual Conference of the Association for Computational Linguistics (ACL-16), p.32–39 2016. This paper is related to Chapter 4 and discusses the initial study on the classification of implicit and explicit opinions.

2. Pavithra Rajendran, Danushka Bollegala and Simon Parsons, Identifying argument based relation properties in opinions, 15th International Conference of the Pacific Association for Computational Linguistics (PACLING'17), 2017. This paper is related to Chapter 4 and presents a detailed study on the classification of implicit and explicit opinions and how the opinions are related to arguments and enthymemes.

3. Pavithra Rajendran, Danushka Bollegala and Simon Parsons, Is Something Better than Nothing? Automatically Predicting Stance-based Arguments Using Deep Learning and Small Labelled Dataset, 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), p. 28–34, 2018. This paper is related to Chapter 4 and explores the creation of large datasets of implicit and explicit opinions for using deep learning techniques.

4. Pavithra Rajendran, Danushka Bollegala and Simon Parsons, Sentiment-Stance-Specificity (SSS) Dataset: Identifying support-based entailment among opinions, 11th Language Resources and Evaluation Conference (LREC'18), p.619–626, 2018. This paper is related to Chapter 5 and discusses the different relations present among implicit and explicit opinions and how the relations help in identifying Freeman-based argument structures.

5. Pavithra Rajendran, Danushka Bollegala and Simon Parsons, Identifying rephrase relation among opinions (to be submitted). This paper is related to Chapter 5 and discusses about automatically identifying the rephrase relation present among a set of implicit and explicit opinions.

6. Pavithra Rajendran, Danushka Bollegala and Simon Parsons, Assessing weight of opinion by aggregating coalitions of arguments,6th International Conference on Computational Models of Argument (COMMA-16), p.43–438 2016. This paper is related to Chapter 6 and discusses on an initial work carried out by using abstract argumentation techniques for converting Freeman-based argument structures into coalitions of arguments and the coalitions are assessed for overall sentiment prediction task of reviews.

7. Pavithra Rajendran, Danushka Bollegala and Simon Parsons, Aggregating coalitions to weigh collective opinions (to be submitted to Argumentation & Computation journal). This paper is related to Chapter 6 and discusses on an extended work carried out on coalitions of arguments.

Chapter 2

Literature Review

The argument mining pipeline proposed in this work makes use of concepts from computational argumentation and natural language processing techniques. In this chapter, I discuss relevant literature work present in computational argumentation and natural language processing and also provide a brief overview of the different existing work carried out in argument mining.

2.1 Computational argumentation

Argumentation has become an area of increasing study in artificial intelligence. Drawing on work from philosophy, which attempts to provide a realistic account of human reasoning [1, 28, 30], researchers in artificial intelligence have developed computational models of this form of reasoning. Over the decades, a lot of work has been developed on formal logic and mathematical models with notable work by Simari and Loui [7], Pollock [8] and Dung [9]. The emergence of non-monotonic reasoning that addresses uncertainty and incomplete information has led to two types of argumentation models: abstract and structured. Dung [9] introduced the Abstract argumentation Framework (AF) which is also modelled as a graphical representation, in which arguments are considered as abstract atomic units that are related by an attack relation. Dung's AF has been extended into different frameworks and one of the reasons for the success of the abstract argument framework is its representation of arguments as simple elements that do not have any internal structure. This is an advantage since we need not have to worry about the intrinsic meaning of an argument.

Definition 2.1 (**Abstract argumentation Framework (AF)**)**.** An abstract argumentation framework is a tuple $(\mathcal{A}, \mathcal{R})$ where $\mathcal{A}$ represents the set of arguments $\mathcal{A} = \{a_1, ...a_m\}$ and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ represents the attack relation such that, $a\mathcal{R}b$ implies a attacks b.

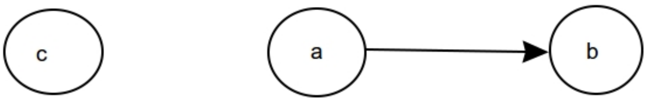


Figure 2.1: An example AF where argument a attacks argument b

However, a disadvantage in an AF is that the attack relation can have several interpretations since the argument structure is not considered. For example, a attacks b can imply that a has a conclusion which is more preferred than the conclusion of b or it can imply that the conclusion of b is a negated conclusion of a. Dung introduced several types of extensions to compute the acceptability of the arguments by grouping the arguments based on the conditions. An extension is, basically a subset of arguments which, together, are a part of an AF which can be believed taking into account all the attacks.

The admissible, grounded, preferred and complete extensions are defined below.

- A subset $\mathcal{S}$ is an admissible set iff $\mathcal{S}$ is conflict-free such that there exist no arguments in $\mathcal{S}$ that attack one another and all the elements of collectively defend the set.
- A subset $\mathcal{S}$ is a grounded extension iff for a characteristic function $\mathcal{F}$, $\mathcal{S}$ represents the least fixed point.
- A subset $\mathcal{S}$ is a preferred extension iff it is a maximal admissible set.
- A subset $\mathcal{S}$ is a complete extension iff for every argument in $\mathcal{S}$, it is defended by an argument in $\mathcal{S}$.

Further, preferred extensions can give rise to credulously or sceptically acceptable arguments. An argument is credulously acceptable if it belongs to at least one of the preferred extensions and is sceptically acceptable if it belongs to all the preferred extensions. In Dung's AF, the support relation is not explicitly present but rather is implicitly present as a collective defense in the extensions since an extension contains arguments that do not conflict with each other.

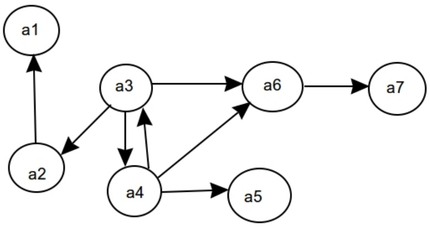


Figure 2.2: An example AF with attack relations between the arguments.

An example abstract argumentation framework is given in Figure. 2.2 with the arrows depicting the attack relation between two arguments. Extensions for the example is as follows:

Admissible $\emptyset$, $\{a_1, a_3\}$, $\{a_3, a_5\}$, $\{a_3, a_7\}$, $\{a_1, a_3, a_5\}$, $\{a_1, a_3, a_7\}$, $\{a_1, a_5, a_7\}$, $\{a_1, a_3, a_5, a_7\}$, $\{a_2, a_4\}$, $\{a_2, a_4, a_7\}$

Preferred $\{a_1, a_3, a_5, a_7\}$, $\{a_2, a_4, a_7\}$

In the example, $a_1, a_2, a_3, a_4, a_5, a_7$ are credulously acceptable arguments and a_7 is a sceptically acceptable argument.

Moving on from AFs, several extended versions have been proposed to remove the drawbacks in Dung's AF. Brewka et al. [31] summarize on the extended work on generalizing Dung's AF in which attacks are prioritised based on preferences and values. The Preference based Argumentative Framework (PAF) [32] takes into account the preference among the attack arguments while making a decision wherein a more preferred argument A1 defeats a less preferred argument A2 when A1 attacks A2. On the other hand, a Value-based Argumentation Framework (VAF) [33] has values for representing the outcome of the arguments and the preferential relation is based upon the values rather than the arguments directly. However, there is an ambiguity present with the usage of preferences or values and hence the Extended Argumentation Framework (EAF) was introduced to produce reasoning about the preferences themselves. This means the arguments can attack other arguments as well as other attacks using the preferences.

While the above existing frameworks all implicitly represent support as a relation in the form of defense, Amgoud et al. [14] is one of the earliest work that explicitly extended Dung's AF with support as an independent relation. Benferhat et al. [34] study two kinds of preferences, positive and negative preferences, for depicting what an agent accepts and rejects respectively. Dubois and Prade [35] discuss different types of bipolarity that exist in knowledge representation which have characteristics such as exclusivity, exhaustivity and duality. Exclusivity refers to a piece of information being both positive and negative, exhaustivity refers to information that is neither positive nor negative and duality refers to inferring negative information from given positive information. Amgoud et al. [14] study bipolarity in abstract argumentation frameworks and argue that the definition of "attack" relation as defined in Dung's AF is a generic term that has different meanings. Two representations of "attack" in Dung's AF are given below.

- an argument attacks another if their conclusions are contradicting
- the conclusion of an argument undermines the premise of another argument

This led to the introduction of support as an independent relation and the extended framework is termed the Bipolar Abstract argumentation Framework (BAF).

Definition 2.2 (**Bipolar Abstract argumentation Framework (BAF)**)**.** A bipolar abstract argumentation framework is a tuple $(\mathcal{A}, \mathcal{R}, \mathcal{S})$ where $\mathcal{A}$ represents the set of arguments $\mathcal{A} = \{a_1, ...a_m\}$ and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ represents the attack relation such that, $a\mathcal{R}b$ implies a attacks b and $\mathcal{S} \subseteq \mathcal{A} \times \mathcal{A}$ represents the support relation such that, $a\mathcal{S}c$ implies a supports c

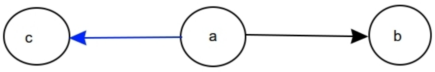


FIGURE 2.3: An example BAF where argument a attacks argument b and a supports c. Blue arrow denotes support relation and black arrow denotes attack relation.

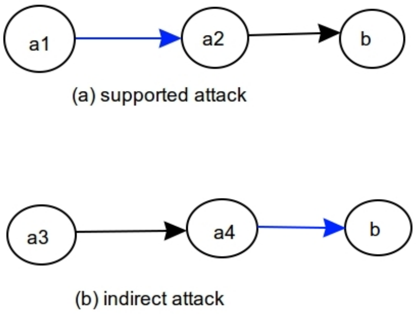
This kind of framework has the following bipolarity characteristics:

- An argument cannot support and attack the same argument.
- Arguments with no relations in the framework may be present.
- Support and attack relations need not occur for the same data and can occur between different data.

Similar to Dung's extensions, several extensions are defined for BAFs for the acceptability of the arguments. To do this, the authors propose two types of attack based on the support relation as follows:

Supported attack For a given argument b, the supported attack is a sequence of arguments $a_1\mathcal{S}_1, a_2, \mathcal{S}_2, ..., \mathcal{R}_1, b$ where $n \geq 3$ where R_1 is the attack relation and S_i are the support relations.

Indirect attack For a given argument b, the indirect attack is a sequence of arguments $a_1\mathcal{R}_1, a_2, \mathcal{S}_1, ..., \mathcal{S}_{n-1}, b$ where $n \geq 3$ where R_1 is the attack relation and S_i are the support relations.



(a) supported attack

(b) indirect attack

FIGURE 2.4: Graphical representation to show supported attack and indirect attack among arguments. Blue arrow represents support relation and black arrow represents attack relation.

These definitions are applied for sets of arguments as follows:

- For a given set $\mathcal{W} \subseteq \mathcal{A}$, and $a \in \mathcal{A}$, $\mathcal{W}$ **set-attacks** a iff an argument present in $\mathcal{W}$ attacks a using a supported attack or an indirect attack relation.

- For a given set $\mathcal{W} \subseteq \mathcal{A}$, and $a \in \mathcal{A}$, $\mathcal{S}$, $\mathcal{W}$ **set-supports** a iff an argument present in $\mathcal{W}$ supports a using a sequence of arguments as $b_1\mathcal{S}_1, b_2, \mathcal{S}_2, ..., \mathcal{S}_{n-1}, a$ where $b_1 \in \mathcal{W}$

In Dung's AF [9], the acceptability of an argument depends on its membership within an extension and this implies that the set of arguments within an extension must be conflict-free. In other words, the arguments within an extension do not attack each other. This concept of conflict-freeness is considered as a form of coherence. In a BAF, this coherence exists as internal and external coherence because of the existence of supported attack relation and the support relation among arguments. The two types of coherence are defined below.

Internal coherence This coherence exists when supported attacks are considered. A given set $\mathcal{W} \subseteq \mathcal{A}$ is conflict-free iff there does not exist any $a, b \in \mathcal{W}$ where a set-attacks b.

External coherence This coherence exists when support and attack relations are considered. A given set $\mathcal{W} \subseteq \mathcal{A}$ is safe iff there does not exist any $b \in \mathcal{A}$ where $\mathcal{W}$ set-attacks b but instead $\mathcal{W}$ set-supports b or $b \in \mathcal{W}$.

The extensions are given below:

Stable extension A given set $\mathcal{W} \subseteq \mathcal{A}$ is a stable extension iff $\mathcal{W}$ is conflict-free and $a \notin \mathcal{W}$ such that $\mathcal{W}$ attacks a.

d-admissible extension A given set $\mathcal{W} \subseteq \mathcal{A}$ is a d-admissible extension iff $\mathcal{W}$ is conflict-free and $\forall a \in \mathcal{W}$, a is supported by $\mathcal{W}$.

s-admissible extension A given set $\mathcal{W} \subseteq \mathcal{A}$ is an s-admissible extension iff $\mathcal{W}$ is safe and $\forall a \in \mathcal{W}$, a is supported by $\mathcal{W}$.

c-admissible extension A given set $\mathcal{W} \subseteq \mathcal{A}$ is a c-admissible extension iff $\mathcal{W}$ is conflict-free and $\mathcal{W}$ is closed for the support relation and $\forall a \in \mathcal{W}$, a is supported by $\mathcal{W}$.

d-preferred extension A given set $\mathcal{W} \subseteq \mathcal{A}$ is a d-preferred extension iff $\mathcal{W}$ is a maximal d-admissible set.

The following properties explain the relation between the different semantics.

Several specialized support relations have been proposed such as deductive support, necessary support and evidential support (has been established within the evidential support framework [36]). A recent study on bipolarity frameworks [37] looks into understanding how these support relations are related to each other and how the support relations are converted from one another. The idea of set-supports has also been dealt with the introduction of coalitions of arguments. At an abstract level, bipolar argumentation graphs can be assembled into meta-level arguments known as coalitions of arguments. Cayrol and Lagasquie-Schiex [38] introduce a framework known as *coalitions of arguments* that is nothing but a conversion of a bipolar argumentation framework into meta-arguments that are connected by the support relation. Earlier work on collective arguments are presented in Bochman [39], and Nielson and Parsons [40] but Cayrol and Lagasquie-Schiex [38] differ in their definition of a coalition from these previous works.

Definition 2.3 (Coalition of arguments). A coalition of arguments is a set of arguments that are related directly or indirectly, connected by a supporting edge and there are no attacking edges present in a coalition. A coalition satisfies the following properties:

1. Each argument in a coalition is directly or indirectly connected by a supporting edge.
2. Any two coalitions are said to attack each other if at least one argument from a coalition attacks at least one argument in the other.
3. There are no attacking edges present in a coalition.

The attack relation is defined as a meta-level relation that occurs between coalitions. Cayrol and Lagasquie-Schiex [38] consider the support relation to have different interpretations and use it as a means of identifying arguments that can be grouped together and in this case, some of Dung's semantics are not satisfied. Several modifications have been introduced to overcome this problem [41, 42].

The work that I have described so far demonstrates the interest shown by researchers of the argumentation community in using support as a relation. Amgoud et al. [14] defines argumentation as a process in which the interaction among the arguments using these relations is valuated by computing the strength of the arguments and this is used for computing the acceptability of arguments. The argumentation process is defined as follows:

1. Identifying or constructing arguments in favour or against a decision.
2. Identifying the relations among the arguments
3. Compute the strength of the arguments
4. Evaluate the acceptability of the arguments
5. Comparing the decisions based on the accepted arguments.

While the different semantics proposed in the above work identify arguments as acceptable or not, this acceptability is not based on any strength values of the arguments, in particular, I have not discussed the notion of graduality in the acceptability values.

Cayrol and Lagasquie [43] proposed a gradual valuation method that computes the strength of an argument in a Dung's AF and a BAF. Some of the earlier work in computational argumentation have dealt with valuating the strength of the arguments. Besnard and Hunter [44] considered scenarios in which some proposition α is the subject of arguments that are for it and arguments that are against it. This work is set in the context logic-based arguments, that is arguments that are constructed from formulae in standard propositional logic. In this context, an argument for α is a logical deduction with conclusion α, and an argument against α is a deduction with conclusion $\neg\alpha$. To summarise opinion about α in the presence of arguments for and against it, Besnard and Hunter [44, page 222] proposed the following accumulator function:

$$Accumulator(\alpha^+, \alpha^-) = \alpha^+ - \alpha^- \tag{2.1}$$

where α^+ is the accumulated value of the arguments for α and α^- is the accumulated value of the arguments against α. If the value in Eq. 2.1 is close to 0, then the arguments that are for and against the formula are considered to be in balance. If the value is positive (negative), then the arguments are strong supporters (attackers) of the formula. In the *counting accumulator* of [44, page 226], α^+ is just the number of arguments for α and α^- is just the number of arguments against α.

Cayrol and Lagasquie-Schiex [43] drew on both the accumulator method from [44] and work such as [45] to define the *gradual valuation function* for arguments.

Definition 2.4 (**Gradual valuation function**). $\forall a \in \mathcal{A}$, where $\mathcal{A}$ represents the set of all arguments, with a set of supporters $\mathcal{B} = \{b_1, ...b_m\}$ and a set of attackers $\mathcal{C} = \{c_1, ...c_n\}$, the strength of the argument is defined as a gradual valuation as below.

$$\nu(a) = g(h^{\text{sup}}(\nu(b_1), ...\nu(b_m)), h^{\text{att}}(\nu(c_1, ...c_n))) \tag{2.2}$$

where $h^{sup}(...)$ and $h^{att}(...)$ are given as above and there are different ways of computing the h function, of which, two are defined below.

$$h^*_{agg}(\mathcal{A}) = \sum_{i=1}^{sizeof(\mathcal{A})} (\nu(a_i)) \tag{2.3}$$

$$h^*_{max}(\mathcal{A}) = \max(\nu(a_1), ...\nu(a_{sizeof(\mathcal{A})})) \tag{2.4}$$

Using the above, the g function is defined as below.

$$g(h^{\text{sup}}_*, h^{\text{att}}_*) = \frac{1}{h^{\text{att}}_* + 1} - \frac{1}{h^{\text{sup}}_* + 1} \tag{2.5}$$

In the above equation, * indicates whether the function uses agg or max for computing the values.

Although not directly related to the argument strength, Leite and Martins [46] introduced the social abstract framework as an abstract framework that considers voting as present in social media content. A Social Abstract Argumentation Framework (SAF) is an abstract framework as a triple $(\mathcal{A}, \mathcal{R}, \mathcal{V})$ where $\mathcal{A}$ is a set of arguments, R represents the attack relation as a binary relation between the arguments and V represents a function that maps each argument with the number of positive and negative votes. They propose a simple vote aggregation function that computes a value based on the positive and negative votes. The authors of [46] show that the work can be useful for social media debate portals but have not empirically evaluated it. Similar to this, Baroni et al. [47] introduced the Quantitative argumentative Debate framework (QuAD) for evaluating answers in IBIS graphs as a 5-tuple $(\mathcal{A}, \mathcal{C}, \mathcal{P}, \mathcal{R}, \mathcal{BS})$ where $\mathcal{A}$ is a set of answer-arguments, $\mathcal{C}$ is a set of con-arguments, $\mathcal{P}$ is a set of pro-arguments, $\mathcal{R}$ is an acyclic binary function and $\mathcal{BS}(a)$ is the base score of argument $a \in \mathcal{A}$.

Apart from the frameworks discussed above, the Abstract Dialectical Framework (ADF) [48] was proposed with the intention of having a flexible relationship among the arguments using a conditional acceptance criterion for each argument with every argument depending upon its corresponding parent argument. In this case, preferences can be used as an alternative for the acceptance criterion as well as has been implemented for logic-based approaches. One of the future research areas proposed by the authors of [48] is to find acceptance conditions that aren't logic based.

Moving from these logic-based frameworks, there has also been a shift towards informal logic with the idea of dealing with practical reasoning or everyday argumentation as discussed by philosophers such as Aristotle. The idea of representing propositions as informal arguments is dealt with the notion of considering arguments as structured in which propositions are premises that infer a conclusion. In structured argumentation, the knowledge is represented as a formal language and this is used for constructing arguments. Some of the notable work in structured argumentation dealing with defeasible logic are assumption-based argumentation (ABA) [49], ASPIC+ framework [50] and Defeasible Logic Programming (DeLP) [51].

Assumption-based Argumentation (ABA) [49] is based on Dung's AF and arguments are represented as deductions that are supported by assumptions. Assumptions are sentences present in the formal language. The attack relation between the arguments is based on whether the assumptions contradict each other. The ASPIC+ framework [50] works on a formal language in which arguments are constructed as trees using two types of rules: strict and defeasible rules. Strict rules are rules that provide strong information for the acceptance of premises for a conclusion and defeasible rules are rules that do not provide sufficient information for the acceptance of the premises for a conclusion. An argumentation system is a triple $(\mathcal{L}, \mathcal{R}, m)$ where $\mathcal{L}$ is the formal language which is closed under negation, $\mathcal{R}$ contains the strict and defeasible inference rules and m represents a partial function that maps these rules to the formal language.

The idea of representing arguments as diagrams or visualisations in the form of graphs has been influenced by the works of Dung [9], Toulmin [1] and Freeman [29]. While Dung's abstract argumentation framework work on an abstract level, relating arguments in a graphical representation, Toulmin's model explored the idea of introducing different components apart from the premises and conclusion constituting an argument. Toulmin's model is represented using the following components:

Data Premises that are used to represent the argument.

Claim The main conclusion of the argument.

Warrant Logical justification or generalization for connecting the data with the claim.

Qualifier A modal qualifier that implies the strength of the inference step from the data to the warrant.

Backing Statements that support the warrant.

Rebuttal Counter-arguments that attack the argument.

An example of Toulmin's model of arguments is given in Figure. 2.5.

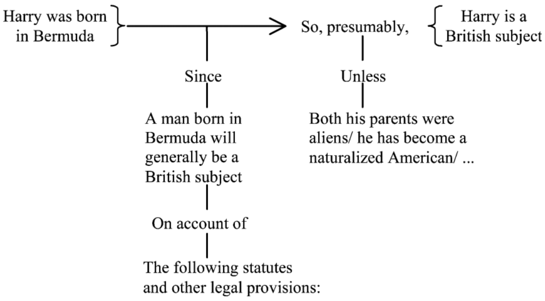

Harry was born in Bermuda
So, presumably,
Harry is a British subject
Since
Unless
A man born in Bermuda will generally be a British subject
Both his parents were aliens/ he has become a naturalized American/ ...
On account of
The following statutes and other legal provisions:

FIGURE 2.5: An example of Toulmin's model of arguments ([1])

While Toulmin's model provided a detailed diagrammatic representation of an argument, it did face criticisms on the way the different components were distinguished. For example, Klein argued that an argument can be represented as a support tree with the claim as the root and in that, it is difficult to distinguish between Toulmin's definition of a data and a warrant. Freeman [29] argued that Toulmin's model is complicated and that some of the components may or may not be present in an argument. In his work, Freeman proposed different argument diagrams with two main components: premises and conclusion and explained how different argument structures are built upon this simple structure by answering questions. Freeman considers the process of answering questions and adding additional premises in a premise-conclusion structure in the form of a dialectical process in which a proponent is questioned by an opponent. Freeman also suggests that warrant and backing are additional premises that support by filling a gap in a premise-conclusion structure. The additional premises that strengthen the main premises are termed *defended rebuttals*.

Freeman's model contains the following elements:

Serial Premises are linked in a chain fashion where a premise supports another premise and so on and, together support a conclusion.

Linked Premises are not related to each other but when grouped together, support a given conclusion. Freeman treats this form of a structure to answer the following question: *Can you provide additional premises that provide reasoning to explain why a premise is connected to a conclusion?*

Convergent Premises are not related to each other and are disjunctively joined together to support a given conclusion. Freeman distinguishes this type of a structure from the linked structure using the following question: *Can you provide other reasoning that supports a given conclusion?.*

Pollock [8] explains about two types of defeat in defeasible reasoning namely undercutting and rebuttal defeaters. A rebuttal defeater is a premise providing reasoning that attacks the premise of a given conclusion and an undercutting defeater is a premise providing reasoning that attacks the conclusion. While Freeman's model has its own disadvantages such as not being able to distinguish between undercutting and rebuttal defeaters, the generalizations of premises in Freeman's models has been useful for constructing arguments in natural language texts [29]. A rebuttal defeater is a premise providing reasoning that attacks the premise of a given conclusion and an undercutting defeater is a premise providing reasoning that attacks the conclusion.

Some of these argumentation models have also been studied for natural language arguments and how they can help in decision making. Some of the existing works in argumentation have carried out an empirical study using human evaluation on natural language arguments considered as abstract argument representations. For instance, Cerutti et al. [52] compare the decision made by people with that of an abstract argumentation framework, while Rosenfeld et al. [53] empirically show why argumentation theory alone does not help in correlating with the decisions made by people. The authors of [53] propose a Predictive and Relevance-based heuristic agent which predicts people's choice with an accuracy of 76%. However, in both these works, arguments are not automatically mined from the textual content as present in social media texts. Recently, Amgoud and Prade [54] discuss how linguists treat an argument as a relation between two functions, one each for representing the premises and conclusion respectively. The authors find that most of the premises within natural language texts are left implicit and exhibit themselves as enthymemes. Building upon this, Amgoud et al. [15] discuss why traditional logical based argumentation fails to represent arguments that are present in natural language texts and dialogues. They propose a formalism based on theories existing in linguistics that can help in representing mined arguments.

Formal representation of natural language data has been empirically evaluated using logical and abstract based argument models [52, 53] but no such work dealt with social media content until a few years back. The advancement in the field of natural language processing domain has led to the emergence of a research area known as argument mining [16–18] aimed at integrating the theoretical aspects of argumentation theory with natural language processing techniques for mining arguments and the relation between them present within natural language texts. Recently, social media content has also been explored for argument mining tasks such as identifying arguments and the relations among the arguments.

In the next section, I give a brief description of some of the topics in natural language processing domain that are relevant to the argument mining tasks and for answering the research questions present in this thesis.

2.2 Natural Language Processing

Sentiment analysis and opinion mining have been an ongoing research area for many years [25] and are still topics of interest. For example, given an online review, we would want to know whether it is positive or negative and how many opinions in the review are positive or negative. Similarly, given a set of tweets about a topic, we are interested in knowing whether the tweets are positive or negative with respect to the topic. In sentiment analysis, the sentiment of a content is identified by extracting linguistic features present in that content and using the features with machine learning techniques. We find numerous publications [25, 55, 56] on mining opinions present in online reviews and predicting the sentiment of the opinions as positive, negative or subjective. A positive (negative) sentiment means that an opinion in a review infers that the reviewer likes (dislikes) the product/service. A subjective opinion means that it is difficult to infer whether the reviewer likes or dislikes a product/service. Pang et al [25], one of the earliest works on identifying the sentiment of opinions, studied the problem of identifying the sentiment of opinions present in movie reviews and observed that using keywords that indicate the positive or sentiment is not better than using unigrams in a machine learning classifier. In the recent SemEval conferences, a lot of tasks have been aimed at sentiment analysis as in identifying the sentiment of tweets [57], aspect-based sentiment analysis [26] etc. Aspect-based sentiment analysis aims at identifying the sentiment of a content with respect to the aspect it is talking about.

Recently, there has been a growing interest in representing words as vectors in a k-dimensional space, known as word embeddings, and there are two main approaches followed to learn these vectors — count-based and prediction-based. In the count-based approach, given a set of documents, the co-occurrences matrix containing the frequency of a word co-occurring with two or more words is considered for learning the word representations. In the prediction-based approach, there are two main models that are trained namely CBOW (Continuous Bag-of-Words) and Skip-gram model. In the CBOW model, given a target word, the model predicts the target word using its context words whereas, in the Skip-gram model, a target word is considered for predicting the surrounding words or context words. One of the earliest works in developing embedding vectors for words is the word2vec [58] Skip-gram model that predicts the neighbouring words for a given target word. Another popular work is Glove [59] embeddings that make use of the co-occurrences probabilities of words.

A growing interest in topics related to word embeddings is reflected by the numerous papers published in various NLP conferences. Recently, Bollegala et al. [60] propose a method to learn a linear transformation between the two approaches to study the difference between the two vector spaces and empirically evaluate it. Maas et al. [61] explore the idea of learning word embeddings that can target features useful for sentiment analysis. The idea of representing sentences as embedding vectors has also been studied nowadays. The simplest method for representing a sentence embedding is to average the word embedding vectors of a sentence and this has been a strong baseline for several tasks. Quite a few unsupervised and supervised approaches have been proposed for representing sentences as embeddings and these have shown to perform well for tasks such as text classification, semantic relatedness etc. Among the unsupervised approaches, Arora et al. [62] is the state-of-the-art unsupervised method that modifies this baseline by representing a sentence as the weighted-average of the word embeddings for the words in that sentence. Infersent [63] is a supervised approach proposed for sentence representations and is trained on the Stanford Natural Language Inference dataset using a Bidirectional long short-term memory neural (Bi-LSTM) model. All the sentence representations explained above have not been developed for identifying argumentative properties but can certainly help in understanding the structure of arguments.

The task of identifying texts that have similar meaning has gained popularity over the years and has been useful for tasks such as text summarization and machine translation. Semantic Text Similarity (STS) and Recognizing Textual Entailment are two tasks that are conducted by the SemEval conference for evaluating systems that identify similar texts.

The semantic similarity task takes two sentences as input and produces a continuous valued score in the range between 0 and 5 where 0 represents least similarity and 5 represents highest similarity. The Takelab semantic text similarity system [64] is a system that was submitted to the SemEval conference 2012 (Task 6) and ranked among the top 5 out of 89 systems. Different linguistic features were used for training a supervised regression classifier and experiments were carried out with different training datasets. The training datasets contain pairs of sentences with human annotated similarity scores ranging from 0 (least similarity) to 5 (highest similarity). Some of features are described below:

Ngrams overlap Given two sentences $\mathcal{S}_1$ and $\mathcal{S}_2$, the ngram overlap where ngram = {unigrams,bigrams,trigrams} is defined as follows:

$$\text{ngram}(\mathcal{S}_1, \mathcal{S}_2) = 2.\left(\frac{|\mathcal{S}_1|}{|\mathcal{S}_1| \cap |\mathcal{S}_2|} + \frac{|\mathcal{S}_2|}{|\mathcal{S}_1| \cap |\mathcal{S}_2|}\right)^{-1} \tag{2.6}$$

Wordnet-based word overlap Given two sentences $\mathcal{S}_1$ and $\mathcal{S}_2$, a partial score is assigned to each word that is not common to both the sentences using wordnet lexicon and the overlap is computed as the harmonic mean of $\mathrm{P}(\mathcal{S}_1, \mathcal{S}_2)$ and $\mathrm{P}(\mathcal{S}_2, \mathcal{S}_1)$ where $\mathrm{P}(\cdot : \cdot)$ is defined as follows:

$$\mathrm{P}(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{|S_2|} \sum_{w \in \mathcal{S}_1} \mathrm{score}(w, \mathcal{S}_2) \tag{2.7}$$

where

$$\mathrm{score}(w, \mathcal{S}) = \begin{cases} 1 & \text{if } w \in \mathcal{S} \\ max\mathrm{sim}(w, w') & \text{otherwise} \end{cases}$$

Vector space sentence similarity Given two sentences $\mathcal{S}_1$ and $\mathcal{S}_2$, $\boldsymbol{v}_{\mathcal{S}_1}$ and $\boldsymbol{v}_{\mathcal{S}_2}$ are their respective vectors obtained by summing up the vectors of each word present in the sentence. The cosine similarity between the two vectors is computed and used as a feature.

The Recognizing Textual Entailment (RTE) task takes a text (T) and a hypothesis (H) as input and returns the relation as entailment, contradiction or neutral. There exists an entailment relation between a text and a hypothesis if the hypothesis is inferred from the text. In contrast, there exists a contradiction relation between a text and hypothesis if the hypothesis inferred by the text contradicts the given hypothesis. This task has been investigated by a lot of researchers [65] and several gold-standard datasets have been created. Different RTE datasets have been released for the PASCAL RTE challenge such as standard RTE [66], SICK [67] and EXCITEMENT [68] and several papers have tackled this problem. For example, Yokote et al. [69] propose a model that transforms similarity measures into a non-linear transformation for predicting textual entailment. Another example is Zanzotto et al. [70] that investigates on identifying patterns based on subject-verb relation to identifying entailment. In their paper, they argue that the logical entailment present between the text and hypothesis is not captured properly.

Stance detection has been a hot topic among NLP researchers. This deals with identifying the standpoint taken by the user. It has also attracted the attention of argumentation researchers since both arguments and stance are closely related to each other. Shobani et al. [71] discuss how identifying argument tags in texts can also help in stance classification. In their work, [71] identify topics using unsupervised approaches for mapping them to the arguments and hence make the annotation process easier. Walker et al. [72], Somasundaran and Wiebe [73] and Hasan and Ng [74] identify stance based on the argumentative properties present within the texts. Somasundaran and Wiebe [73] identify argument based lexicon present in an annotated corpus automatically using a supervised approach and results show that it outperforms sentiment-based methods. A recent work by Valerio et al. [75] finds an interesting pattern between the argument structures present within debates, the polarity of the argument present and the emotion of the user and in their findings, they show that there are mismatches between the polarity and the emotion which can be overcome by understanding the support and attack relations among the arguments.

The idea of exploring natural language techniques and computational argumentation to identify arguments in natural language texts has given rise to a new field known as argument mining. In the next section, I explain the different existing work in argument mining.

2.3 Argument mining

Argument mining is the study of identifying arguments and their relations present in natural language texts. Existing works in argument mining consider arguments as abstract arguments or as structured arguments. Abstract arguments are arguments that do not have any internal structure whereas structured arguments are of the premise-conclusion form. I begin with a brief summary of those work that considers arguments as abstract arguments.

Cabrio and Villata [76] extract abstract arguments from debates to form a bipolar argumentation framework, with the support and attack relation automatically identified using textual entailment. In this paper, the authors empirically demonstrate that, in most cases, support and attack relation satisfy entailment and contradiction relations respectively. In contrast, textual entailment does not give good results for extracting these relations among tweets [77]. Similar to this, Yaglikci et al. [78] investigate microdebates, which are arguments posted on twitter using the Microdebates app, and how the arguments present are evaluated as a weighted abstract argumentation framework. The authors conducted a survey by asking different participants to debate on topics and further analyse their behaviour and analyse how it correlates with the weighted abstract argumentation framework. This work gives us an overview of how the behaviour of people affects the performance of computational argumentation models in assessing the arguments. Cocarascu and Toni [79] make use of bipolar argumentation frameworks for analysing the support and attack relations present in online reviews based on a topic. In their work, they conduct a pilot experiment in which they group user comments in a temporal fashion and identify the support and attack relations. The work can be further developed to evaluate the BAF graphs against human annotation or any other ground truth data for assessing the usefulness of the method. Recently, Pazienza et al. [80] propose a new framework based on BAF known as Bipolar Weighted Argumentation framework which is evaluated for online debates. They use strength propagation methods for weighing the relations and apply it to Reddit comments discussion data. The work is focussed on addressing online debates and has not been evaluated against any other prior work that has applied argument mining techniques on online debates. Previous to this work, Patkos et al. [81] and Leite and Martins [46] have also proposed their own framework for evaluating arguments that are present in user comments. But, neither of these frameworks have been illustrated using real-world datasets.

Method	**Sentiment**	**Stance**	**Topic**	**BAF**	**Coalitions**
Wyner et al. (2012) [88]	No	No	No	No	No
Wachsmuth et al. (2012) [89]	Yes	No	No	No	No
Liu et al. (2017) [90]	No	No	No	No	No
Dragoni et al. (2018) [90]	Yes	No	Yes	No	No
Cocaracsu and Toni (2012) [91]	Yes	No	Yes	Yes	No
Villalba and Saint-Dizier (2012) [92]	Yes	No	Yes	No	No
Proposed work	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Table 2.1: The features and methodologies used in this thesis (proposed work) is compared with existing work on mining arguments and their relations in online reviews. *Sentiment*, *stance* and *topic* are NLP-based features used for representing structured arguments and, *BAF* and *Coalitions* denotes whether bipolar argumentation framework and coalitions of arguments are used for representing arguments as abstract arguments.

Based on the domain at hand, we can further categorize existing works based on structured arguments as one, where the work is based on monological texts and the other, where the work is on dialogical texts. Some of those based on monological texts deal with persuasive essays [82], articles [83, 84] and online reviews [85]. Stab and Gurevych [82] propose a joint model for identifying argument structures of the premise-conclusion form present in persuasive essays. The model was evaluated on a novel corpus containing 402 persuasive essays and the results have shown that the proposed model outperforms baseline classifiers. Another work on essays, Persing and Ng [86] propose a novel corpus on student essays and an argumentative approach for computing the strength of arguments present in essays. Their proposed classifier uses features such as part-of-speech n-grams, semantic frames, coreference etc. and results show that this approach outperforms the baseline classification. The classifier performance is not evaluated against other domain datasets that have a different structure. Lippi and Torroni [87] propose a method using partial tree kernels for detecting claims present in articles without depending on the context of the texts. An advantage of this work is the empirical evaluation on the large IBM dataset that contains annotated claims which show that the proposed method works well as other state-of-the-art methods that take context into consideration. This method has not been investigated for other domains such as reviews where the structure of a claim is different.

Tabe. 2.1 represents the different features and methodologies that are used in this thesis and are compared with existing work on mining arguments from online reviews. Wyner et al. [88] is one of the earliest work for identifying argumentative structures present in online reviews. In this paper, different argumentation based schemes useful for identifying arguments are proposed but further work has not been carried out on it. Wachsmuth et al. [89] developed the ArguAna corpus for studying the argumentative patterns present in online reviews of hotels. Each review is manually annotated by three expert annotators for the following:

- Each sentence in a review is annotated as positive, negative or neutral.

- Each aspect in a sentence is labelled. Some examples of aspects are *hotel*, *service* etc.

The ArguAna corpus contains manually annotated sentences from 2100 reviews that are annotated as positive, negative or objective based on the sentiment of the sentences along with identifying aspects present within the sentences. The main objective behind the creation of this corpus is to identify argumentative patterns based on the sentiment of the sentences and the usefulness of these patterns for text classification is studied.

Liu et al. [90] developed a corpus containing 110 reviews from TripAdvisor hotel reviews and each sentence in the review is manually annotated by three annotators using the following component labels: *major claim*, *claim*, *premise*, *premise supporting an implicit claim*, *background*, *recommendation* and *non-argumentative.* All these components are investigated on whether argument-based features help in predicting the usefulness of reviews. Results show that combining argument-based features with baseline features can improve the performance of the classification.

Dragoni et al. [93] present a system that combines argumentation with aspect-based opinion mining using three main components namely argument module, sentiment module and visualization module. In the argument module, sentences containing aspect based opinions are extracted as arguments and I follow a similar approach in this thesis. In the next step, which is the sentiment module, the sentiment of the argument with respect to each aspect is detected and in the final step, the relations are visualized. The system is provided as a user interface for users to identify the most positive and negative opinion and assumptions are made on how the argumentation graph is built. An advantage of this system is its ability to show researchers how the graph is produced and for experts to understand how aspects are summarized.

Dialogical domain deals with debates, tweets, dialogues and other forms of user interaction. Ghosh et al. [94] annotate user comments in forums as target-callout pairs based on pragma-dialectic theory and also investigate on the difficulties faced in doing the annotation task. This work is useful for the research community to understand the difficulties of annotating arguments in social media texts. Boltuzic et al. [95, 96] have done a continuous assessment of the task of automatically identifying premises and claim, in particular, how they are related to each other in debates. Their definition of support and attack depends on whether the relation is explicit or not. The authors also have created a dataset consisting of 125 claim pairs containing annotated premises for filling the gap between a user claim and the main claim of a topic. An advantage of this work is the availability of the dataset that can be used for comparing whether the model proposed can be useful for other available datasets.

Habernal et al. [97] build a large corpus based on the extended Toulmin model from debate portals using a semi-supervised approach. The different components annotated to represent an argument are the following:- premise, claim, backing, rebuttal and refutation. The semi-supervised approach automatically extracts features from an unlabelled corpus by clustering word embedding vectors for classifying whether a given sentence is an argument or not. An advantage of this work is the semi-supervised approach that has been evaluated in detail for in-domain and cross-domain data along with detailed error analysis. Differing from the rest, Duthie et al. [98] work on a political based debate corpus to identify *ethos* which is linked with the credibility of the user. Walker et al. [99] determine how persuasive arguments are from the audience perspective while Oraby et al. [100] classify a dialogue based on whether it is factual or emotional. Experiments conducted in [100] show that there are patterns present in factual arguments containing argument phrases whereas emotional arguments tend to be claims/arguments based on the user's beliefs. In this thesis, some of the arguments extracted can be considered as emotional arguments and can be useful for analysing the patterns in online reviews, which is not addressed in the prior work.

Not only does argument mining focus on annotating arguments and its components, but there are also quite a number of works that deal with relation extraction too. As discussed earlier, Boltuzic et al. [96] relate arguments using implicit/explicit support and attack relations. Similarly, Bosc et al. [101] annotate the support and attack relation among arguments present in tweets. Park and Cardie [102] focus on identifying different forms of support relations that are present among user comments. They classify the support relation based on whether evidential reasoning is present or not. Among scientific articles, the support and attack relation were extracted by Kirschner et al. [103].

Instead of extracting relations, Carstens and Toni [104] investigate towards how relation information can help in identifying arguments. In their work, they show how in many cases, the objective statements often ignored can actually constitute an argument. Thus, they consider pairs of sentences that satisfy either the *support*, *attack* or *neither* relation to demonstrate the same. Cocarascu and Toni. [105] extend this work by investigating on how deep learning can help in identifying the relations. An advantage of the above works is that it helps in identifying arguments that may be difficult to extract without understanding the relations between them. Taking a new direction in this process, Habernal and Gurevych. [106] compare arguments using human annotators on which is convincing more than the other. A gold standard corpora with 11,650 argument pairs were annotated as A > B or A < B. An RBF-SVM with different features such as unigrams, bigrams etc. resulting in a feature vector of dimension 64K is used to automatically classify the argument pairs. The binary classification is also carried out using a Bi-directional Long Short Term Memory neural model consisting of two bi-directional networks, each with an input layer consisting of pre-trained embedding vectors. These are concatenated to a single dropout layer and further to a sigmoid layer for classifying the argument pairs as A > B or A < B. The results show that such complicated argument relations are difficult to predict using current NLP techniques.

A few other works use argumentation schemes for extracting arguments and the relations between them. Peldzus and Stede [107] experiment on a set of microtexts for a joint prediction of arguments and their relation using Freeman's argumentation schemes. In their work, they provide a detailed description for modifying the schemes and how it can help to understand natural language arguments. Feng and Hirst's [108] earlier work developed on using argumentation schemes for classifying arguments and enthymemes. Not directly related to enthymemes, Green [109] defined semantic rules for identifying arguments where the conclusion is left implicit and experiment over biology-related articles. It is not clear how these rules can be helpful for other domains where the structure of the arguments is different. Recently, Razuvayevskaya and Teufel [110] as well as Becker et al. [111] have shown how most natural language texts are in the form of enthymemes. Habernal et al. [112] automatically identify warrants that are implicitly stated in arguments in debates, which are similar to enthymemes. This work shows the growing interest in discovering implicitly present information in social media texts.

2.4 Summary

In this chapter, a brief summarization on background topics in computational argumentation and natural language processing was studied. Argument mining is a relatively new research area that combines computational argumentation and natural language processing techniques for understanding how arguments are present in social media texts. Recent work in the field of argument mining was explored with their advantages and challenges along with how the pipeline of steps followed by this thesis differs from the rest of the work. This chapter provides an explanation of the existing work that I make use of in the different steps of the argument mining pipeline proposed in this thesis. The different steps followed in the proposed argument mining pipeline, to understand arguments and enthymemes present in opinionated texts, are explained in the next chapter.

Chapter 3

Argument Mining Pipeline for Opinions

The internet has become an important platform for people to share and gather information. A vast amount of information available is highly unstructured and is found in the form of user reviews and online discussions. People rely on these reviews to make a decision on whether a product is good or bad and if it is profitable to purchase it. The reviews are not only limited to products but can also be relevant to hotels, restaurants and other services. With the increase in the number of reviews available online, it becomes a daunting task for people to read them and make a decision. This chapter introduces an argument mining pipeline and a method for evaluating the pipeline and its argumentative approach for predicting the overall sentiment of reviews. In this chapter, I discuss the different steps in the pipeline, the research questions that are answered in each of these steps and which chapters address them, how the different steps are connected and what existing methodologies are adapted and why they are considered in this work.

The following steps represent the argument mining pipeline that I propose in this thesis:

INPUT A set of reviews for some product/service.

1. Identifying opinions as argumentative based on their linguistic properties: sentiment, stance and topic.

2. Identifying explicit and implicit opinions based on how stance is expressed in the opinions.

3. Identifying relations among arguments for constructing Freeman-style [29] serial and linked argument structures in favour/against a decision.

4. Computing the strength of the opinions by identifying attack and support relation among opinions as arguments.

5. Aggregating opinions as coalitions of arguments and assessing their strength for the overall sentiment prediction task.

OUTPUT This gives the overall opinion of a set of reviews for the product/service.

In argument mining, two main steps are involved in the process, identifying arguments and identifying the relationship between these arguments. Starting with this aim, the first step involves identifying opinions that are argumentatively related to each other. By looking at different linguistic attributes, it gives us a way of understanding the opinions better. To do this step, I consider three different linguistic properties – sentiment, stance and topic. In order to explain why I chose these three properties, I consider two hypotheses to illustrate how the three different linguistic properties fail to capture the support relation between arguments if the properties are not considered together. These hypotheses were used to understand why the three properties together help in identifying arguments and the relation among the arguments and hence I do not evaluate them further in this thesis. Some examples are given below.

Hypothesis 1 Two opinions with the same sentiment support each other.

Let us consider an example where the hypothesis is true but there is no justification or evidence to support the relation.

"*not good enough for a hotel charging these prices*"

"*Very bad!*"

In the above example, although the two opinions support each other, there is no evidence to show that the opinion "*Very bad!*" is about the topic "*hotel*".

Hypothesis 2 Two opinions with the same sentiment and same topic support each other.

Let us first consider an example where it is true.

"*The room is clean*"

"*The room is good*"

Let us consider an example where the hypothesis fails.

"*not good enough for a hotel charging these prices*"

"*the problem with the hotel is the staff*"

In the above example, although both the opinions are about *hotel*, the first opinion is specifically about the *price* of the hotel and the second opinion is about the *staff* of the hotel. These two aspects are unrelated to each other. Let us consider another example.

"*the staff were helpful and polite*"

"*the staff was great*"

In the above example, in the second opinion the stance is explicitly expressed about the *staff* whereas in the first opinion the stance is implicitly expressed using adjectives such as "*helpful*" and "*polite*". Here, the first opinion can support the second opinion but we do not know whether the second opinion supports the first opinion since it does not specifically talk about the staff.

These examples show us that a combination of the three linguistic attributes is a useful way of choosing opinionated texts as argumentative. Wachsmuth et al. [89] studied the sentiment patterns of reviews in relation with the overall sentiment of reviews (projected by the overall star ratings) and found that most of the negative opinions are present in reviews with an overall negative sentiment and most of the positive opinions are present in reviews whose overall sentiment is positive. Therefore, for step 1, opinions whose sentiment is positive or negative and which are about a topic or topics related to the subject of a given review (in reviews these are known as aspects) are extracted from a set of hotel reviews. By extracting opinions that are positive or negative, we are able to compare supporting and attacking opinions that support/attack the overall sentiment of reviews.

The second step in the argument mining pipeline is to automatically classify these extracted opinions as explicit or implicit based on whether the stance is expressed in the content explicitly or implicitly. Presuppositions or information that is left out or missing has been widely studied in pragmatics and argumentation as enthymemes [113], implicit warrants and unexpressed premises [114]. In Chapter 4, I discuss automatically classifying these extracted opinions as explicit or implicit opinions based on how the stance is expressed (Step 2) using natural language processing, machine learning and deep learning methods. This kind of classification gives us two types of opinions, one in which the information is left missing and the other in which this information is explicitly expressed. An interpretation of these opinions as enthymemes and arguments is studied.

An enthymeme, in Aristotle's view, is a logical syllogism where the major premise is missing. It is not necessary that this kind of a syllogism is present in natural language texts and Hamilton[115] argues that Aristotle's definition of enthymeme does not always apply- enthymemes need not always have a logical construction but instead are based on signs. Research in informal logic has some notable works representing argument diagrams like Toulmin's model [1] and Freeman's argument structures [29] that were proposed for everyday argumentation. A drawback of Toulmin's model as argued by Freeman is the identification of warrants that are not explicitly represented in everyday argumentation. Freeman's [29] argument diagrams consider implicit warrants and unexpressed premises as filling the gap between a premise and a conclusion.

The research question that is answered by performing the second step is as follows:

Research Question 1a: How is implicit information identified in natural language arguments present within opinionated texts?

Research Question 1b: How does "stance" in opinions help as a means of filling the gap between a premise and a conclusion?

The above questions address the monological structure of an opinion by considering its linguistic construct and gives one possible way to reconstruct a complete argument. The next step is to understand the kind of information that is helpful to fill the gap by relating different opinions that support a particular conclusion. In doing so, we are able to identify generalisations for a premise which can either become a warrant or an unexpressed premise. To do this we adopt Freeman's model of an argument. An advantage of Freeman's argument diagrams in comparison with Toulmin's model is Freeman's model's simplicity in identifying relations among opinions as it is easily adaptable for natural language arguments that exist in social media texts. One main reason for this is because it does not consider the logical inferential relation between the premises and the conclusion.

The third step of the argument mining pipeline is discussed in Chapter 5. In this chapter, the different support relations that help to construct two different Freeman-style argument structures using implicit and explicit opinions are explored – serial and linked argument structures. This is done by answering the following research question:

Research Question 2: How does "stance", "sentiment" and "topic" help in relating opinions as premises supporting a conclusion and what kind of argument structures are obtained?

Two types of relation exist among opinions based on the three linguistic properties: support-based entailment and rephrase. As discussed earlier, a relationship between the topics or aspects is present. The information or knowledge that relates the topics relates two premises and the gap between them can be filled using the knowledge. A knowledge base representing the ontology between the aspects is created and used for predicting the support relation between a specific premise and a generalised premise. This gives rise to a serial argument structure where a specific premise supports a generalised premise and so on and, together support a conclusion. This type of relation is analogous to entailment studied in textual entailment but the latter does not take into consideration the knowledge base.

This type of a support relation in a serial argument structure occurs between two explicit opinions or between an implicit and an explicit opinion. But, in the case of an implicit and an explicit opinion, it need not always have an entailment-based relation.

For certain opinions, explicit and implicit opinions are two ways of expressing the same argument. Again, recalling what Freeman suggests, premises in a linked argument structure are either warrants or unexpressed premises for a given premise. The difference between the premises being warrants or unexpressed premises depends on the generalisation of these premises that are used along with a given premise. By assuming that an explicit opinion is a generalisation of an implicit opinion such that together independently can support a conclusion, the definition of a rephrase relation [116] among premises is considered for identifying the corresponding explicit and implicit opinions that can become linked arguments. The rephrase relation, closely related to, but different from paraphrasing, is where one premise can replace another without changing the meaning of the argument.

These kinds of argument structures, although simple, become complicated with the introduction of attacks which can either be rebuttals or undercuts. In this thesis, I focus on understanding how argument structures are formed using support relations for both conclusions: (a) the reviewer likes the product/service and (b) the reviewer does not like the product/service and hence do not focus on attack relations. Moving away from these notions, an abstract bipolar argumentation framework gives a simple diagram that does not look into the internal structure of arguments but rather, the attack and support relation is on a higher level between the arguments. This kind of an attack/support relation can correspond between a premise/conclusion of an argument with a premise/conclusion of another argument.

In the next step, the Freeman-style serial and linked argument structures are converted into different bipolar argumentation graphs [38]. One reason for building bipolar argumentation graphs is their adaptability to identify support and attack relations among natural language arguments that have been demonstrated in the relevant literature on argument mining. For example, Cabrio and Villata [76] map textual entailment relations with the support and attack relation and analyse the differences among them. Similarly, I explore the use of sentiment, topic and semantic similarity for identifying support and attack relations among opinions. This kind of support/attack relation supports or attacks the overall conclusion of the opinions that are left implicit in the opinions. The bipolar argumentation graphs are further explored for answering the following research questions which are addressed in Chapter 6:

Research Question 3: Can bipolar abstract argumentation help in computing the strength of the identified opinions present in the Freeman-style arguments?

Research Question 4: What kind of an argument structure can we build when the internal structure relating these opinions is ignored and how does "stance", "sentiment" and "topic" affect this?

In the previous step, the relations among topics and the topics themselves were an important aspect for identifying serial/linked argument structures. The impact of topics on opinionated texts as present in online reviews has been studied widely in the NLP domain. By studying these topics and the relations among them, it is interesting to analyse how argumentatively there are related. Although the linked/serial argument structures capture this relation, the relation does not provide any reasoning that connects the structures with the overall sentiment of reviews. An important task of the argument mining pipeline is to assess whether it is useful for an NLP based task and hence, studying the sentiment, stance and topic with respect to the overall sentiment of reviews can provide a justification for performing the steps of the pipeline.

Bipolarity in abstract argumentation [38] suffers from several complex issues in which the introduction of support relation gives rise to several notions of attack relations and thus different results for the acceptability of arguments. To address this, bipolar argumentation graphs are converted into coalitions of arguments [14] to represent meta-arguments that are supporting each other directly or indirectly. Here, the support relation is replaced using the coalitions such that these coalitions are represented in a Dung's abstract argumentation framework. But, the strength of these coalitions of arguments has not been studied. Another research question that is addressed in Chapter 6 is:

Research Question 5a: Can converting bipolar argumentation graphs into coalitions of arguments represent the strength of combined arguments about a topic?

Research Question 5b: If coalitions of arguments represent the strength of combined arguments about a topic, can different coalitions represent the overall sentiment of a set of opinions in a review?

There has been no work on assessing how the strength of combinations of arguments relates to the human assessment of arguments. In computational argumentation, there are two kinds of approach for combining arguments, argument accrual [117–119] and coalitions of arguments [14]. The coalitions of arguments model is used in this thesis and the reasons to do this are as follows:

- the support relation for combining arguments can be any kind of support relation
- it is modelled by converting bipolar argumentation graphs
- there is existing work in the literature that studied the computation of strength of arguments [43, 44] based on the support and attack relations in a bipolar argumentation graph [38] which can be used for computing the strength of combined arguments.

This makes it easier to adapt for opinionated texts and in this thesis, I propose different ways of computing the strength of the coalitions for predicting the strength of a set of opinions in a review, which in turn, is used to predict the overall sentiment of that review.

To conclude, this chapter gives an overview of the different steps of the argument mining pipeline and the corresponding chapters and the different research questions addressed. The following chapter addresses the second step of the argument mining pipeline by extracting opinions that are argumentative from a set of hotel reviews and discusses the implicit/explicit opinion classification of the extracted opinions using machine learning, natural language processing and deep learning methods. By doing so, the research question on how implicit information is identified is studied. The implicit/explicit opinions are further studied using existing literature in argumentation and pragmatics and the research question on how "stance" helps in filling the gap between a premise and a conclusion is answered.

Chapter 4

Implicit and Explicit Opinions

In Chapter 3 I proposed an argument mining pipeline for processing natural language arguments. The first step in the argument mining pipeline is to extract argumentative opinionated texts using linguistic attributes: sentiment, stance and topic. Different examples are provided in the previous chapter that explains why these features are useful. This chapter discusses the next step in the argument mining pipeline and investigates a pattern observed in opinionated texts where certain information is not explicitly present in some opinions, yet is present in other opinions.

In doing so, we aim to answer the following research questions:

Research Question 1a: How is implicit information identified in natural language arguments present within opinionated texts?

Research Question 1b: How does "stance" in opinions help as a means of filling the gap between a premise and a conclusion?

Stance detection in NLP aims to classify texts based on the standpoint taken by the user, that can either be in favour or against a given topic. However, current state-of-the-art stance and sentiment detection methods do not help in understanding why certain information is left out under the assumption that the audience will still interpret the text correctly, and texts with implicit information give no clues as to how these differ from those texts in which the information is explicitly present. This gap in understanding can be overcome by exploiting concepts drawn from Freeman's argument model [29]. A different notion of stance as described in linguistics [27] refers to the expression of the user's attitude or judgement towards the standpoint taken in the content. I use this definition to classify opinions with a stance as implicit or explicit opinions.

To answer the first part of the research question, I investigate methods for automatically classifying opinions as implicit or explicit. Three different features are useful in general for, capturing linguistic properties: (1)*surface-based*, (2)*embedding-based* and (3)*hybrid*. For this purpose, I develop a corpus using opinions extracted from a set of hotel reviews. Human annotators were asked to manually annotate these opinions as explicit or implicit. An advantage of using reviews is the availability of a large dataset of online reviews with the sentiment and aspects annotated manually. Another advantage of these reviews is the presence of the overall star rating that serves as a form of conclusion for the identified arguments and enthymemes. A set of guidelines is proposed for this task. During the annotation process, I noticed that annotating a large corpus of opinions is a tedious and time-consuming process. Prior work [120] also discusses this issue. Hence, I consider different semi-supervised and weakly supervised approaches for automatically labelling larger datasets. Further, these datasets are investigated on their capability for modelling deep learning models.

The structural properties of these implicit and explicit opinions make them resemble arguments and enthymemes respectively, where an enthymeme is an argument with some information missing. The second part of the research question is answered by considering Freeman's [29] argumentation model that does not model the premise-conclusion argument structure as a logical syllogism and the logical reasoning connecting the premise with the conclusion is ignored. Instead, Freeman's argument structures relate premises with a premise-conclusion model for filling the gap between the premise and conclusion in the model. In the final section of this chapter, I discuss how these implicit and explicit opinions are interpreted as arguments and enthymemes, and how stance helps as a means of filling the gap in implicit opinions using explicit opinions. This kind of reconstruction to achieve a complete argument considers the monological structure of an argument but can also help in understanding the relationship among the different explicit and implicit opinions that support the same conclusion. In the next chapter, I investigate the next step of the argument mining pipeline by relating the identified explicit and implicit opinions using different relations that can help in constructing Freeman-style argument structures.

4.1 Definitions

Below, I present the definitions of explicit and implicit opinions. For the classification, an opinion with a stance is only considered such that the opinion talks about a topic (i.e an aspect/entity in the case of online reviews) and is of either positive or negative sentiment.

Explicit opinion In this opinion, the expression of the attitude of the user towards the standpoint towards a particular topic is linguistically expressed in a straightforward manner. For instance, if a user states "*I do not like the product*", the phrase *"do not like"* is an explicit expression that implies that the user is against the *product*. This is not only limited to expressing approval/disapproval but certain phrases/words can imply a stance straightforwardly with respect to the topic discussed.

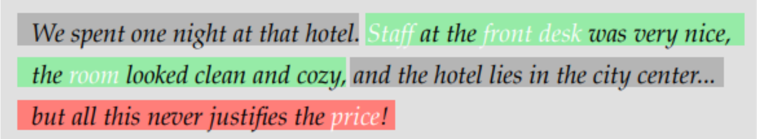
Implicit opinion In this opinion, the user expresses an attitude, emotion or uses evaluative expressions that indirectly implies the standpoint taken by the user. For example, in a hotel review, *rooms are filthy* may not directly state the stance about the *room* or *hotel* but the stance can be inferred from it.

4.2 Data Annotation

The *ArguAna* [2] corpus is used as the basis for the annotation of implicit and explicit opinions. As previously mentioned, this dataset contains 2100 manually annotated hotel reviews crawled from the TripAdvisor website. The reviews are balanced based on the overall sentiment scores and the hotels are from seven different locations. Three annotators annotated the sentiment of each sentence-level statement extracted from the hotel reviews as positive, negative or objective and the inter-rater agreement between them is computed using Cohen's κ as 0.67. Another set of two expert annotators annotated the different aspects that are present in each sentence-level statement and the inter-rater agreement between them using Cohen's κ is 0.73. The annotations present in the existing corpus are as follows:

Sentiment Each sentence-level statement is annotated as either positive, negative or objective.

Aspect The aspect/entity that is described in the statement is annotated.



We spent one night at that hotel. Staff at the front desk was very nice,
the room looked clean and cozy, and the hotel lies in the city center...
but all this never justifies the price!

Figure 4.1: An example from ArguAna corpus [2] is shown with texts highlighted as green, red and grey representing positive, negative and objective sentiment. The aspects in each sentence is highlighted as white text.

It is not an easy task for a human to annotate the opinions as implicit/explicit without any prior information. An issue that will arise is the variation in understanding the implicit/explicit definitions which may vary from person to person. To maintain consistency across the annotators, the following guidelines are provided.

Explicit opinion An opinion is explicit if:

Direct approval/disapproval contains a direct approval or disapproval. For instance, phrases such as *I like...*, *I recommend...*, *I do not like...* etc. linguistically express the stance taken by the user.

(or)

Strong intensity it contains words/phrases that express a strong form of intensity for the topic in discussion. For example, *worst staff!* expresses a stronger intensity against the aspect *staff* in comparison with *the staff were helpful.*

Implicit opinion An opinion is implicit if:

it does not contain any direct approval or disapproval

(or)

Low intensity it contains words/phrases that do not express a strong form of intensity for the topic in discussion. For example, *the staff were friendly and helped us with our baggages.*

(or)

Justifications/incidents describes an incident or a justification that indirectly implies the stance taken by the user. For example, *we were made to wait for a long time to check-in...* indicates the dissatisfaction of the user and indirectly implies the disapproval of the user.

(or)

Subjective facts it contains subjective facts. For example, *small room*, *large bed* etc. are subjective facts stated by the user that can indirectly imply the stance taken by the user. For instance, a user expresses the opinion *large bed* to indicate that it is a positive aspect of the room.

An opinion that has the stance expressed explicitly as well as implicitly is annotated as an explicit opinion.

Further, three different cues are introduced that can help in the annotation process.

General expression cues In explicit opinions, we can find that it contains words such as *great*, *recommend*, *worst* etc. that can be considered as general expression cues.

Specific expression cues Statements that indirectly implies the stance taken where the given reason is specific to a domain and hence, varies from domain to domain. For example, *lightweight laptop* is a positive opinion about the laptop whereas *lightweight story* is a negative opinion about the book.

Event-based cues These describe an incident that a user expresses to show indirect approval/disapproval.

The above guidelines were used for annotating opinions in hotel reviews and hence contain examples that are relevant to this domain. However, these guidelines are not limited to the hotel domain but can be reused for other domains in online reviews. For example, in a hotel review, *small room* can express a subjective fact to indicate that it is a negative aspect of the room while a laptop review may contain a subjective fact such as *small battery* to indicate that it is a positive aspect of the laptop.

In the first stage, a single annotator is asked to annotate 1861 positive/negative statements as implicit/explicit. In total, this annotated dataset contains 475 explicit arguments with the remainder being implicit arguments. This is highly imbalanced and hence any classifier learning this data is biased towards the majority class. I use the

Explicit opinions
“worst **hotel** ever!!!”
“just spent 3 nights at this hotel 5th march 04 -8th march 04. the **location** is excellent and the **hotel** is very grand. ”
“the **prices** are very high, even for a 5 star hotel.”
“not the **service** we expected ”
“**Parking** was expensive at $35 per night (2003).”
Implicit opinions
“during the rest of my stay i also noted peeling wallpaper in some areas and in others the walls were covered with pencil scribbles - the **room** was better than the first but was still pretty tired looking.”
“the **bathroom** is small and outdated.”
“Paying this sort of money, I expected, rightly or wrongly so, to have some sort of standard of **service**”
“Upon our return we were told a table was not ready and that we should go up to the bar and they would let us know when a table was ready” (*aspect ’service’ is implicitly implied*)
“initially, a new **receptionist** mistakenly gave us a smoking room but the very capable and pleasant assistant general **manager** laura rectified this problem the next day.”

Table 4.1: Some examples of explicit and implicit opinions. Bold text represents the aspect(s) present in the opinions.

1-Nearest Neighbour classifier to undersample the dataset [121, 122]. The entire set of 475 explicit opinions is taken as the training data and the label of a randomly selected implicit opinion is predicted. If the predicted label is incorrect, the implicit opinion is updated to the training data and the process is repeated until the remaining implicit opinions are predicted. This provides a *undersampled* dataset containing 475 explicit opinions and 769 implicit opinions.

In the second stage, two expert annotators from computer science background are asked to manually annotate the opinions present in the undersampled dataset as being implicit/explicit. The inter-rater agreement between the two annotators was computed using Cohen’s κ as 0.71. According to Cohen, an agreement in the range of 0.61-0.80 is considered as a substantial agreement. I will use the undersampled dataset for the experiments carried out. A selected set of examples are present in Table. 4.1. [1]

4.3 Data Analysis

The undersampled dataset is analysed based on the aspects present and the different guidelines given. In short, I name the guidelines as follows:

- (1) Direct approval/disapproval

[1] The corpus is publicly available at goo.gl/vkfNkm

Explicit opinion		Implicit opinion		
Direct approval/disapproval	Strong intensity	Low intensity	Justifications	Personal facts
212	282	98	30	78

Table 4.2: The number of opinions present in each category is reported.

- (2) Strong intensity
- (3) Low intensity
- (4) Justifications/Incidents
- (5) Personal facts

In Table. 4.2, I report the number of explicit opinions satisfying (1) and (2) and the number of implicit opinions satisfying (3), (4) and (5) present in randomly selected 495 explicit opinions and 206 implicit opinions from the undersampled dataset.

4.4 Features for supervised approach

I propose a supervised approach for automatically classifying opinions as implicit/explicit with the help of the undersampled dataset as the training set. Three different methods are explored and these represent the different features of the classifier. The methods are described below.

4.4.1 Surface-level method

In this method, different surface-level linguistic features present in a opinion are considered. Features are described below.

1. **Unigrams (Uni) and Bigrams (Bi)** Each word present in an opinion is a unigram and each consecutive pair of words in an opinion is a bigram.
2. **Part of Speech Tags (PoS)** Each word can be tagged with its corresponding part of speech tags. Common part of speech tags (Noun, Verb and Adjective) are considered.
3. **SentiWordNet scores (Senti)** The SentiWordNet lexical resource is used to assign three different sentiment scores (positive, negative and objective) for a given word. For each opinion S, where $\mathrm{S} = \{\mathrm{s}_1, ..., \mathrm{s}_m\}$ is the set of words present in S, the score is computed as:

$$\mathcal{P}(\mathrm{S}) = \frac{1}{|\mathrm{S}|} \sum_{i=1}^{m} \mathrm{pos}(\mathrm{s}_i) - \mathrm{neg}(\mathrm{s}_i) \tag{4.1}$$

such that

$$\mathrm{pos}(\mathrm{s}_i) + \mathrm{neg}(\mathrm{s}_i) + \mathrm{obj}(\mathrm{s}_i) = 1.0 \tag{4.2}$$

4. **Noun-Adjective patterns (Noun-Adj)** Let us consider $\mathcal{N}$ to represent the list of nouns and $\mathcal{A}$ to represent the list of adjectives in an opinion. The combination of each noun with an adjective is considered as a Noun tag + Adjective tag feature.

$$\mathrm{C} = \sum_{i=1}^{k} \sum_{j=1}^{l} NN + JJ \tag{4.3}$$

where k is the total number of nouns present and l is the total number of adjectives present.

4.4.2 Embedding-based method

In this method, each word in an opinion, $s_i \in \mathcal{S}$ is represented by an embedding vector $\mathbf{s}_i$ that belongs to a k dimensional vector space, $\mathbf{s}_i \in \mathbb{R}^k$. There are pre-trained word embeddings available and I use Glove [59] for the experiments. Three different ways of representating an opinion is computed as given below.

Average The averaged embedding vector computed using the embeddings of the words present in the opinion is considered.

$$\mathbf{v} = \frac{1}{|\mathcal{S}|} \sum_{i=1}^{|\mathcal{S}|} \mathbf{s}_i \tag{4.4}$$

Sum The embedding vector computed by summation of the embeddings of the words present in the opinion is considered.

$$\mathbf{v} = \sum_{i=1}^{|\mathcal{S}|} \mathbf{s}_i \tag{4.5}$$

Single Each element present in the embedding vector of the words present in the opinion is considered.

$$\mathbf{v} = \sum_{i=1}^{|\mathcal{S}|} \sum_{j=1}^{|\mathbf{s}_i|} s_{ij} \tag{4.6}$$

4.4.3 Hybrid method

In this method, I combine the different embeddings from the embedding-based method with each of the linguistic features described in the surface-based method. Embedding vectors capture contextual information that is not obtained using linguistic features.

4.5 Experiments and Results

Three different experiments are carried out. First, baseline features (unigrams and bigrams) are used for identifying the best classifier to perform the supervised approach.

Classifer	**Explicit opinion**	**Implicit opinion**
Linear SVM	0.77	0.85
Logistic Regression	0.71	0.84
Multinomial Naive Bayes	0.62	0.83

Table 4.3: A 5-fold cross-validation is performed on the undersampled dataset using different classifiers, F1-scores are reported for the same.

Then, an evaluation of the best classifier using the linguistic features present in the surface-based method is carried out. A comparison of results achieved using the three methods as features is performed. The scikit-learn [2] package is used for the experiments.

4.5.1 Choosing the best classifier

The baseline features, unigrams and bigrams, are used as features to train three different classifiers: (1) Linear SVM, (2) Logistic Regression and (3) Multinomial Naive Bayes. A 5-fold cross-validation on the undersampled dataset is carried out and the F1-scores are reported in Table. 4.3. The F1-score is computed as follows:

$$\mathrm{F}_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4.7}$$

The best classification is obtained using a linear SVM with the regularisation parameter value C = 10. This value is obtained using Scikit-learn GridSearchCV function that does an evaluative search on the dataset. Different values of C ranging from 0.001 to 10000.0 were tested by obtaining the F1-scores of explicit and implicit opinions using 5-fold cross-validation on the development dataset. Figure. 4.2 plots the F1-scores against the different values of C parameter ranging from 0.001 to 20.0. The scores remain constant for values greater than 10.0.

[2] http://scikit-learn.org

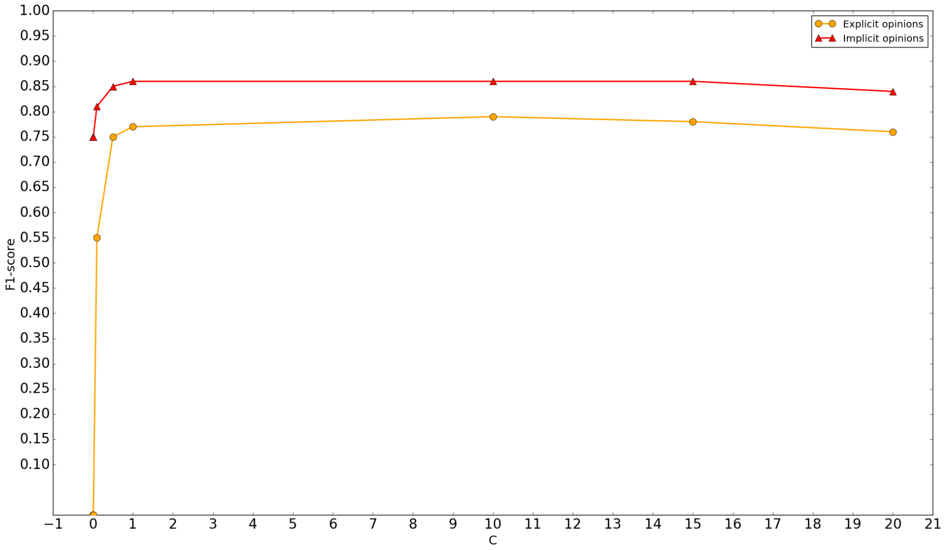
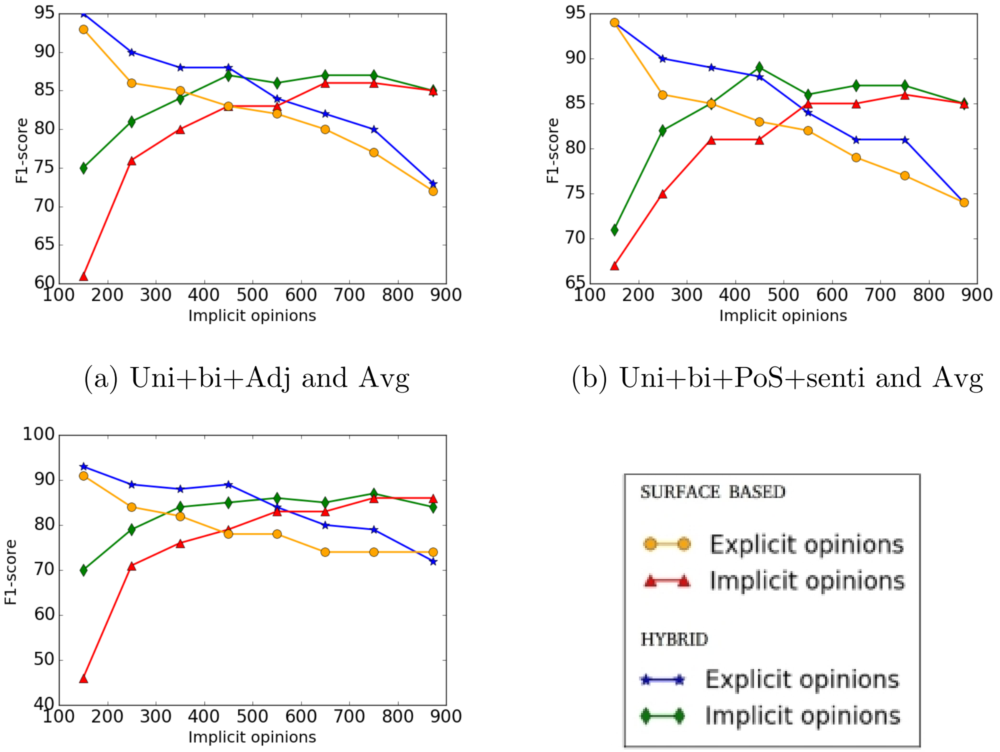


Figure 4.2: F1-scores of a 5-fold cross validation experiment with varying values of C parameter. The F1-scores of explicit opinions (circle dots) and implicit opinions (triangle dots) are plotted against different values of C.

4.5.2 Features evaluation

Features	**Explicit opinions**	**Implicit opinions**
Surface-based		
Uni	0.75	0.83
Uni + Bi	0.76	0.84
PoS	0.59	0.77
Senti	0.02	0.75
Adj-Noun	0.24	0.73
Uni + Bi + PoS	0.75	0.83
Uni + Bi + Senti	0.79	0.86
Uni + Bi + Adj-Noun	0.75	0.83
Uni + Bi + PoS + Senti	0.73	0.82
Uni + Bi + PoS + Adj-Noun	0.74	0.82
Uni + Bi + Senti + Adj-Noun	0.74	0.82
Embedding-based		
Average	0.64	0.81
Sum	0.61	0.81
Single	0.61	0.80

Table 4.4: Cross-validation results of experiments performed using different features of *surface-based* and *embedding-based* methods on the undersampled dataset containing 495 explicit opinions and 749 implicit opinions. F1-scores are reported.



(a) Uni+bi+Adj and Avg

(b) Uni+bi+PoS+senti and Avg

(c) Uni+bi+Adj-Noun and Avg

Figure 4.3: Cross-validation experiments performed using surface-based features as well as hybrid-based features for different sets containing 495 explicit opinions and varying size of implicit opinions. The F1 scores are plotted against the varying implicit opinions size respectively for both the surface-based and hybrid-based methods. Three different surface-based method features using Unigrams, bigrams, PoS tags, Sentiwordnet scores and Adj-Noun pairs count are tested. In the hybrid method, we combine these three features with the average embedding-based method. Each F1-score is plotted with the corresponding marker as shown in the figure.

In this section, a 5-fold cross-validation experiment using a linear SVM, which gives the best performance on the development dataset, is performed with features present in the following methods: (1) *surface-based* and (2) *embedding-based*. In the experiment carried out on the development data, it is found that the linear SVM performs better than an RBF-SVM. Table.4.4 contains the results of the experiment with *Average* as the best *embedding-based* method and, this is used for the *hybrid* method in combination with different features of the *surface-based* method. The hybrid method is evaluated against the different features of the surface based method by performing five-fold cross-validation experiments on different sets of opinions containing the 495 explicit opinions with varying numbers of implicit opinions. Figure 4.3 represents a detailed visualisation of the different F1-scores for each of the varying sizes of the set of implicit opinions. From the figure, it is also evident that the results improve in the case of the hybrid based method and hence features captured by the embeddings are useful in improving the overall performance.

4.5.3 PCA Visualisation

The *Average* embedding based features is studied by visualising the explicit and implicit opinions using Principal Component Analysis using the scikit-learn package. This is carried out to understand whether the average embedding vectors as sentence representations is able to distinguish between the two types of opinions. Figure. 4.4 presents the visualisation where the explicit opinions are more scattered than the implicit opinions. This is mainly because explicit opinions are mostly shorter statements such as "great hotel!", "the location is perfect".

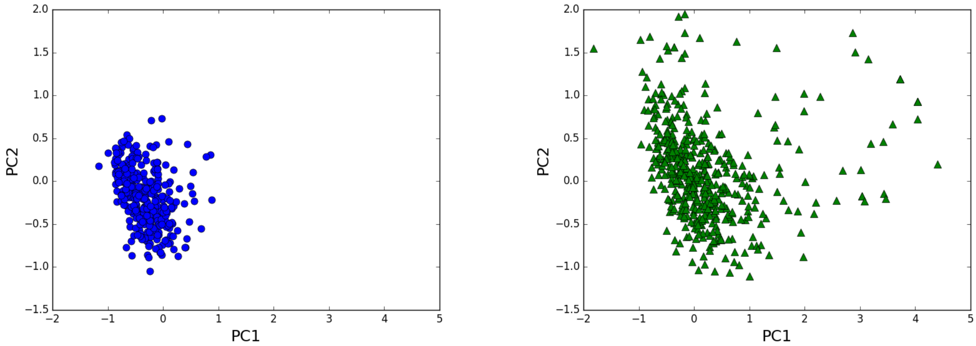


FIGURE 4.4: Visualisation of explicit and implicit opinions using the first two principal components. Here, the average embedding based method is used for features. Green dots (right) represent explicit opinions and blue dots (left) represent implicit opinions respectively. PC1 and PC2 represent the two principal components.

4.5.4 Error Analysis

The results so far suggest that embedding-based features help in distinguishing explicit and implicit opinions. To gain additional insights as to whether embedding-based features are able to capture additional contextual information that is not identified by the surface-based features, further experiments are carried out. An error analysis is performed using randomly selected 94 opinions that are extracted from 14 different hotel reviews. The experiment is performed as follows: Opinions present in each review are considered as a test set and the rest of the opinions are considered as the training set and, this is carried for each of the 14 different hotel data. The results are present with the following information as given in Table. 4.5. Table. 4.6 contains the results.

The results are presented in Table. 4.6 with respect to the column corresponding to E_cH_c and the column corresponding to S_cH_c. It is observed that for correctly predicted opinions using the hybrid method, the performance is better with the embedding-based method than that obtained with the surface-based method. To illustrate this, suppose we consider a particular feature as the unigrams and bigrams, the number of correct explicit opinions in E_cH_c and S_cH_c are 22 and 17 respectively. Clearly, the former hybrid method outperforms the latter. Again, comparing the results present in columns corresponding to S_cH_c and $S_{ic}H_c$ with those present in columns corresponding to E_cH_c and $E_{ic}H_c$, it is observed that the incorrectly predicted opinions by the surface-based features as well as the embedding-based features affect the performance of the classifier and hence, combining these two features is a better way than using them separately. Overall, the embedding-based features are able to capture the features of explicit opinions better than the features present in the surface-based method.

Label	**Description**
S_cH_c	Number of opinions correctly predicted using *surface-based* and the *hybrid* method.
S_cH_{ic}	Number of opinions correctly predicted using *surface-based* method and incorrectly predicted using the *hybrid* method.
$S_{ic}H_c$	Number of opinions incorrectly predicted using *surface-based* method and correctly predicted using the *hybrid* method.
$S_{ic}H_{ic}$	Number of opinions incorrectly predicted using *surface-based* and the *hybrid* method.
E_cH_c	Number of opinions correctly predicted using *embedding-based* and the *hybrid* method.
E_cH_{ic}	Number of opinions correctly predicted using *embedding-based* method and incorrectly predicted using the *hybrid* method.
$E_{ic}H_c$	Number of opinions incorrectly predicted using *embedding-based* method and correctly predicted using the *hybrid* method.
$E_{ic}H_{ic}$	Number of opinions incorrectly predicted using *embedding-based* and the *hybrid* method.

Table 4.5: The labels represented in Table. 4.6 is described here.

	Type	S_cH_c	S_cH_{ic}	$S_{ic}H_c$	$S_{ic}H_{ic}$	E_cH_c	E_cH_{ic}	$E_{ic}H_c$	$E_{ic}H_{ic}$
Uni+bi	Exp	17	0	4	3	22	1	2	3
	Imp	41	13	2	4	46	0	1	8
Uni+bi+pos	Exp	21	1	2	5	21	3	2	3
	Imp	46	13	1	5	48	8	3	6
Uni+bi+senti	Exp	18	0	3	3	23	1	3	2
	Imp	43	11	2	4	46	10	1	8
Uni+bi+adj-noun	Exp	20	2	3	0	21	3	2	3
	Imp	48	9	2	5	49	7	4	5

Table 4.6: Error analysis of 94 opinions from 14 reviews. Opinions in each review considered as the test set and the remaining as the training set. Error analysis was produced based on the results of each test set or each review. S represents the surface-based method, E represents the average embedding-based method and H represents the hybrid method. Subscripts c and ic indicate the number of correct and incorrect opinions. Type refers to the implicit/explicit opinion classification where exp indicates explicit and imp indicates implicit.

4.6 Automatically Labelled Dataset

In the previous section, a detailed evaluation of the different ways of automatically classifying opinions as explicit and implicit is presented. However, a drawback of the work so far is the lack of a large annotated corpus. Asking a human to annotate a huge corpus is a tedious and time-consuming task and in order to avoid this and still achieve a large corpus, I experiment using two different approaches based on semi-supervised and weakly supervised learning for labelling a large unlabeled dataset. Despite the noise present in the automatically labelled dataset, the experiments carried out empirically show that these datasets are useful for modelling deep learning models. In the weakly-supervised approach, the annotated opinions in the undersampled dataset are divided into different training sets. These different sets are used to train the SVM-based classifier using the *hybrid* method for automatically labelling unannotated opinions. These unannotated opinions are labelled based on different voting criteria, which is used to predict the final output based on certain conditions. I represent the different conditions as *Fully-Strict*, *Partially-Strict* and *No-Strict*. In the semi-supervised approach, there are two ways in which the SVM-based classifier, again using the *hybrid* method is trained: (1) using only a portion of the annotated implicit/explicit opinions or (2) using the entire data. The predicted unannotated opinion with the highest confidence obtained from the resulting classifier is appended to the training data and the process is repeated for m iterations or until all the opinions are predicted.

The different sets of automatically labelled opinions using the above approaches are used to train a Long Short-Term Memory (LSTM) [123] model and tested on the undersampled dataset. The following subsections explain the different approaches and the experiments carried out.

4.6.1 Weakly supervised approach

In this approach, I use a method that is similar to bagging [124]. It is a method where multiple classifiers are trained on randomly selected subsets of training data. This type of method can avoid overfitting. As an initial step, three different training sets T_1, T_2 and T_3 are randomly selected from the undersampled dataset. These three training sets are fed into the SVM-based classifier using unigrams, bigrams, Noun-Adjective pattern and the Average-based embedding-based method as features. Then, 4931 unannotated opinions are automatically annotated using the three resulting SVM-based classifiers. These automatically labelled opinions are used to train an LSTM classifier in two different ways that are described below.

Average-Based Three different SVM-based classifiers are trained using each training set T_1, T_2 and T_3 for automatically labelling the unlabelled opinions that gives us the corresponding annotated opinion sets U_1, U_2 and U_3. These three newly annotated opinion sets are used for training three different LSTM models and tested on the undersampled data. The averaged performance across the three LSTMs is considered as the final output.

Voting-Based Similar to the above approach, three different SVM classifiers are trained using T_1, T_2 and T_3 for automatically labelling unlabelled opinions that gives us the corresponding annotated opinion sets U_1, U_2 and U_3. Similar to the approach in [125], I combine the opinions in U_1,U_2 and U_3 into a single set, denoted by U_F, using the following voting criteria:

Fully-Strict If the same stance label is predicted by all three classifiers, the opinion is included in U_F.

Partially-Strict If the opinion is predicted as explicit by all the three SVM classifiers or if it is predicted as implicit by at least two of the SVM classifiers, then the opinion is included in U_F.

No-Strict If the opinion is predicted as implicit by at least one of the classifiers, then it is included in U_F as implicit, otherwise it is included as explicit.

An LSTM classifier is trained on the final dataset U_F and tested on the undersampled dataset. It has to be noted that as we move from Fully-strict → Partially-Strict → No-Strict, it relaxes the requirement on the inclusion of an opinion in U_F such that the number of opinions in the training data increases.

4.6.2 Semi-supervised approach

Two different approaches based on semi-supervised learning are carried out and these are described below.

Self-training method In this method, an SVM classifier is trained using D, which is the labelled data to annotate the unlabelled data U. Those opinions in U with the highest probability are added to D and this process is repeated for m times.

Reserved method This is based on existing work [126], where given training data D, a portion of it represented as R is reserved, and the remaining $D - R$ is used for training the SVM. The opinions predicted with the highest probability from U and those predicted with the lowest probability from R with the correct label are added to the training dataset. This process is repeated for m times. To do this, the undersampled data is divided as follows: 222 explicit opinions and 287 implicit opinions are randomly considered as the training data and the remaining 237 explicit opinions and 462 implicit opinions are considered as the reserved portion.

Again, the labelled opinions in U is used to train an LSTM model and tested on the undersampled dataset.

Dataset	Labelled Data		Average-based		Fully-Strict		Partially-Strict		No-Strict	
	Exp	Imp	Size	Acc	Size	Acc	Size	Acc	Size	Acc
D1	100	749	4931	73.95	4376	72.99	4541	75.56	4931	67.76
D2	200	749	4931	79.5	4310	75.64	4575	82.07	4931	71.66
D3	300	749	4931	80.99	4427	79.50	4655	83.36	4931	73.71
D4	400	749	4931	81.50	4541	78.13	4726	84.08	4931	76.36
D5	495	100	4931	76.41	3411	76.20	4113	75.32	4931	82.23
D6	495	200	4931	81.72	3742	83.52	4276	80.30	4931	83.19
D7	495	300	4931	83.01	4054	83.36	4409	83.44	4931	79.90
D8	495	400	4931	82.42	4054	83.60	4498	84.08	4931	82.31
D9	495	500	4931	83.54	4501	83.44	4762	84.00	4931	82.63
D10	495	600	4931	83.75	4484	83.52	4762	83.52	4931	82.39
D11	495	700	4931	82.15	4678	83.19	4797	84.00	4931	82.55

Table 4.7: Datasets vary in the number of explicit and implicit opinions that are randomly sampled from the labelled data to be trained by the SVM classifier. For each of the weakly supervised approach, we give *size*, the number of the predicted labels that are used to train an LSTM-based model. This model was then tested on the entire labelled data, and the accuracy of this LSTM model is reported.

4.6.3 Experiment and Results

The LSTM model is implemented using Keras[3] with an embedding layer using pre-trained 300-dimensional GloVe embeddings, followed by an LSTM layer of size 100 with a dropout rate of 0.5 and a sigmoid output layer. The input length of the opinions is padded to 50. Parameter optimisation is done using Adam [127]. The number of iterations m ranges from 1 to 25 for the semi-supervised approaches. In Table. 5.6, the results of labelled datasets using the weakly-supervised approaches are reported with *Size* representing the number of automatically labelled unannotated data. Further, the corresponding columns *Exp* and *Imp* report the number of opinions used from the undersampled dataset for training the SVM classifier. The accuracy of the LSTM model trained on the automatically labelled data for predicting the undersampled dataset is reported in the corresponding column *Acc*.

In Table. 5.6, it is observed that by comparing the accuracy (*Acc*) with the corresponding columns denoting the varying size of the explicit (*Exp*) and implicit (*Imp*) opinions, the largest set of explicit opinions used for training the initial SVM classifier produces automatically labelled data that when used to train on an LSTM, gives the best performance on the undersampled data. Overall, the best performance is achieved using the entire undersampled data for training the SVM classifier and the *Partially-Strict* voting based method and this gives an accuracy of 0.84.

[3] `https://keras.io/`

Iterations	Self-training		Reserved	
	Size	Accuracy	Size	Accuracy
1	22	49.43	511	67.68
5	2110	80.86	1717	68.24
10	2574	81.83	2194	70.25
15	3600	82.71	3152	70.98
20	3613	82.71	3708	68.81
25	4931	82.71	4931	64.22

Table 4.8: Accuracy of the LSTM model on annotated data using a set of automatically labelled unannotated opinions of *Size*.

The results obtained using the self-training and the reserved method are present in Table. 4.8 and contain details on the different sizes of the labelled unannotated dataset for each iteration showing how many opinions are added to the training data. As the size increases, the accuracy of the LSTM model for predicting the labels of the undersampled dataset improves. But, the performance decreases after 20 iterations in the case of the reserved method, as a result of the addition of less-reliable examples to the training data. A comparison of the two different methods shows that the best performance is given by the self-training method and hence, using the lowest confidence data for training is not useful for the classification task.

4.7 Argument-based analysis

In this section, I explain the interpretation of the classified implicit and explicit opinions as enthymemes and arguments. An argument requires at least a premise to be related to a conclusion. In this work, I assume that the conclusion is provided with the knowledge that a review with an overall star rating of 1.0 or 2.0 has a conclusion "*The reviewer does not like the hotel*" and a review with an overall star rating of 4.0 or 5.0 has a conclusion "*The reviewer likes the hotel*". In certain cases, the explicit opinions serve as the conclusion itself, which I do not focus on in this thesis. Assuming that a review is made up of a set of explicit and implicit opinions and the conclusion, we observe a pattern between implicit and explicit opinions. The stance or expression of attitude is not expressed linguistically in implicit opinions and this missing information is otherwise expressed in the explicit opinions. There are several definitions of what constitutes an enthymeme: (1) where it is a logical syllogism with the major premise missing and (2) where it is based on signs.

A famous example that is related to the interpretation based on the logical syllogism is as follows:

Major premise All men are mortal (unstated)

Minor premise Socrates is a man (stated)

Conclusion Socrates is mortal

In the case of enthymemes based on signs [115], it is not necessary to relate an enthymeme to a logical syllogism and instead an enthymeme is interpreted in terms of the signs that have been stated. For example, "*He is ill, since he has a fever*" is based on the sign or fact that fever causes illness. It is not always possible to reconstruct natural language arguments as logical syllogisms as they are not written as such and it is easier, looking at the missing information in these implicit opinions, to see the implicit opinions as enthymemes based on signs. Here, I refer to the unexpressed stance as the sign that is implied from the content present in these implicit opinions. Such an interpretation also leads us to a way to recreate the complete argument from the enthymeme by combining it with a related explicit opinion.

Let us look at an example below:

Explicit opinion room was great

Implicit opinion rooms had plenty of room and nice and quiet (no noise from the hallway hardwood floors as suggested by some - all carpeted)

Here, the implicit opinion can be rewritten as follows: "*the room was great*" since "*rooms had plenty of room and nice and quiet (no noise from hallway hardwood floors as suggested by some - all carpeted)*" where the implicit opinion is a sign of what is implied in the explicit opinion. This is generally true since the facts present in the implicit opinion are good aspects related to a room.

The idea of reconstructing these sign-based enthymemes and evaluating them is a subjective task and hence using Freeman's [29] argument structures can help in justifying the reconstruction. Freeman considers relating different premises as filling the gap between a premise and a conclusion. He explains that an additional premise that is used to fill the gap is either an implicit warrant or an unexpressed premise. By looking at the examples, I suggest that, given a conclusion that "*the reviewer likes the hotel*" we are able to construct an argument using the two opinions as follows: "*the reviewer likes the hotel*" (because) "*the room was great*" and "*the reviewer likes the hotel*" (because) "*rooms had plenty of room and nice and quiet (no noise from hallway hardwood floors as suggested by some - all carpeted)"* and that these two actually mean the same or one can be inferred from the other. In either of these cases, we find that the missing information in implicit opinions is otherwise present in explicit opinions.

In the next chapter, I explain the different types of relations that occur between explicit and implicit opinions. In particular, the relation between an implicit opinion and an explicit opinion can have two types of relations – (1) support or inference relation where one argument infers the other and (2) rephrase relation where two arguments that are not syntactically similar are considered similar, if one argument can replace the other and still preserves the meaning.

4.8 Conclusion

In this chapter, I begin with the first step of an argument mining pipeline where opinions are considered as argumentative using three linguistic features: stance, sentiment and topic. The next step of the argument mining pipeline is investigated for classifying the opinions as explicit or implicit. In addressing this, the following research question is answered:

Research Question 1a: How is implicit information identified in natural language arguments present within opinionated texts?

I investigated a particular domain of online reviews in which, the opinions containing a stance is classified as implicit or explicit based on whether the stance expressed is missing. Opinions are automatically classified using a supervised approach with a linear SVM-based classifier. For this task, 1244 opinions are manually annotated as implicit or explicit. Different sets of features: (1) *surface-based*, (2) *embedding-based* and (3) *hybrid* method are explored. A five-fold cross-validation experiment on the dataset of 1244 manually annotated opinions is carried out with the different features for varying sizes of implicit opinions. The best performance gives an F1-score of 0.88 and 0.86 for explicit and implicit opinions respectively using surface based features, unigrams and bigrams, in combination with Average based embedding vector. The features that are captured by the embedding vectors are analysed using principal component analysis and an error analysis is performed on the results obtained using the three different feature sets. The results show that the features captured by the embedding vectors are useful for identifying implicit and explicit opinions.

While the results look promising, the annotated dataset is small and cannot be useful for training deep learning models as a small dataset result in overfitting. To overcome this problem, different approaches based on weakly-supervised methods and semi-supervised methods are considered for automatically labelling a large unlabelled dataset using the annotated dataset. Further, these automatically created datasets are learned using an LSTM classifier and tested on the annotated dataset using the different methods. The best performance with an accuracy of 0.84 is obtained using partially-annotated weakly supervised approach and a larger dataset of 4797 opinions classified as implicit or explicit is created.

The above experiments discuss the pattern of information present in opinionated texts and classify the texts as implicit or explicit. The implicit/explicit opinion classification answers the first part of the research question following which, a theoretical approach that relates explicit opinions as arguments and implicit opinions as enthymemes is discussed. In doing so, the second part of the research question is answered.

Research Question 1b: How does "stance" in opinions help as a means of filling the gap between a premise and a conclusion?

In the discussion, I explain about the different existing views on enthymemes and how Freeman's view on relating premises with a premise-conclusion model help in relating the implicit and explicit opinions. This interpretation is presented for monological argument structures and this is considered for the next step of the argument mining pipeline, where different Freeman-style argument structures are constructed using different relations. The next chapter discusses the relations among different explicit and implicit opinions supporting the same conclusion and how it helps in constructing Freeman-style arguments.

Chapter 5

Argument-based relations

This thesis explores the different steps of the argument mining pipeline described in detail in Chapter 3. The first step is to identify opinionated texts that are argumentative using linguistic properties sentiment, stance and topic. This is followed by classifying them as explicit or implicit opinions based on the stance expressed. This step is dealt with in the previous chapter where I discuss how these explicit and implicit opinions resemble arguments and enthymemes respectively. The discussion in the previous chapter is based on the monological structure of the opinions and in this chapter, I relate the explicit and implicit opinions to construct argument structures as described by Freeman [29]. There are several existing argument structures that are discussed in detail in the literature review and one of the main reasons for choosing Freeman's approach is its adaptability to natural language texts. A simple premise-conclusion structure would require a logical reason for relating the premise to the conclusion whereas a more complex structure as Toulmin's model requires a lot of components like warrant and backing that may not exist in natural language texts. However, Freeman's [29] approach does not require the premise-conclusion structure to have any logical reasoning for relating a premise to another premise or to a conclusion and since we cannot expect all natural language arguments to have some form of logical reasoning, the approach proposed by Freeman looks promising for understanding the argument structure in natural language texts. In doing so, I answer the following research question:

Research Question 2: How do "stance", "sentiment" and "topic" help in relating opinions as premises supporting a conclusion and what kind of argument structures are obtained?

Two types of relations are identified among the opinions and two types of argument structures are explored based on the relations and these are described in the following sections.

5.1 Types of relations

The two types of relation that are present between explicit and implicit opinions are explained below.

Support-based entailment relation This relation holds when a specific premise argumentatively supports as well as textually entails a generalised premise. The specific premise in such a case is the text that infers the generalised premise as the hypothesis. This relation may be present between two explicit opinions and between an implicit opinion and an explicit opinion.

Rephrase relation A rephrase relation [116] holds between two premises when one premise can argumentatively mean the same if it is replaced by the other. In particular, an implicit opinion with missing information can be rephrased by an explicit opinion, in which the information is explicitly present.

In this work, I am interested in identifying the support-based relations that can connect different opinions as premises for a given conclusion, I do not focus on the attack relation. Freeman's [29] argument structure is constructed by answering questions that an opponent may ask for a premise that a proponent raises in support of a conclusion. In this work, I consider the different ways of constructing arguments for two different conclusions – (a) the reviewer likes the product/service and (b) the reviewer does not like the product/service and according to Freeman, the questions raised in order to derive the two conclusions is actually defended by adding additional premises. Since I am interested in understanding how argument structures are constructed in support of both the conclusions (a) and (b) and not whether one of them holds true, I do not consider the attack relation. Prior work [18] has investigated on refining Freeman's model to include undercutting and rebuttal based attack relations which can be useful for future work in introducing attack relations within the argument structures.

5.2 Argument structures

Freeman [29] defines different types of argument structures that can be constructed using premises and conclusion. Among the different argument structures, I study two types of structures as follows:

Serial In a serial argument, a premise supports another premise and a series of premises in a chained fashion support a conclusion.

Linked In a linked argument, there are different premises that are not related to each other but as a group support a conclusion.

5.3 Support-based entailment relation

The different linguistic attributes sentiment, stance and specificity are useful for identifying support-based entailment in ways which are not captured by textual entailment. I present a few examples below, which are opinions extracted from hotel reviews, to illustrate the usefulness of combining the three different attributes for predicting the relation.

Sentiment In these two opinions, the sentiment is the same but there exists no support or entailment relation.

> *"not good enough for a Hotel charging these prices"*
>
> *"the problem with the hotel is the staff"*

Stance The implicit opinion supports as well as textually entails the explicit opinion but not vice versa.

> Implicit opinion: *"the staff were helpful and polite"*
>
> Explicit opinion: *"the staff was great"*

Topic The two opinions may not talk about the same topic, but there might exist a relation between the opinions based on the two topics of the opinions.

> *"the staff was great"*
>
> *"overall, great service!"*

A distant-supervision based approach for identifying text and hypothesis pairs, for which the support-based relation holds true, is carried out by proposing a manual set of rules for predicting the *support-based entailment* relation among opinions extracted from hotel reviews. A text or hypothesis is considered as a collection of premises. A premise, in this case, is considered as an atomic unit about a particular topic. It means that any text or hypothesis that talks about several topics is considered as a collection of premises. Here, the different aspects and aspect categories present in a hotel domain are considered as topics.

These rules are designed for identifying the support-based relation among opinions present in hotels but can also be reused for other domains in online reviews, if we have the knowledge base relating the different aspects and aspect categories and we are able to classify opinions as explicit and implicit opinions. The rules can be adapted for other domain areas other than online reviews if the texts can be related to entities representing aspects or aspect categories.

Before defining the rules, two pre-steps are carried out. The first step is to identify text-hypothesis pairs based on certain conditions. The first condition is to make sure that the text and hypothesis have the same sentiment. This is done to avoid conflicting relations. The second condition is that the opinions that are treated as text or hypothesis must be either implicit or explicit based on the stance expressed. The next step is to create a knowledge base with the aspects and aspect categories of the hotel domain such that one aspect is a sub-class of the other. This knowledge base is used for defining three domain-based ontology relations that are used for defining the rules. These relations are defined below.

Definition 5.1 (**Subsumption,** $\sqsubseteq_{sub}$)**.** Two premises present within an opinion, satisfy $\mathcal{P}(attr1, op1, exp) \sqsubseteq_{intrasub}$ (intra-subsumption) $\mathcal{P}(attr2, op1, exp)$ if *attr1* is a sub-class of *attr2*.
Two premises present in two different opinions satisfy
$\mathcal{P}(attr1, op1, exp) \sqsubseteq_{intersub}$ (inter-subsumption) $\mathcal{P}(attr2, op2, exp)$ if *attr1* is a sub-class of *attr2*.

Definition 5.2 (**Inclusion,** $\sqsubseteq_{inc}$)**.** Two premises, one present in an implicit opinion and the other present present in an explicit opinion satisfy $\mathcal{P}(attr1, op1, imp) \sqsubseteq_{inc}$ (is-inclusive of) $\mathcal{P}(attr2, op2, imp)$ such that *attr1* and *attr2* are the same.

Definition 5.3 (**Equivalence,** $\equiv$)**.** $\mathcal{P}(attr1, op1, exp) \equiv$ (equivalent) $\mathcal{P}(attr2, op2, exp)$ if *attr1* and *attr2* are the same. $\mathcal{P}(attr1, op1, imp) \equiv$ (equivalent) $\mathcal{P}(attr2, op2, imp)$ if *attr1* and *attr2* are the same.

5.3.1 Support-based Entailment Rules (SER)

Let us consider the following example.

"and the ***service*** *from the* ***staff*** *was extremely poor"*

This contains two premises, one about the *service* and the other about the *staff*. These premises are not decomposed based on the linguistic structure of the opinion and instead used for identifying text-hypothesis pairs with the support-based entailment relation. The support-based entailment (SER) rules are useful for creating datasets containing text-hypotheses pairs and using the relation, these can form argument structures.

A simple structure is of the form $(implicit_1, explicit_1, explicit_2)$ and the following relations can exist:

- if there exist two premises, one in $implicit_1$ and the other in $explicit_1$ and these are about the same aspect, then there is an inclusion relation between them
- if there exist two different premises belonging to $explicit_1$ or $explicit_2$, there is an intra-subsumption relation between the premises
- if there exist two premises, one in $explicit_1$ and another in $explicit_2$, there is an inter-subsumption or inclusion relation, depending on the aspects present.

I begin with the conditions on which the three ontology-based relations exist — all these relations require two premises. Hence, for every opinion, regardless of whether it is a text or a hypothesis, the rules are designed such that at most two premises are considered at a time and whether the two premises are related or not. To illustrate this, let us consider an example below.

Opinion 1 *the* ***hotel*** *was exceptionally clean, the* service *was very friendly at all times and nothing seemed to be too much and the* location *is quiet and peaceful...*

Opinion 2 *this is very nice* ***hotel*** *that exceeded our expectations*

There are three premises present in Opinion 1: $\mathcal{P}(hotel, Op1, imp)$, $\mathcal{P}(service, Op1, imp)$ and $\mathcal{P}(location, Op2, imp)$ and one premise in Opinion 2, $\mathcal{P}(hotel, Op2, exp)$.

Given the initial condition, which requires us to consider atmost two premises at a time, the following are the different possible combinations of premises: $(\mathcal{P}(hotel, Op1, imp)$, $\mathcal{P}(service, Op1, imp))$, $(\mathcal{P}(hotel, Op1, imp)$,$\mathcal{P}(location, Op2, imp))$, $(\mathcal{P}(service, Op1, imp)$, $\mathcal{P}(location, Op2, imp)$, $(\mathcal{P}(hotel, Op2, exp)$,$\mathcal{P}(hotel, Op1, imp))$, $(\mathcal{P}(hotel, Op2, exp)$, $\mathcal{P}(service, Op1, imp))$ and $(\mathcal{P}(hotel, Op2, exp)$,$\mathcal{P}(location, Op2, imp))$. Among these, it is evident that there exists an inter-subsumption relation in $(\mathcal{P}(hotel, Op2, exp)$, $\mathcal{P}(hotel, Op1, imp))$. If an opinion has more than one premise, it means that, rules written for a single premise are not to be considered. For instance, in the above example, Opinion 1 cannot be considered for rules based on a single premise.

The support-based entailment relation is considered to hold for two opinions if at least one of the rules is satisfied by those opinions. This is to avoid identifying the same text-hypothesis pair using different rules. For example, consider a text containing three premises *a,b* and *c* with *a* and *b* related. If a hypothesis is matched with the text, we might find that one rule will be satisfied based on the related premises *a* and *b* while some other rule might be satisfied based on two premises that are not related (eg. *a* and *c*).

I create different sets of rules, one based on the subsumption relation and the other based on the inclusion relation. Considering a single premise or at most two premises at a time, there are nine different possible combinations based on whether there exists an inter-subsumption relation in the text/hypothesis or not. This holds for rules that are based on the subsumption relation as well as the inclusion relation. These rules are shown below:

Recalling the definition of *support-based entailment*, a specific premise supports a generalised premise. To ensure this is satisfied, certain rules are ignored. In particular, the rules relating two implicit opinions are ignored since the support-based entailment relation does not hold. The reason why two implicit opinions are not related by the support-based entailment relation is that the support-based entailment holds between a specific premise and a generalised premise. In the case of two implicit opinions, both of them are representations of some form of a specific premise. For the rules based on subsumption relation, those that look into hypothesis containing non-related premises are ignored. This means we have only six different combinations to deal with. The inter-subsumption relation cannot exist for implicit opinions as texts and those combinations are ruled out. In total, there are six different rules based on each of inclusion and subsumption and they are present in Table. 5.1.

The prediction process is as follows:

1. IF two opinions are explicit with the same sentiment, apply the rules based on the subsumption relation.
   (a) IF premises in text and hypothesis are related using intra-subsumption, apply the corresponding rules if any.
   (b) ELSE apply the corresponding rules based on unrelated premises and single premises
2. ELSE-IF an implicit opinion and an explicit opinion have the same sentiment, apply the rules based on the inclusion relation
   (a) IF there are text and hypothesis with a single premise, apply the corresponding rules if any.
   (b) ELSE-IF there are hypotheses with related premises, apply the corresponding rules if any.
   (c) ELSE there are hypothesis and text with unrelated premises, apply the corresponding rules if any.
3. ELSE discard the text and hypothesis pair

Examples are present in Table. 5.2.

5.3.2 Support based entailment dataset

I created three different datasets, containing text-hypothesis pairs, for which the rules predict the support-based entailment relation. Together, I call them as the *SSS* datasets representing sentiment, stance and specificity respectively. Here, specificity refers to the three domain-based ontology relations that are proposed in the previous section. Opinions were extracted from an existing corpus, ArguAna, containing hotel reviews (see literature review for details). I created the knowledge base for the hotel reviews by collecting all the manually annotated aspects identified in the corpus. Some examples are (Location $\sqsubseteq_{sub}$ Hotel), (Service $\sqsubseteq_{sub}$ Hotel), (Cleanliness $\sqsubseteq_{sub}$ Hotel), (Staff $\sqsubseteq_{sub}$ Service), (Restaurant service $\sqsubseteq_{sub}$ Service) etc. It is possible to construct the knowledge base for other domains if we are able to manually identify aspects present but the challenge is whether they can be grouped together based on categories or are completely independent of each other or, in some cases, some of the aspects can be related to two or more categories.

The three datasets are created as follows:

Rule	# Aspects (Text)	#Aspects (Hypothesis)	Text	Hypothesis	Relation
Rule 1	>1	>1	$a \sqsubseteq_{intrasub} b$	$c \sqsubseteq_{intrasub} d$	$b \sqsubseteq_{intersub} d$ or $b \equiv d$ and $a \sqsubseteq_{intersub} c$ or $a \equiv c$
Rule 2	>1	1	$a \sqsubseteq_{intrasub} b$	c	$b \sqsubseteq_{intersub} c$ or $b \equiv c$
Rule 3	>1	1	a,b and not related	c	$a \sqsubseteq_{intersub} c$ and $b \sqsubseteq_{intersub} c$
Rule 4	>1	1	a,b and not related	c	$a \equiv c$ or $b \equiv c$
Rule 5	1	1	a	c	$a \sqsubseteq_{intersub} c$
Rule 6	1	1	a	c	$a \equiv c$
Rule 1	1	1	a	c	$a \sqsubseteq_{inc} b$
Rule 2	1	>1	a	$b \sqsubseteq_{intrasub} c$	$a \sqsubseteq_{inc} b$
Rule 3	>1	>1	a,b and not related	$c \sqsubseteq_{intrasub} d$	$a \sqsubseteq_{inc} c$ and $b \sqsubseteq_{inc} d$
Rule 4	>1	1	a,b and not related	c	$a \sqsubseteq_{inc} c$ or $b \sqsubseteq_{inc} c$
Rule 5	1	>1	a	b,c and not related	$a \sqsubseteq_{inc} b$ or $a \sqsubseteq_{inc} c$
Rule 6	>1	>1	a,b and not related	c,d and not related	$a \sqsubseteq_{inc} c$ or $b \sqsubseteq_{inc} d$

Table 5.1: Each proposed rule for subsumption (top) and inclusion (bottom) relation is presented. The number of aspects (premises) that must be present in text and hypothesis is given. Conditions that must hold true in text, hypothesis and between them is also given. Here, we consider a,b,c and d to represent the aspects (premises) present.

Rule	Text	Hypothesis	Relation
Rule 1	and the **service** from the **staff** was extremely poor ($staff_{text} \sqsubseteq_{intrasub} service_{text}$)	it is the worst **service** i have seen in a five star **hotel** ($service_{hyp} \sqsubseteq_{intrasub} hotel_{hyp}$)	$service_{text} \sqsubseteq_{intersub} hotel_{hyp}, staff_{text} \sqsubseteq_{intersub} service_{hyp}, service_{text} \equiv service_{hyp}$
Rule 2	**location** of the **hotel** is really well placed - you're in the middle of everything ($location_{text} \sqsubseteq_{intrasub} hotel_{text}$)	overall a very good **hotel** ($hotel_{hyp}$)	$hotel_{text} \equiv hotel_{hyp}$
Rule 3	weak **service** for very high **prices** ($service_{text}, prices_{text}$)	i would not plan to stay at this **hotel** again ($hotel_{hyp}$)	$service_{text} \sqsubseteq_{intersub} hotel_{hyp}, prices_{text} \sqsubseteq_{intersub} hotel_{hyp}$
Rule 4	weak **service** for very high **prices** ($service_{text}, prices_{text}$)	however this is probably the worst **service** we have ever experienced ($service_{hyp}$)	$service_{text} \equiv service_{hyp}$
Rule 5	great **location** ($location_{text}$)	i absolutely loved this **hotel** ($hotel_{hyp}$)	$location_{text} \sqsubseteq_{intersub} hotel_{hyp}$
Rule 6	i absolutely loved this **hotel** ($hotel_{text}$)	overall a very good **hotel** ($hotel_{hyp}$)	$hotel_{text} \sqsubseteq_{intersub} hotel_{hyp}$
Rule 1	**hotel** infrastructure is in need of serious upgrading ($hotel_{text}$)	so believe me when i say do not stay at this **hotel** ($hotel_{hyp}$)	$hotel_{text} \sqsubseteq_{inc} hotel_{hyp}$
Rule 2	the **staff** that we encountered were very friendly and helpful ($staff_{text}$)	and the **service** from the valet and front desk **staff** is very good ($staff_{hyp} \sqsubseteq_{intrasub} service_{hyp}$)	$staff_{text} \sqsubseteq_{inc} staff_{hyp}$
Rule 4	to their credit the management was more responsive and very apologetic for the condition of my **room** and the rude treatment by their **staff** ($room_{text}, staff_{text}$)	dissapointed from the **room** ($room_{hyp}$)	$room_{text} \sqsubseteq_{inc} room_{hyp}$
Rule 5	the **staff** was not friendly nor helpful ($staff_{text}$)	overall its a dark dated **hotel** let down badly by the unhelpful and rude **staff** ($hotel_{text}, staff_{hyp}$)	$staff_{text} \sqsubseteq_{inc} staff_{hyp}$

Table 5.2: Examples for different rules satisfying subsumption (top) and inclusion (bottom) relations.

Fully Annotated (FA) A balanced set of 369 reviews from 15 different hotels were extracted from the ArguAna corpus. Each review in the ArguAna corpus contains the sentiment of each sentence-level opinion and the aspects present in the sentences manually annotated through crowdsourcing. Further, the opinions are manually annotated as implicit/explicit in the undersampled dataset (Section 4.2, Chapter 4) and the dataset comprises 264 explicit opinions and 720 implicit opinions. Each of the explicit opinions was paired with each of the implicit opinions as well as with each of the other explicit opinions. These opinion pairs are considered as text-hypothesis pairs. The six subsumption SER rules are used to predict whether a given explicit opinion as text supports as well as entails an explicit opinion as a hypothesis. In total, there are 808 text-hypothesis pairs that are predicted using the subsumption rules. The six inclusion SER rules are used to predict whether a given explicit opinion as text supports as well as entails an implicit opinion as a hypothesis and these rules predicted 1412 text-hypothesis pairs to satisfy the support-based entailment relation.

Semi-Annotated (SA) A balanced set of 707 reviews from 33 different hotels were extracted from the ArguAna corpus and seen in the Fully Annotated dataset, the sentiment of each sentence-level opinion and the aspects present in the sentences are manually annotated. However, the opinions extracted from the 707 reviews are not manually annotated as being implicit or explicit. Instead, the linear SVM classifier that has been used in the previous chapter for automatically annotating opinions as explicit or implicit is used. The linear SVM classifier is trained with the undersampled dataset containing 475 opinions annotated as explicit and 1386 opinions as explicit. This classification is described in detail in Section 4.6.2 of Chapter 4 and I make use of the surface-based and embedding based features for training the classifier i.e. using the best surface-level features in combination with the Average based embedding vector. The classifier predicts 1001 opinions as explicit and 4359 opinions as implicit and similar to the steps followed for creating the Fully Annotated dataset, the explicit opinions are paired with the explicit opinions as well as implicit opinions. Both the subsumption and the inclusion rules predict 11892 text-hypothesis pairs to satisfy the support-based entailment relation.

Unannotated (UA) This dataset is created with an unbalanced set of reviews from 30 different hotels extracted from the ArguAna additional corpus and hence, does not contain any manual annotation. The main reason to create this dataset is to understand the noise in the data if we are not able to manually annotate the sentiment and aspects present in the opinions in the reviews. To address the issue of automatically predicting the sentiment of an opinion, an SVM-based classifier proposed for the ArguAna tool [89] is used for automatically predicting whether a given opinion is positive, negative or objective. All opinions that have the sentiment as objective were discarded and not used for building the dataset. The

Data	Reviews	Explicit	Implicit	Subsumption	Inclusion
FA	369	264	720	Rule 1: 14	Rule 1: 271
				Rule 2: 138	Rule 2: 25
				Rule 3: 27	Rule 3: 6
				Rule 4: 218	Rule 4: 619
				Rule 5: 193	Rule 5: 147
				Rule 6: 218	Rule 6: 344
SA	707	1001	4359	Rule 1: 92	Rule 1: 1790
				Rule 2: 566	Rule 2: 137
				Rule 3: 82	Rule 3: 55
				Rule 4: 344	Rule 4: 3418
				Rule 5: 842	Rule 5: 933
				Rule 6: 1834	Rule 6: 1799
UA	3271	564	5933	Rule 1: 34	Rule 1: 3708
				Rule 2: 467	Rule 2: 148
				Rule 3: 55	Rule 3: 33
				Rule 4: 119	Rule 4: 4726
				Rule 5: 428	Rule 5: 2189
				Rule 6: 1354	Rule 6: 3053

Table 5.3: In each dataset: total number of reviews (Rev) present, total number of explicit opinions (Exp) and implicit opinions (Imp) found and total number of TH pairs satisfying each rule in SER based on subsumption (Sub) and inclusive (Inc) relation is present.

next challenge is to identify the aspects present in the opinions and while several existing works are present on the aspect identification (which is beyond the scope of this dataset), I consider using the manually annotated aspects present in the entire ArguAna corpus as a list of aspects that are considered. Again, by using the linear-SVM classifier that I have described in Section 4.6.2 of Chapter 4, these opinions were automatically classified as implicit/explicit, which gives us 564 explicit opinions and 5933 implicit opinions. There are 16314 text-hypothesis pairs that are predicted with support-based entailment by the rules.

Table 5.3 contains a detailed description of these three datasets.

5.3.3 Experiments and Results

In this section, I explain the experiments carried out on the three different datasets. Firstly, we need to evaluate the performance of the support-based entailment rules (SER) against human annotation. For this purpose, 160 text-hypothesis pairs are randomly selected from text-hypothesis pairs that are predicted using the SER and from text-hypothesis pairs that do not satisfy the rules. These text-hypothesis pairs were manually annotated by two expert annotators from a computer science background who were not provided with any information about the rules. The inter-rater agreement computed using Cohen's κ was 0.80. According to Cohen, any value in the range of 0.61-0.80 is considered as representing a substantial agreement. Further, the performance of the SER was tested by considering: (1) the intersection of answers of the annotators as the ground truth data, for which the accuracy of the SER was 0.83 and, (2) the union of the answers of the two annotators as the ground truth data, for which the accuracy of the SER was 0.93.

Support-based entailment is a subtype of the entailment relation present in textual entailment if we are to consider the entailment relation between opinions. This does not hold true for other domains where entailment need not necessarily mean that the support relation exists. To study how reliable the SER rules are in predicting the entailment relation, I used an existing state-of-the-art textual entailment tool, the Excitement Open Platform (EOP) [128]. Given a text and a hypothesis, it predicts whether the text entails the hypothesis or not. Three different datasets were investigated for this tool namely standard RTE-3 [66], SICK [67] and EXCITEMENT [68]. For the supervised approach, four different entailment decision algorithms were investigated: MaxEntClassificationEDA, AdArteEDA, EditDistanceEDA and an alignment-based entailment decision algorithm (P1EDA). To test these algorithms, I use the Fully-Annotated dataset as this contains less noisy data. The MaxEntClassificationEDA based on the maximum entropy classifier trained with the RTE-3 dataset gives the best performance for predicting entailment relation among the text-hypothesis pairs present in the Fully-Annotated dataset with an accuracy of 89.54 % and this classifier is used for further experiments.

Different sets of experiments based on different conditions to predict the text-hypothesis pairs are carried out and evaluated against the textual entailment prediction and the accuracy is reported. These experiments are described below.

1. **Subsumption based SER** Two explicit opinions are considered as a text-hypothesis pair if the relation is predicted using subsumption based rules.

2. **Subsumption based Non-SER** Two explicit opinions are considered as a text-hypothesis pair if they do not satisfy any of the subsumption based rules.

3. **Inclusion based SER** An implicit opinion and an explicit opinion are considered as a text-hypothesis pair if the relation is predicted using inclusion based rules.

4. **Inclusion based Non-SER** An implicit opinion and an explicit opinion are considered as a text-hypothesis pair if they do not satisfy any of the inclusion based rules.

5. **SER** Text-hypothesis pairs extracted in both **Subsumption based SER** and **Inclusion based SER** are considered.

6. **Non-SER** Text-hypothesis pairs extracted in both **Subsumption based Non-SER** and **Inclusion based Non-SER** are considered.

7. **Subsumption** Each individual subsumption based rule is considered and the corresponding text-hypothesis pairs are extracted.

8. **Inclusion** Each individual inclusion based rule is considered and the corresponding text-hypothesis pairs are extracted.

9. **Implicit-Explicit Entailment** An implicit opinion and an explicit opinion, of the same sentiment, are considered as a text-hypothesis pair. This experiment is

Experiment	FA	SA	UA
SER	89.54	90.00	96.19
Non-SER	76.18	72.69	88.01
Subsumption based SER	81.63	75.82	92.11
Subsumption based Non-SER	73.91	67.93	86.21
Inclusion based SER	95.83	96.49	97.68
Inclusion based NON-SER	76.87	73.84	88.31
Implicit-Explicit Entailment	75.94	71.03	87.89
Subsumption			
-Rule 1	100.0	83.69	100.0
-Rule 2	86.95	92.40	96.14
-Rule 3	44.44	52.43	80.0
-Rule 4	89.44	93.89	99.15
-Rule 5	62.69	46.67	83.64
-Rule 6	86.69	81.35	92.17
Inclusion			
-Rule 1	92.61	93.74	94.76
-Rule 2	96.0	95.62	96.62
-Rule 3	100.0	94.59	100.0
-Rule 4	97.25	98.50	98.47
-Rule 5	89.79	92.60	95.56
-Rule 6	95.63	97.72	98.59
Random sentiment (SER)	45.62	45.31	47.98
Random sentiment (Non-SER)	38.64	36.37	44.02

Table 5.4: An experiment was run on each dataset by (a) SER — TH pairs satisfying either of the six subsumption or six inclusion rules (b) Non-SER — TH pairs that do not satisfy any of the 12 rules. (c) Subsumption and Inclusion — TH pairs satisfying each individual rule and (d) Random sentiment — assigning sentiment of opinions present in TH pairs of SER and Non-SER randomly. Accuracy is reported.

performed to understand whether textual entailment is able to capture the difference in implicit and explicit opinion.

10. **Random sentiment** Each of the implicit and explicit opinion present in SER and Non-SER is randomly assigned a sentiment and the support-based entailment relation is predicted based on this misinformation.

Before analysing the results, it is to be noted that the Semi-Annotated and Unannotated datasets are noisier than the fully annotated dataset. It means that, although the accuracy of the results present in Table. 5.4 are higher for both these datasets, the accuracy may not be an accurate form of evaluation. Hence, I discuss the results by comparing the different sets of experiments performed on the Fully-Annotated dataset.

The accuracy results of SER, Subsumption based SER and Inclusion based SER are compared and show that textual entailment is able to perform best in identifying relations that are predicted using the inclusion rules. It does show that the implicit/explicit classification of opinions based on stance in combination with surface-based features is useful for predicting entailment. Comparing the results of Subsumption based SER and Subsumption based Non-SER, we find that textual entailment is not able to distinguish between the text-hypothesis pairs with the support relation and those that do not have a support relation and, this might be because of the lack of external knowledge such as the domain-based knowledge base that is not exploited by textual entailment. The results of Inclusion-based SER is significantly better than the Inclusion-based Non-SER which may be accounted to the inclusion relation. However, textual entailment is not able to distinguish between the topics. Still, this does not show how implicit opinions as texts and explicit opinions as hypotheses works. The results of Inclusion-based SER is compared with those of Implicit-Explicit Entailment, which clearly shows that implicit/explicit classification is useful for identifying better support-based entailment relation.

I analysed the individual results of methods using the Subsumption relation and I found that the results of Rule 3 and Rule 5 have the worst performance as these two rules are heavily dependent on the domain-based knowledge base. Overall, it does show how the linguistic properties useful for identifying support relation are not captured by textual entailment and that combining the ontology-based relations and domain-based knowledge base can improve its performance.

5.4 Rephrase relation

The rephrase relation [116] was introduced in argument structures constructed from dialogues. A speaker might utter similar premises as a form of repetition and these premises or in some cases, conclusions are argumentatively the same. The argumentative meaning of a premise is preserved if it is replaced by a rephrased premise. In dialogues, the authors of [116] explain that premises that are elaborated or contain more information rephrase the less informative premise. Due to this, although the rephrase relation is closely related to paraphrasing, it is directional.

Premise A (rephraser) → Premise B (rephrasee)

Here, given a premise A that rephrases B, we refer A as the *rephraser* and B as the *rephrasee*.

Borrowing the definition of rephrase relation from [116] and using enthymemes based on signs [115] as evidence, we find that explicit opinions can rephrase implicit opinions about the same topic. This is because enthymemes based on signs and arguments are argumentatively similar but syntactically do not have a similar construct. In the opinions dataset, we find that opinions are present in monological texts of reviews that are not related to each other and there is no evidence to replace an explicit opinion by the implicit opinion. In the previous section, we find that there exists a support-based entailment relation between implicit and explicit opinions about the same topic. In this section, I introduce the existence of another kind of relation, which actually treats the two arguments as same, despite their structural properties supporting entailment. Some examples are present in Table. 5.5.

I experimented with a bipartite graph-based approach to match appropriate explicit opinions for a given implicit opinion using three different features for computing the

Implicit opinion	Explicit opinion
rooms had plenty of room and nice and quiet (no noise from the hallway hardwood floors as suggested by some - all carpeted)	room was great
we received a lukewarm welcome at check in (early evening) and a very weak offer of help with parking and our luggage	we were extremely unimpressed by the quality of service we encountered
i have been meaning to write a review on this hotel because of the fact that staying here made me dislike Barcelona (hotels really can affect your overall view of a place, unfortunately)	this hotel was just a great disappointment

Table 5.5: Implicit opinions with their corresponding explicit opinions that can rephrase it.

cost function: (1) similarity measure, (2) sentiment and (3) target. Using this approach I explored different sentence embedding representations are explored for measuring the similarity. Sentiment and target are manually present in the dataset.

5.4.1 Bipartite Opinion Matching

In the bipartite graph-based approach, rephrase relation prediction is formulated as a maximum cost K ranked bipartite-graph matching problem using a set of explicit and implicit opinions. The bipartite graph is constructed as follows. Each implicit opinion is matched with each of the explicit opinions and the cost is computed using a cost function. For every implicit opinion, the top K explicit opinions with the highest cost are considered. The cost function is computed using the three different features as follows:

$$\mathrm{C}(i,j) = \mathrm{sim}(\boldsymbol{s}_{\mathbf{i}}, \boldsymbol{s}_{\mathbf{j}}) + \mathrm{Q}(i,j) + \mathrm{R}(i,j) \tag{5.1}$$

where sim represents the similarity measure computed between two sentence embedding vectors $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, Q represents the cost value by checking whether the sentiment of the two sentences are the same or not and R represents the cost value by checking whether the target present in the two sentences are the same or not.

5.4.2 Unsupervised Sentence Embedding

The similarity between two sentences can be measured using both unsupervised and supervised sentence embedding representations. For the unsupervised methods, each word is initialised with pre-trained embedding vectors. Existing works by Arora et al. [62] and Mu et al. [129] are used to perform different steps on the initialised word embeddings to create sentence embedding vectors. There are two post-processing steps that are performed by Mu et al. [129] on pre-trained word embedding vectors. The motivation of their work is to create better word embedding representations and hence do not focus on sentence representation. They show that word embeddings are narrowly distributed in a cone and by subtracting the mean vector and applying Principal Component Analysis

(PCA), it is possible to obtain an isotropic spherical distribution. As a result, the common parts are eliminated and similar word pairs move close to each other.

The two post-processing steps that are performed on the pre-trained word embedding vectors are described next.

Diff Let us assume that we are given a set $\mathcal{V}$ (vocabulary) of words w, which are represented by a pre-trained word embedding $\boldsymbol{w}_i \in \mathbb{R}^k$ in some k dimensional vector space. The mean embedding vector, $\hat{\boldsymbol{w}}$, of all embeddings for the words in $\mathcal{V}$ is given by:

$$\hat{\boldsymbol{w}} = \frac{1}{|\mathcal{V}|} \sum_{w \in \mathcal{V}} \boldsymbol{w} \tag{5.2}$$

Using the steps in Mu et al. [129], the mean is subtracted from each word embedding to create isotropic embeddings as follows:

$$\forall_{w \in \mathcal{V}} \quad \tilde{\boldsymbol{w}} = \boldsymbol{w} - \hat{\boldsymbol{w}} \tag{5.3}$$

WordPCA The mean-subtracted word embeddings given by (5.3) for all $w \in \mathcal{V}$ are arranged as columns in a matrix $\mathbf{A} \in \mathbb{R}^{k \times |\mathcal{V}|}$, and its d principle component vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d$ are computed. Mu et al. [129] observed that the normalised variance ratio decays until some top $l \leq d$ components, and remains constant after that, and proposed to remove the top l principle components from the mean-subtracted embeddings as follows:

$$\boldsymbol{w}' = \tilde{\boldsymbol{w}} - \sum_{i=1}^{l} \left(\boldsymbol{u}_i \boldsymbol{w} \right) \boldsymbol{u}_i \tag{5.4}$$

The different methods used to represent the sentence embeddings using word embeddings are described below.

AVG One of the simplest, yet surprisingly accurate, method to represent a sentence is to compute the average of the embedding vectors of the words present in that sentence. Given a sentence $\mathcal{S}$, we first represent it using the set of words $\{w | w \in \mathcal{S}\}$. We then create its sentence embedding $\boldsymbol{s} \in \mathbb{R}^k$ as follows:

$$\boldsymbol{s} = \frac{1}{|\mathcal{S}|} \sum_{w \in \mathcal{S}} \boldsymbol{w} \tag{5.5}$$

Three different variants for sentence embeddings are possible depending on the preprocessing applied on the word embeddings used in (5.5): **AVG** (uses unprocessed word embeddings $\boldsymbol{w}$), **Diff+AVG** (uses $\tilde{\boldsymbol{w}}$) and **WordPCA+AVG** (uses $\boldsymbol{w}'$).

WEmbed Arora et al. [62] proposed a method to create sentence embeddings as the weighted-average of the word embeddings for the words in a sentence. The weight

$\psi(w)$ of a word w is computed using its occurrence probability $p(w)$ estimated from a corpus as follows:

$$\psi(w) = \frac{a}{a + p(w)}\boldsymbol{w} \tag{5.6}$$

$$\boldsymbol{s} = \frac{1}{|\mathcal{S}|}\sum_{w \in \mathcal{S}} \psi(w)\boldsymbol{w} \tag{5.7}$$

Here, a is a small constant[1]. Intuitively, frequent words such as stop words will have a smaller weight assigned to them, effectively ignoring their word embeddings when computing the sentence embeddings.

SentPCA Given a set of sentences $\mathcal{T}$, Arora et al. [62] applies PCA on the matrix that contains individual sentence embeddings as columns to compute the first principle component vector $\boldsymbol{v}$, which is subtracted from each sentence's embedding as follows:

$$\boldsymbol{s}' = \boldsymbol{s} - \boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{s} \tag{5.8}$$

These give us five sentence embedding methods (**AVG**, **Diff+AVG**, **WordPCA+AVG**, **WEmbed** and **SentPCA**). The similarity measure between an implicit and an explicit opinion in an unsupervised approach is computed using the cosine similarity between their corresponding sentence embeddings.

5.4.3 Supervised Sentence Similarity

For a supervised approach, the similarity is computed between two sentence embeddings using a training dataset. The training dataset consists of pairs of sentences that are manually rated for the degree of their semantic similarity. Given two sentences s_i, s_j, their sentence embeddings are computed, respectively $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, using one of the unsupervised sentence embedding methods described in the previous section. Then, each pair of sentences is represented using two operators: $\boldsymbol{h}_\times$ (elementwise multiplication) and $\boldsymbol{h}_-$ (elementwise absolute value of the difference). The arguments of the operators are dropped to simplify the notation.

Intuitively, $\boldsymbol{h}_\times$ captures common attributes in the two sentences, whereas $\boldsymbol{h}_-$ captures attributes unique to one of the two sentences. We then feed $\boldsymbol{h}_\times$ and $\boldsymbol{h}_-$ to a neural network containing a sigmoid ($\sigma(\cdot)$) hidden layer and a softmax ($\phi(\cdot)$) output layer parametrised by a set $\theta = \{\mathbf{W}^{(\times)}, \mathbf{W}^{(-)}, \mathbf{W}^{(p)}, \boldsymbol{b}^{(h)}, \boldsymbol{b}^{(p)}\}$ as follows:

$$\begin{aligned}
\boldsymbol{h}_\times &= \boldsymbol{s}_i \odot \boldsymbol{s}_j \\
\boldsymbol{h}_- &= |\boldsymbol{s}_i - \boldsymbol{s}_j| \\
\boldsymbol{h}_s &= \sigma\left(\mathbf{W}^{\times}\boldsymbol{h}_\times + \mathbf{W}^{(-)}\boldsymbol{h}_- + \boldsymbol{b}^{(h)}\right)
\end{aligned}$$

[1] Set to 0.001 in the experiments

$$\hat{\boldsymbol{p}}_\theta = \phi\left(\mathbf{W}^{(p)}\boldsymbol{h}_s + \boldsymbol{b}^{(p)}\right)$$

For the training dataset, I used the SICK [67] sentence similarity dataset that consists of pairs of sentences manually rated in an ordinal range from 1 to 5, where 1 represents the lowest and 5 represents the highest similarity. I denote this gold standard rating for s_i and s_j by $y(s_i, s_j) \in [1, K]$, where $K = 5$ for the SICK dataset. The class probability distribution, $\hat{\boldsymbol{p}}_\theta$ is used to compute the expected similarity rating $\hat{y}(s_i, s_j)$ between s_i and s_j as follows:

$$\hat{y}(s_i, s_j) = \boldsymbol{r}\hat{\boldsymbol{p}}_\theta \tag{5.9}$$

Here, the rating vector $\boldsymbol{r} = (1, 2, \ldots, K)$. In order to keep the expected rating to be close to the gold standard rating, following [130], a sparse target distribution $\boldsymbol{p}$ that satisfies $y = \boldsymbol{r}\boldsymbol{p}$ is defined below:

$$p_i = \begin{cases} y - \lfloor y \rfloor & \text{if } i = \lfloor y \rfloor + 1 \\ y - \lfloor y \rfloor + 1 & \text{if } i = \lfloor y \rfloor \\ 0 & \text{otherwise} \end{cases}$$

The parameters θ of the model are found by minimising the KL-divergence between $\boldsymbol{p}$ and $\hat{\boldsymbol{p}}_\theta$ subject to ℓ_2 regularisation over the entire training dataset $\mathcal{D}$ of sentence pairs as follows:

$$J(\theta) = \sum_{(s_i, s_j) \in \mathcal{D}} \mathrm{KL}\left((p^{(k)} || \hat{p}_\theta^{(k)}\right) + \frac{\lambda}{2} ||\theta||_2^2 \tag{5.10}$$

Here, $\lambda \in \mathbb{R}$ is the regularisation coefficient, set using validation data.

The cost function of the bipartite matching problem using sentence similarity can then be defined as follows.

$$\mathrm{C}(i, j) = \mathrm{sim}(\boldsymbol{w}_\mathbf{i}, \boldsymbol{w}_\mathbf{j}) \tag{5.11}$$

Here, sim is the cosine similarity between sentence embeddings in the case of the unsupervised approach and for the supervised approach is the predicted similarity rating $\hat{y}$.

5.4.4 Sentiment and topic

Sentiment and topic are two important linguistic attributes of stance-bearing opinions, and these two features are useful for maximizing the cost function. The cost function is redefined as follows:

$$\mathrm{C}(i, j) = \mathrm{sim}(\boldsymbol{s}_\mathbf{i}, \boldsymbol{s}_\mathbf{j}) + \mathrm{Q}(i, j) + \mathrm{R}(i, j) \tag{5.12}$$

In this equation, Q and R output a threshold value if both the S_i and S_j have the same sentiment and the same topic respectively.

5.4.5 Experiments and Results

In the experiments, I use pre-trained Glove embeddings [59] with 300 dimensions. Following [129], l = 2 is used for the third step of *WordPCA*. The sentiment of an opinion and the topic present in it are manually annotated. The aspects and aspect categories of the hotel domain represent the topics. The domain knowledge base created previously (section) relating the different aspects present in the hotel domain is also used. For the sentiment and topic function, the threshold values are varied from 0 to 1 on the development data and this gives a value of 0.5 such that the cost function is not biased towards the sentiment and topic information alone.

The following evaluation measures are used.

Precision@K (P@K) For every implicit opinion, the corresponding top K explicit opinions are considered.

$$\text{P@K} = \frac{1}{m}\sum_{i=1}^{m}\frac{n_i}{K} \tag{5.13}$$

where m is the total number of implicit opinions, n_i is the number of correct explicit opinions for the corresponding *i-th* implicit opinion, and K is the number of top explicit opinions that are considered.

Average precision@K (Avg P@K)

$$\text{Avg P@K} = \frac{1}{K}\sum_{i=1}^{K} P@i \tag{5.14}$$

Here, K is the number of top explicit opinions that are considered and $P@i$ represents the precision@i score.

Mean reciprocal rank (MRR)

$$\text{MRR} = \frac{1}{m} = \sum_{i=1}^{i=m}\frac{1}{R_i} \tag{5.15}$$

where m is the total number of implicit opinions and R_i is the rank of the first correct explicit opinion for the i-th implicit opinion.

Accuracy (Acc)

$$\text{Acc} = \frac{1}{m}\sum_{i=1}^{m} l \tag{5.16}$$

where $l = 1$ if at least one of the correct explicit opinions is present within the top 10 explicit opinions; otherwise 0.

Task 1: Implicit/Explicit opinion dataset

For this task, 57 implicit opinions from the undersampled dataset are chosen. For each implicit opinion, an annotator is asked to chose three appropriate explicit opinions that rephrase the corresponding implicit opinion. This gives us 56 explicit opinions. Again, for each implicit opinion, an annotator is asked to compare against the 56 explicit opinions to choose those that rephrase the corresponding implicit opinion. The number of explicit opinions that can rephrase an implicit opinion ranges from a minimum of 1 to a maximum of 13, and on an average is 6.

A bipartite graph is formed with the implicit and explicit opinions as nodes and edges from each implicit opinion to every explicit opinion. For every implicit opinion, the top K explicit opinions with the cost function score ranging from highest to lowest are considered as correctly predicted rephrasers. The cost function is computed using the different similarity measures, sentiment function and target function. The top K explicit opinions are compared against the manually chosen explicit opinions.

The results are reported in Table. 5.6. The $P@K$ for values of $K = 10$, 15 and 20 and the *Avg* $P@K$ for $K = 15$ and 20 are present in the table. The **SENTPCA** that performs well on the similarity tasks does not perform better than the simple baseline **AVG**. It is an interesting result that shows how common words ignored by the **SENTPCA** method are also important in determining the rephrase relation. The best performance is achieved using **WordPCA+AVG** as the sentence representation. Among the supervised approaches, there is not much difference between **AVG** and **WordPCA+AVG** and both these methods perform better than the rest. By comparing both the supervised and unsupervised approaches, the best performance of the unsupervised method is better than that of the supervised approach. Although the results obtained using sentiment are not better than those obtained by other methods, the best performance is achieved by combining all three features. The implicit/explicit opinion classification plays an important role in predicting the directionality of the rephrase relation, which is evaluated in the next task.

Task 2: Implicit/Explicit dataset and Citizen's corpus dataset

In this task, there are two questions that are answered:

1. How useful is implicit/explicit opinion classification for identifying the rephrase relation?

2. The rephrase relation was initially [116] proposed to identify premises present in the same dialogue such that a generalised premise is rephrased by a premise with specific and detailed information. Assuming that we are given a classification system that aims to classify premises as generalised or not, how useful are our experiments for the Citizen Dialogue corpus? The motivation to introduce the classification system is because, firstly it gives us a way to compare the results with those obtained with our dataset and, secondly it helps in analysing whether this classification system that identifies a pattern in the dialogue is useful for identifying rephrase relation in dialogic datasets? If so, does identifying patterns

Methods	**P@10**	**P@15**	**P@20**	**Avg P@15**	**Avg P@20**
UNSUPERVISED					
AVG	0.15	0.22	0.30	0.13	0.16
Diff+AVG	0.15	0.21	0.27	0.12	0.15
WordPCA+AVG	**0.17**	**0.23**	**0.30**	**0.14**	**0.17**
WEmbed	0.14	0.20	0.25	0.12	0.15
SENTPCA	0.14	0.20	0.27	0.12	0.21
SUPERVISED					
AVG	0.14	0.19	0.25	0.12	0.15
Diff+AVG	0.14	0.19	0.24	0.11	0.14
WordPCA+AVG	0.14	0.21	0.25	0.12	0.15
WEmbed	0.07	0.12	0.18	0.05	0.08
SENTPCA	0.10	0.14	0.22	0.08	0.11
Sentiment	0.08	0.14	0.17	0.06	0.13
Target	0.16	0.20	0.24	0.12	0.19
Sentiment + target	0.17	0.22	0.25	0.13	0.20
WordPCA+AVG+sentiment+target	**0.28**	**0.34**	**0.39**	**0.21**	**0.26**

Table 5.6: For a given set 57 implicit opinions and 56 explicit opinions, we compute the cosine similarity between each pair of implicit and explicit opinions using each of the methods described in Section 5.4.2. Moreover, sentiment and topic functions are computed. Precision@K with K = 10,15,20 are computed and the results are present. In addition, average Precision@K with K = 15 and 20 are computed and the results are shown.

among premises present in monological or dialogical texts help in identifying the rephrase relation?

The Citizen's dialogue corpus [116] contains pairs of premises related by the rephrase relation and the premises are considered as mere repetitions uttered by the speaker. Hence, premises that contain specific information rephrase generalised premises. I collected 64 premise pairs from this corpus for the experiment. Some examples are given below:

Example 1 **Rephraser** Where does it stand on getting the next steps approved

Rephrasee I don't have a timeframe for you, but that gives you an idea of what we're looking at

Example 2 **Rephraser** We're going to keep you informed

Rephrasee During this construction phase, we're going to be doing everything we can to keep you informed and keep you safe and keep traffic moving safely.

As discussed earlier, the implicit/explicit opinions dataset was manually created by considering explicit opinions as arguments expressing the same argument as that of implicit opinions, which are considered to be enthymemes. The missing information present in the enthymemes is explicitly present in the relevant premises. Thus, explicit opinions rephrase implicit opinions that express the same argument. An implicit opinion containing a justification or reasoning with detailed information cannot rephrase an explicit opinion as these opinions may belong to different monological texts that are unrelated to each other and there is no evidence to relate them. Hence, in the implicit/explicit opinions dataset, a generalised premise rephrases a premise with specific information.

Methods	Without Information				With Information			
	Citizen Dialogue		Implicit/Explicit		Citizen Dialogue		Implicit/Explicit	
	MRR	**Acc**	**MRR**	**Acc**	**MRR**	**Acc**	**MRR**	**Acc**
UNSUPERVISED								
AVG	0.56	0.75	0.13	0.31	0.62	0.81	0.29	0.75
Diff+AVG	0.55	0.75	0.12	0.28	0.61	0.81	0.28	0.75
WordPCA+AVG	0.59	0.80	0.07	0.24	0.64	0.86	0.25	0.82
WEmbed	0.52	0.67	0.15	0.49	0.55	0.72	0.32	0.68
SENTPCA	0.51	0.67	0.16	0.47	0.55	0.72	0.35	0.65
SUPERVISED								
AVG	0.56	0.78	0.10	0.31	0.63	0.83	0.27	0.68
Diff+AVG	0.54	0.78	0.10	0.30	0.61	0.83	0.25	0.68
WordPCA+AVG	0.57	0.76	0.06	0.24	0.63	0.80	0.26	0.74
WEmbed	0.004	0.03	0.08	0.23	0.04	0.16	0.23	0.70
SENTPCA	0.007	0.04	0.10	0.31	0.03	0.16	0.13	0.35

Table 5.7: We compute the sentence similarity based on the methods described in Section 5.4.2. Mean reciprocal rank (MRR) and accuracy (Acc) is computed. The results are reported based on the following: the information whether an opinion is implicit/explicit for the implicit/explicit dataset and the category to which an argument belongs to for the Citizen Dialogue corpus is given (With Information) or not given (Without Information).

The main motivation behind this task is to identify the usefulness of the implicit/explicit opinion classification. The two datasets are completely different. To make a fair comparison with the Citizen Dialogue corpus and assess the adaptability of the proposed method, I assume that there is a classification system that is able to classify a premise as a rephraser or a rephrasee. For instance, the length of the premise could be considered as one such feature.

The experiment is carried out on different settings:

1. In the first part of the experiment, the classification system is considered for splitting the dataset into two two categories, one containing rephrasers and the other containing the rephrasees. For the implicit/explicit opinions dataset, implicit opinions and explicit opinions are categorized as rephrasees and rephrasers respectively. A bipartite graph is built with two sets of nodes, one with the rephrasers and the other with the rephrasees. For every rephrasee, the corresponding top 10 rephrasers with the highest cost function is considered as predicted rephrased premises.

2. In the second part of the experiment, the classification system is not considered. Here, we assume that we are given a list of rephrasees for which we need to predict the correct rephrased premises from a given set of premises. A bipartite graph is built with two sets of nodes, one with the rephrasees and the other containing a set of premises (containing rephrasers as well as rephrasees). Every rephrasee node is mapped to every other node, except itself. The corresponding top 10 rephrasers for a given rephrasee node is chosen. By doing this, it is easier to analyse the usefulness of the given information.

The results are reported in Table. 5.7. Two evaluation measures, **MRR** and **Acc** are used, since the Citizen's Dialogue corpus contains only one correct rephraser for every rephrasee node. This is not the case of the implicit/explicit opinions dataset where there are multiple correct rephrasers for every rephrasee node. In the first setting, where the information is given: (1) there are 57 rephrasee nodes and 56 rephraser nodes in the implicit/explicit opinions dataset and (2) there are 64 rephrasees and 64 rephrasers nodes in the Citizen's Dialogue corpus. In the second setting, where the information is not given: (1) there are 57 rephrasee nodes and (56*56) rephraser nodes and (2) there are 64 rephrasee nodes and (63*64) rephraser nodes. By observing the results, there is an improvement when the information is given. This improvement is significantly more for the implicit/explicit opinions dataset and that shows the importance of the stance classification. Again, the sentence embedding representation using **WordPCA+AVG** yields the best performance.

5.4.6 Result Analysis

In this subsection, I investigate the performance of similarity measure, sentiment and topic in predicting the correct rephrased premises for the rephrasees by looking into the results. Firstly, I consider the results when the cost function uses all three functions – $\mathrm{sim}, \mathrm{Q}, \mathrm{R}$ (Eq. 5.12) for computing the cost. I compare these with the results when the cost function uses only the sentiment and topic function (Q, R in Eq. 5.12). The similarity measure is computed using sentence embeddings obtained using *WordPCA+AVG.*

By observing the results, in some cases, sentiment and topic are not able to predict the answers correctly while in other cases, the similarity measure fails to the capture the information that is explicitly provided by sentiment and target.

To illustrate this, I use a few examples. Let us consider an implicit opinion *"but the service is totally different with so many rooms for improvement it became not acceptable"* for which the first ranked predicted explicit opinion when using all three functions ($\mathrm{Sim} + \mathrm{Q} + \mathrm{R}$) for computing the cost was *"we were extremely unimpressed by the quality of service we encountered"*. The answer is predicted correctly since both the implicit and explicit opinion express the same argument about the aspect "service". However, the first ranked predicted explicit opinion using the sentiment and topic functions ($\mathrm{Q} + \mathrm{R}$) for computing the cost is *"the rooms are not worth the money"*, which is not a correct answer. It can be seen that the word "rooms" in the implicit opinion has been wrongly interpreted to hotel rooms and this mismatch cannot be captured by sentiment and topic information alone. This is because the sentiment and topic functions, unlike the similarity measure, do not capture any contextual information resulting in predicting answers randomly based on the sentiment and topic information.

Another example is an implicit opinion *"this hotel could easily be 5 star, the facilities are fantastic, the rooms beautifully furnished and equipped with all the latest technology"* with the top-ranked explicit opinion using $\mathrm{Sim} + \mathrm{Q} + \mathrm{R}$ as *"the hotel rooms are nice"*. This answer, while bland, is a correct match for the implicit opinion. For the same example, the top-ranked opinion using $\mathrm{Q} + \mathrm{R}$ is *"the rooms are not worth the money"* which is completely wrong, even though the aspect has been correctly determined.

In both these examples, the similarity measure seems to work well. However, there are cases where the contextual information captured by the similarity measure is not sufficient, especially where the domain knowledge information that identifies different aspects as the same topic. For example the implicit opinion *"the laundry came back promptly"* is correctly matched with the explicit opinion *"the service was great"* by the sentiment and topic functions, but the similarity measure does not recognise these opinions as being similar. This might be because both sentences are quite short, and many of the words they contain — "came", "was", "back" and so on — are common words that are not good features for opinion matching. It is also possible that the embeddings of the words "laundry" and "service" were not available or were not present as close word pairs.

5.5 Conclusion

The thesis explores the different steps of the argument mining pipeline that I propose for processing natural language arguments present in opinionated texts. In the previous chapter, the second step of the argument mining pipeline is investigated where opinions are classified as implicit and explicit based on the stance expressed. This kind of classification helps in interpreting implicit and explicit opinions as enthymemes and arguments respectively. In this chapter, the next step of the argument mining pipeline is explored by making use of the implicit and explicit opinions and relating them using two types of relation: (1) support-based entailment and (2) rephrase relation that exist among implicit and explicit opinions. These relations can help in relating arguments with similar enthymemes and can be useful for reconstructing enthymemes. To do this, Freeman-style [29] serial and linked argument structures are constructed using support-based entailment and rephrase relations respectively. In doing this, the following research question is explored.

Research Question 2: How do "stance", "sentiment" and "topic" help in relating opinions as premises supporting a conclusion and what kind of argument structures are obtained?

The research question is answered by predicting the support-based entailment and the rephrase relations among a set of opinions extracted from hotel reviews. The three linguistic properties "stance", "sentiment" and "topic" are used for proposing different rules for predicting the support-based entailment relation and experimental results show that these properties are able to capture the support as well as entailment relation which is not captured by state-of-the-art existing textual entailment methods. A serial argument structure is constructed using the support-based entailment relation where a set of premises are linked in a serial fashion and support a conclusion. Again, "stance", "sentiment" and "topic" are used for predicting the rephrase relation and there exists a semantic similarity between two opinions related by the rephrase relation. Experiments and results show that the implicit/explicit opinion classification has been useful for predicting the rephrase relation. A linked argument structure is constructed using the rephrase relation in which a set of premises, not related to each other, together support a conclusion.

Different set of rules are proposed for a distant-supervision based approach for creating datasets with T-H pairs satisfying the support-based entailment relation. Experiments are conducted to analyse the performance of the existing state-of-the-art textual entailment algorithm for predicting the entailment in these datasets. Experiments show that current textual entailment fails to capture the support relation where a specific premise supports a generalised premise. This is overcome with the help of three different linguistic attributes: (1) sentiment, (2) classifying opinions as implicit/explicit, which gives us three different domain-based ontology relations namely subsumption, inclusion and equivalence and (3) the domain based knowledge base. The accuracy reported for the Fully-Annotated, Semi-Annotated and Unannotated datasets for datasets created using the support-based entailment rules are 89.54%, 90.00% and 96.19% respectively. The support-based entailment relation gives us a serial argument structure that supports a conclusion, in which, a set of premises are linked in a chain fashion.

But, among certain implicit and explicit opinions, there exists a rephrase relation such that both the implicit and explicit opinion express the same argument and the explicit opinion rephrases the implicit opinion. Annotating a large dataset is time-consuming and hence a small dataset of 57 implicit opinions was manually compared against 56 explicit opinions. An unsupervised bipartite graph-based approach is proposed using three different features: similarity measure, sentiment and topic for identifying implicit-explicit opinions that satisfy the rephrase relation. Different sentence embedding representations were investigated and the best performance was achieved by performing two post-processing steps on pre-trained word embeddings and averaging the word embeddings of an opinion to represent its embedding. Results are analysed based on the different features used and it shows that combining all three features can give us the best performance. This type of relation leads to a linked argument structure supporting a conclusion, in which a set of premises are not related but support as a group. The adaptability of the proposed method for a dialogue based dataset also shows that it can be useful for predicting the rephrase relation in other domains.

The two relations, support-based entailment and rephrase relations, can give us a combination of a serial and linked based argument structure summarizing an argumentative insight into a set of reviews that have a common conclusion, which is either in favour or against the product or service. The results observed in this chapter explains how opinions as premises of arguments or enthymemes are related in two different structures depending on the overall conclusion that they support. These structures, while strengthening the conclusion that they support, are actually difficult to evaluate without any human intervention. These structures may strengthen the persuasiveness of a conclusion by investigating the accrual properties of these structures. However, again, this is not investigated in this thesis. This work presents a detailed analysis of the use of natural language processing methods and their disadvantages for identifying different argument structures.

In the next chapter, I investigate on the next step of the argument mining pipeline by constructing bipolar argumentation graphs. This does not directly make use of the work discussed in this chapter. That is because it works at the level of abstract arguments rather than the structured arguments investigated here. However, structured arguments can easily be converted to abstract arguments, so pieces of work fit into the same conceptual argument mining framework. One reason to shift towards abstract argumentation techniques is its adaptability for natural language arguments that helps in exploring the different areas of computational argumentation. Both, explicit and implicit opinions, are assumed as abstract arguments that can be related using support and attack relations. In doing so, there are two things that are explored. The strength of the opinions is computed using support and attack relations. Next, the bipolar argumentation graphs are converted into a coalition of arguments, in which, arguments supporting each other are grouped together. I propose different methods for computing the strength of the coalitions, different methods of aggregating these coalitions, different functions based on the aggregation methods for predicting the overall sentiment of reviews.

Chapter 6

Aggregating abstract arguments

6.1 Introduction

In the previous chapters, the different steps of an argument mining pipeline for constructing structured arguments are explained. Opinions were classified as implicit and explicit opinions based on how the stance expressed is studied and these implicit and explicit opinions are shown to have two types of support relations – support-based entailment relation and rephrase relation. The two relations are useful for constructing structured arguments in the form of premises that together support a particular conclusion. There is no direct relation between the previous chapters and this chapter since evaluating the structured arguments is a subjective task and is not explored in this thesis. Instead, the opinions identified to form the structured arguments are considered as abstract arguments. Opinionated texts are often written to present a viewpoint regarding a certain topic or an issue and so can be considered as putting forward a set of abstract arguments (Def. 2.1) about that topic. The overall view presented in the text is clearly dependent on the combination of the arguments. This chapter addresses the problem of automatically weighing up such a set of abstract arguments from a piece of text to establish the overall view expressed in the text. In particular, the work in this chapter considers how to identify relations among abstract opinions and how to model these arguments as an abstract framework using their linguistic properties. I term this process as "*collective opinions*". This is the process of combining a set of opinions about a common topic that together can strengthen a conclusion which is either for or against the topic.

This type of summarizing or clustering opinions is made possible by exploiting their textual properties but such a representation does not provide an understanding of how the set of opinions can strengthen the conclusion. Instead, modelling the set of opinions using methods from formal argumentation can utilize the argumentative relations among these opinions to capturing their strength towards a conclusion. One such method is that of "*coalitions of arguments*" (Def. 2.3) and that is what is used here to represent these collective opinions. It has to be noted that this definition of coalitions is not related to the coalitions represented in game theory. This method is based on a bipolar argumentation representation where the arguments are related by support and attack relations. The next step in the argumentation process depends on evaluating these arguments based on their strength and in this chapter, I answer the question of how to adapt formal argumentation techniques and combine them with natural language processing methods for this task. I also evaluate the argumentation techniques as a machine learning approach for a particular NLP task. Reviews are a good example of the kind of opinionated texts that present a viewpoint about an aspect or an aspect category. In this work, I consider a particular task that is relevant to reviews for evaluating the usefulness of an argumentation based aggregation process that is summarizing the overall view expressed in the review. An approach that is similar to a supervised approach in machine learning is proposed for predicting the overall sentiment of the review. But instead of learning the linguistic properties from the training data as done in a traditional supervised approach, the coalitions are aggregated using a given method and values of their strengths are learned. This value is used for predicting the strength of a review and that in turn is used for predicting the overall sentiment as a binary classification – positive or negative.

A coalition is a set of arguments supporting each other directly or indirectly. The main motivation is to propose different ways of constructing these coalitions by exploiting the linguistic attributes present in opinionated texts and assessing whether these coalitions can effectively become arguments on their own. The coalitions are aggregated based on conditions that take into account the linguistic properties of the arguments that are useful for predicting the overall sentiment of reviews.

An abstract argument present in reviews consists of the following meta-attribute properties: (1) sentiment, (2) stance and (3) aspect or aspect category. Hence, every argument in a review has what I term a "***local sentiment***" and the overall star rating of a review can be considered as indicating, as above, what I call the "***overall sentiment***".

Task Description

In this task, a set of reviews belonging to a set of training data are categorized based on the overall sentiment as either low rated reviews or high rated reviews. Low rated reviews are those with an overall sentiment as negative, that is having a star rating of 1.0 or 2.0, and high rated reviews are those with an overall sentiment as positive, that is having a star rating of 4.0 or 5.0. Coalitions are formed such that they satisfy the following definition:

- *A coalition consists of a set of arguments supporting each other directly or indirectly such that the local sentiment of these arguments are same as the overall sentiment of the reviews that contain them and, the strength of the coalition promotes the value of the overall sentiment.*

The above definition differs from the existing definition of coalitions of arguments (Def. 2.3) and takes the sentiment of the arguments into consideration.

Results are analysed for two main factors: (1) the effect of the strength of coalitions on the performance of the prediction and, (2) the effect of the aggregation of coalitions on the performance of the prediction. Evaluating an NLP-based task such as the overall sentiment prediction helps in understanding the use of formal argumentation for real-world applications, which looks beyond the shallow linguistic features captured by current NLP based techniques.

The following subsections present the different methods that are proposed for the different steps taken to predict the overall sentiment and these are present below:

Step 1 Computing the strength of the coalitions.

Step 2 Aggregating coalitions as individual combined arguments.

Step 3 Sentiment prediction on the basis of the support relation and the coalitions strength values for computing the overall sentiment value of a review.

6.1.1 Step 1: Computing coalition strength

Given a set of reviews, arguments are extracted from low rated and high rated reviews separately. By doing this, we are evaluating the arguments that help in promoting the overall sentiment with arguments that are not strong enough to attack the overall sentiment as a conclusion. Here, I consider arguments that support the overall conclusion as strong arguments and arguments that are against the overall conclusion as arguments that attack the conclusion but not strong enough to change the overall conclusion. Bipolar argumentation graphs are then formed by relating the arguments using support relation $\mathcal{S}$ and attack relation $\mathcal{R}$ (Def. 2.2).

The strengths of the arguments in the bipolar argumentation graphs $\nu(a_i) \in \mathbb{R}$, are computed using the attackers and supporters of the arguments. There are several ways of computing the strength of the arguments and these are widely studied in the argumentation community. In this work, I do not consider all the semantics but instead, use some of them to show how these can be used for computing the strength of natural language arguments.

Definition 6.1 (Attacker and Supporter of an argument). For every $a, a' \in \mathcal{A}$, where both belong to the same BAF, a is an attacker of a' if $a, a' \in \mathcal{R}$ and a is a supporter of a' if $a, a' \in \mathcal{S}$.

Existing approaches in defeasible reasoning and abstract argumentation, for computing the strength of an argument are discussed in Chapter 2. These depend upon the supporters and attackers of a given argument.

More formally, consider that we have a set of arguments $\mathcal{A} = \{a_1, a_2, ..., a_n\}$ in a bipolar argumentation framework with an attack relation $\mathcal{R}$ and a support relation $\mathcal{S}$. Each argument a_i has an associated value $v(a_i)$ and can be thought of as the strength of a_i. Now, a particular argument a will be supported by some of the a_i and will be attacked by others. I want to combine the strengths of the arguments for a and the arguments against a to come up with an overall value for a and to do this, I use Eq. 2.1 and Eq. 2.5 and define the following.

Definition 6.2 (Valuation). For every argument $a \in \mathcal{A}$ with a set of supporters $\mathcal{B} = \{b_1, b_2, ..., b_n\}$ and attackers $\mathcal{C} = \{c_1, c_2, ..., c_m\}$ the valuation of a is defined as:

$$v(a) = f(h^{sup}(v(b_1), ..., v(b_n)), h^{att}(v(c_1), ..., v(c_m))) \tag{6.1}$$

where:

$$h^r : \mathbb{R}^n \to \mathbb{R}$$

is a function that maps a given set of arguments to a single value with the argument relation r in consideration. I then use h^{sup} and h^{att} to denote the cases where $r =$ support, and $r =$ attack respectively.

In this work, I use the following to compute the function h^r.

$$h^r_{agg}(\mathcal{A}) = \sum_{i=1}^{n} v(a_i) \tag{6.2}$$

Besnard and Hunter proposed Eq. 2.1 for propositional logic-based argumentation where arguments are constructed from formulae and for a given proposition α, α^+ and α^- are the accumulated values of arguments that are for and against the proposition respectively. Since we are dealing with an abstract view of arguments, we can't talk about arguments for and against a proposition. Instead, I take an argument a as our starting point and using the strength values of the arguments that support and attack it, I compute the strength of the argument a as follows:

$$f(h^{sup}_{agg}, h^{att}_{agg}) = h^{sup}_{agg} - h^{att}_{agg} \tag{6.3}$$

I call Eq. 6.3 the accumulator method because the strength of the argument is the accumulated value based on its supporters and the accumulated value based on its attackers.

Now, based on the Eq. 2.5, in the same bipolar argumentation framework as above, we have:

Definition 6.3 (Gradual valuation)**.** For every argument $a \in \mathcal{A}$ with a set of supporters $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$, and attackers $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$, the gradual valuation of a is defined as:

$$\nu(a) = g(h^{\text{sup}}(\nu(b_1) \ldots, \nu(b_n)), h^{\text{att}}(\nu(c_1), \ldots, \nu(c_m)) \tag{6.4}$$

where $h^{\text{sup}}(\cdot, \ldots, \cdot)$ and $h^{\text{att}}(\cdot, \ldots, \cdot)$ are as above.

Again, I compute the function h^r as in Eq. 6.2 and I also follow [43] in defining $g(h^{sup}_{agg}, h^{att}_{agg})$ by:

$$g(h^{sup}_{agg}, h^{att}_{agg}) = \frac{1}{(h^{att}_{agg} + 1)} - \frac{1}{(h^{sup}_{agg} + 1)} \tag{6.5}$$

I call Eq. 6.5 the graduality method because the strength of the argument depends on the gradual increase and decrease of the accumulated values of its supporters and the accumulated values of its attackers.

The support relations present within the bipolar argumentation graphs are used to convert them into coalitions of arguments (Def. 2.3) such that, every argument associated with a coalition has a direct or indirect supporter within it and there are no attackers present.

Three different measures are proposed for associating a strength value to the coalitions, on the basis of the strength of their constituent arguments:

agg The sum of the strength values of every argument present in a coalition.

max The maximum strength value among the arguments present in a coalition.

min The minimum strength value among the arguments present in a coalition.

These three measures are studied to analyse how arguments within a coalition are influenced by each other and can represent the strength of the coalition.

6.1.2 Step 2: Aggregating Coalitions

The next step in the process is to aggregate different coalitions constructed from a given set of low rated and high rated reviews. The coalitions are treated as if they are single, combined arguments and associated with a strength derived, as above, from their constituent arguments. Prior work [2] based on reviews has observed that most of the negative opinions are present in low rated reviews and most positive opinions are present in high rated reviews. From this observation, it seems that certain conditions need to be looked at for picking the relevant coalitions to predict the sentiment of a review. Coalitions of arguments with a negative sentiment that are formed from low rated reviews and those with a positive sentiment that are formed from high rated reviews are considered.

Different ways of aggregating the coalitions are investigated based on three different criteria as discussed below.

1. **Criteria 1: Strength of the coalitions** *In this criteria, one of the three strength measures (*agg, max, min*) is used to represent the strength of the coalitions.*

2. **Criteria 2: Topic-topic relation for coalition formation** *Online reviews consist of aspects or entities as meta-attributes within the arguments present in opinions. These aspects can be grouped into different categories based on their common properties. In this criteria, the coalitions are formed based on the relevant aspect or the aspect category.*

3. **Criteria 3: Choosing coalitions** *Coalitions are present in two different categories – each representing a conclusion. There are low rated reviews, which give the*

Method	**Criteria 1: Strength of coalitions**	**Criteria 2: Topic-topic relation for coalition formation**	**Criteria 3: Choosing coalitions**
$\mathcal{C}(agg, AC, U)$	*agg*	Aspect category	All
$\mathcal{C}(min, AC, U)$	*min*	Aspect category	All
$\mathcal{C}(max, AC, U)$	*max*	Aspect category	All
$\mathcal{C}(agg, A, U)$	*agg*	Aspect	All
$\mathcal{C}(min, A, U)$	*min*	Aspect	All
$\mathcal{C}(max, A, U)$	*max*	Aspect	All
$\mathcal{C}(agg, AC, S)$	*agg*	Aspect category	Strongest
$\mathcal{C}(min, AC, S)$	*min*	Aspect category	Strongest
$\mathcal{C}(max, AC, S)$	*max*	Aspect category	Strongest
$\mathcal{C}(agg, A, S)$	*agg*	Aspect	Strongest
$\mathcal{C}(min, A, S)$	*min*	Aspect	Strongest
$\mathcal{C}(max, A, S)$	*max*	Aspect	Strongest

Table 6.1: Each method is represented as a tuple $\mathcal{C}$ with the three criteria as reported above.

> *conclusion that the overall sentiment is negative, and high rated reviews, which give the conclusion that the overall sentiment is positive. There are two ways of choosing the coalitions: (1) choosing all the coalitions and (2) choosing the strongest coalition, one from each category.*

These methods are represented as a tuple $\mathcal{C}(Criteria1, Criteria2, Criteria3)$ and in total, 12 different methods are present as reported in Table 6.1.

6.1.3 Step 3: Sentiment prediction methods

In this step, we consider the coalitions that are selected using one of the above aggregation methods described in the previous subsection. These coalitions are considered as arguments on its own, and for a given review, the coalition-to-argument support relation is defined as below.

Definition 6.4 (Coalition-to-argument support relation). For every argument in coalition $\mathcal{C}$, there exists a support relation between $\mathcal{C}$ and an argument b, if and only if at least one argument $a \in \mathcal{C}$ supports b.

An argument supported by a coalition has the strength of the coalition associated with it. There can exist different coalition-to-argument support relations for a given review and we distinguish the values that arise from the support relations as follows: the values of the supporting coalitions which I term as *supporting coalitions values* (SCV) and the values of the attacking coalitions which I term as *attacking coalitions values* (ACV). Given the arguments in a review, these values summarize the weight of support (SCV) and attack (ACV) for a hotel as represented in that review, and I use this information to compute a sentiment score for that review.

To do this computation, I introduce two sentiment prediction functions, based on Eq. 6.3 and Eq. 6.5 that are used for computing the strength of an argument. In both the equations, ACV and SCV are used rather than the count of attacking and supporting arguments. However, there is one important way in which the sentiment prediction functions differ from Eq. 6.3 and Eq. 6.5. In Eq. 6.3 and Eq. 6.5, supporting and attacking arguments are weighted equally. In the sentiment prediction functions, SCV and ACV are weighted with a pair of values α and β. This is done in order to capture the fact that people seem to weight the two classes of argument differently, and introducing the coefficients allows us to capture how people do this weighting. This is empirically shown in Figure 6.3 and explained in detail in Section 6.3.3.

$$f'_{agg}(SCV, ACV) = \alpha \sum(SCV) - \beta \sum(ACV) \tag{6.6}$$

$$f'_{max}(SCV, ACV) = \alpha(\max(SCV)) - \beta(\max(ACV)) \tag{6.7}$$

$$g'_{agg}(SCV, ACV) = \frac{1}{(\beta \sum ACV + 1)} - \frac{1}{(\alpha \sum SCV + 1)} \tag{6.8}$$

$$g'_{max}(SCV, ACV) = \frac{1}{(\beta(\max(ACV)) + 1)} - \frac{1}{(\alpha(\max(SCV)) + 1)} \tag{6.9}$$

α and β are factors used to allow the support and attack components to be weighted differently and $\alpha + \beta = 1$.

I term Eq. 6.6 the accumulator function for sentiment prediction as the function output depends on the accumulated values of supporting coalitions and the accumulated values of attacking coalitions. I term Eq. 6.8 the graduality function for sentiment prediction as the function output depends on the gradual increase and decrease of the accumulated values of supporting coalitions and attacking coalitions respectively.

6.2 Proposed methodology

A supervised-based approach using a given set of reviews as training data is carried out. There are two different ways of computing the strength of the arguments present in the training data, after which coalitions are formed and aggregated according to the methods described in the above sections. For reviews present in the test data, the overall sentiment is predicted using the scores that can be obtained using either of the two sentiment prediction functions.

Hence, there are four different combinations to be carried out.

1. **Accumulator-Accumulator** In this method, the strength of the arguments in the training data is computed using Eq. 6.3 which is based on the accumulator function (Eq. 2.1). The sentiment prediction function (Eq. 6.6) based on the accumulator function is used to predict the overall score of the test review.

2. **Graduality-Graduality** In this method, the strength of the arguments in the training data is computed using Eq. 6.5 which is based on the gradual valuation function (Eq. 2.5). The sentiment prediction function (Eq. 6.8) based on the gradual valuation function is used to predict the overall score of the test review.

3. **Accumulator-Graduality** In this method, the strength of the arguments in the training data is computed using Eq. 6.3 which is based on the accumulator function (Eq. 2.1). The sentiment prediction function (Eq. 6.8) based on the gradual valuation function is used to predict the overall score of the test review.

4. **Graduality-Accumulator** In this method, the strength of the arguments in the training data is computed using Eq. 6.5 which is based on the gradual valuation function. The sentiment prediction function (Eq. 6.6) based on the accumulator function (Eq. 2.1) is used to predict the overall score of the test review.

6.3 Experiments and Results

6.3.1 Data

As previously discussed (Chapter 2), the ArguAna [2] corpus contains manually annotated hotel reviews from TripAdvisor.com and this data is used for the experiments. [2] used a crowdsourcing approach for manually annotating the reviews with the following features:

- local sentiment of the statements
- aspects present in the statements are highlighted

Hotel	**Location**	**Service**	**Room**	**Value**	**FrontDesk**
hotel	location	service	bathroom	value	front desk
5/4/3/2/1 star	shop(s)	breakfast	bed	price	staff
inn	underground	restaurant	decor	cheap	receptionist
motel	transport	laundry	suite	overprice	check-in
	route	bar	internet	money	manager

Table 6.2: Examples of *aspects* (normal face) present within each *aspect category* (bold face) for the hotel dataset.

The aspects that are manually identified in the ArguAna corpus are extracted and grouped into different categories. This gives a list of aspects and five different aspect categories namely *location*, *service*, *room*, *value* and *frontdesk* respectively. A few examples are present in Table. 6.2.

6.3.2 Arguments and relation extraction

The arguments are extracted such that an argument is a sentence-level statement that talks about an aspect or aspect category, with a sentiment that is positive or negative. As these arguments are pieces of natural language text, semantic similarity and sentiment are used to identify the support and attack relations. A support relation means that two arguments have the same sentiment, talk about the same aspect or aspect category and are semantically similar. An attack relation means two arguments are opposite in sentiment, talk about the same aspect or aspect category but are semantically similar. The support and attack relations, unlike the abstract support or attack relation, are symmetric in nature. This is because, in the natural language processing community, the semantic similarity measure between two statements is considered to be symmetric in nature.

An existing semantic similarity measure tool, Takelab system [64] is used to measure the semantic similarity score between a pair of arguments. This system takes a pair of arguments as input and produces a score ranging from 0.0 (lowest similarity score) to 5.0 (highest similarity score). It follows a supervised approach where a supervised regression model is trained using a large number of features on the MSR-video training dataset that contains 750 pairs of sentences. These sentence pairs are manually annotated with a similarity score ranging from 0 (lowest similarity score) to 5 (highest similarity score). Some of the features include:

- N-grams overlap between the pairs of sentences for unigrams, bigrams and trigrams.
- Wordnet-based word overlap where words that are not common in both the sentences are present with partial scores using Wordnet.
- Weighted word overlap where important words are weighted based on their frequencies obtained from the Google N-grams corpus.
- Greedy lemma aligning overlap where similarity is measured between the lemmas present in both the sentences.
- Sentences are represented as vectors that are the summation of the distributional vector of each word in the sentence and cosine similarity is used to measure the similarity.
- Syntactic features where the overlap between the dependency relations of the two sentences is considered.

A minimum similarity score of 1.0 is considered as a threshold value above which a relation is present and below which it is discarded. However, the semantic similarity score cannot capture the difference between support and attack relation and using the sentiment information about the arguments can help in predicting the relation as follows:

Support relation A support relation is present between a pair of arguments with the same sentiment and the semantic similarity score between the arguments is above 1.0.

Attack relation An attack relation is present between a pair of arguments with opposite sentiment where the semantic similarity score between the arguments is above 1.0.

Arguments	**Type**	**Sentiment**	**Aspect**	**Aspect category**
A1: the whole trip was ruined by the guest service of the hotel.	attacking	negative	guest service	service
A2: the valet guy was extremely helpful	supporting	positive	valet guy	service
A3: only negative is the bathroom area is a little small.	attacking	negative	bathroom	room
A4: the room is medium size, clean and comfortable.	supporting	positive	room	room

Table 6.3: Examples of statements that constitute arguments. Each statement is described by its type, sentiment, aspect and aspect catgeory. "Type" is whether the argument is supporting or attacking, itself established by the sentiment of the argument.

These pairs of arguments are paired either based on the aspect or aspect category. A few examples are present in Table. 6.3.

6.3.3 Coalitions vs Arguments

Reviews from a randomly selected hotel are used to compare the results of two different methods: (1) using individual arguments i.e arguments with positive sentiment present in low rated reviews and arguments with negative sentiment present in high rated reviews and, (2) using a particular coalition method $\mathcal{C}(max, A, S)$. For both the methods, Eq. 3.2 is used for predicting the overall sentiment score and the values of α and β are varied. Figure. 6.1 gives the comparison between the two different methods using the overall sentiment scores across the full range of α and β values. The figure shows that, when using the coalitions method, there is a clear gap between low rated and high rated reviews that is captured by weighing the support and attack components using α and β values. In the experiments that are carried out in the following subsections, $\alpha = 0.75$ and $\beta = 0.25$ are considered, since these values correspond to the gap between high and low reviews.

6.3.4 Coalitions methods

Reviews from 23 different hotels are used in this experiment. For each hotel, a given set of reviews are present. Suppose, N_i represents the number of reviews present for

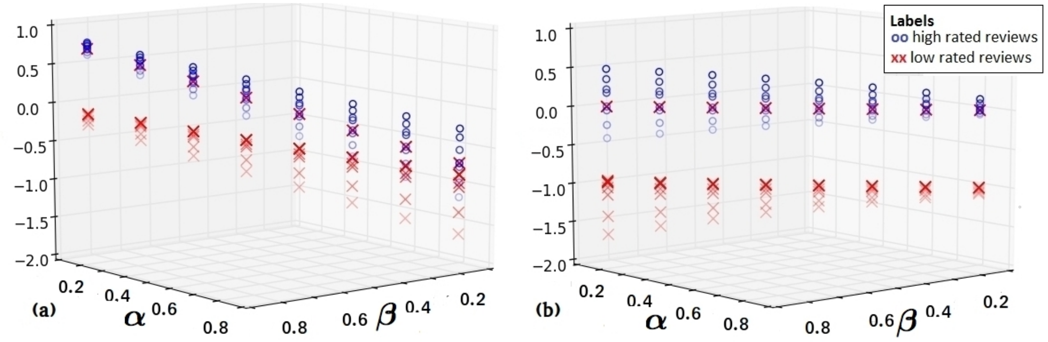


FIGURE 6.1: Scores for each review in a hotel is plotted against varying α and β values. Red cross denotes a review that is low rated and blue circle denotes a review that is high rated. In the Figure, (a) Scores vs α vs β using individual arguments and (b) Scores vs α vs β using coalition method $\mathcal{C}(max, A, S)$. Eq. 3.2 is used for predicting the overall sentiment score.

i-th hotel. For each i-th hotel, each j-th review such that $j = 1, ..., N$ is considered as the testing data and the remaining $N - 1$ reviews are considered as the training data. Reviews in the training data are used for constructing bipolar argument graphs, the strength of the arguments are computed, coalitions are formed and the coalitions are aggregated. For the given testing data, the support from the different coalitions to the arguments present in the testing data is collected as ACV and SCV values. These values are fed into the sentiment prediction functions and the overall sentiment score is predicted. If the score is above 0, the overall sentiment is considered positive; if the score is below 0, the overall sentiment is taken to be negative.

The micro-averaged per-class accuracy for predicting the two classes, low-rated and high-rated, for reviews present in 23 different hotels is defined as follows:

$$Per - Class - Accuracy_{class_i} = \frac{\sum_{j=1}^{23} TP_j}{\sum_{j=1}^{23} TR_j} \tag{6.10}$$

In the above equation, $class_i \in \{Low, High\}$ and TP represents the true positive predictions for each hotel data and TR represents the total number of reviews that belong to $class_i$.

The per-class accuracy across the 23 different hotels using the different coalitions methods is reported in Table. 6.4. Further, the argument strength functions are modified to consider supporters only and the results are present in Table. 6.5. Again, the argument strength functions are modified to consider attackers only and the results are present in Table. 6.6. The results of the *Acc-Acc* and *Acc-Grad* comparisons remain the same regardless of whether the overall sentiment prediction function uses graduality or not. The same can be said for the results of *Grad-Grad* and *Grad-Acc*. The observations from the results present in Tables. 6.4,6.6,6.5 show that the overall sentiment depends on the different ways in which the strength of the arguments is computed and not the sentiment prediction methods. Hence for further experimentation, I will be using *Acc-Acc* and *Grad-Grad*.

		Acc-Acc,Acc-Grad		**Grad-Grad,Grad-Acc**	
Methods	**Category**	*agg*	*max*	*agg*	*max*
$\mathcal{C}(agg, AC, U)$	Low	0.61	0.56	0.57	0.47
	High	0.22	0.22	0.22	0.22
$\mathcal{C}(min, AC, U)$	Low	0.57	0.46	0.56	0.49
	High	0.22	0.24	0.22	0.21
$\mathcal{C}(max, AC, U)$	Low	0.61	0.49	0.58	0.47
	High	0.23	0.24	0.21	0.23
$\mathcal{C}(agg, A, U)$	Low	0.61	0.56	0.59	0.50
	High	0.23	0.26	0.22	0.25
$\mathcal{C}(min, A, U)$	Low	0.56	0.45	0.56	0.45
	High	0.22	0.24	0.22	0.23
$\mathcal{C}(max, A, U)$	Low	0.62	0.49	0.63	0.49
	High	0.23	0.24	0.21	0.24
$\mathcal{C}(agg, AC, S)$	Low	0.71	0.68	0.56	0.47
	High	0.17	0.18	0.16	0.17
$\mathcal{C}(min, AC, S)$	Low	0.62	0.53	0.29	0.28
	High	0.19	0.20	0.12	0.12
$\mathcal{C}(max, AC, S)$	Low	0.64	0.55	0.43	0.40
	High	0.19	0.20	0.14	0.14
$\mathcal{C}(agg, A, S)$	Low	0.77	0.72	0.75	0.61
	High	0.19	0.21	0.18	0.20
$\mathcal{C}(min, A, S)$	Low	0.72	0.57	0.70	0.56
	High	0.20	0.19	0.19	0.20
$\mathcal{C}(max, A, S)$	Low	0.77	0.64	0.79	0.60
	High	0.19	0.21	0.18	0.20

Table 6.4: Four different combinations as discussed in Sec. 3.2 are used for predicting the overall sentiment of reviews present in 23 different hotel data. The strength of the arguments present in the train data is computed using both supporters and attackers. Micro-averaged per-class accuracy of the predicted results for low rated reviews (Low) and high rated reviews (High) is reported.

		Acc-Acc,Acc-Grad		**Grad-Grad,Grad-Acc**	
Methods	**Category**	*agg*	*max*	*agg*	*max*
$\mathcal{C}(agg, AC, U)$	Low	0.61	0.56	0.61	0.44
	High	0.22	0.25	0.23	0.23
$\mathcal{C}(min, AC, U)$	Low	0.58	0.46	0.59	0.49
	High	0.22	0.24	0.22	0.23
$\mathcal{C}(max, AC, U)$	Low	0.61	0.49	0.62	0.50
	High	0.22	0.24	0.23	0.23
$\mathcal{C}(agg, A, U)$	Low	0.64	0.56	0.64	0.43
	High	0.23	0.26	0.21	0.23
$\mathcal{C}(min, A, U)$	Low	0.60	0.49	0.62	0.47
	High	0.24	0.25	0.23	0.27
$\mathcal{C}(max, A, U)$	Low	0.62	0.49	0.63	0.47
	High	0.23	0.24	0.21	0.24
$\mathcal{C}(agg, AC, S)$	Low	0.72	0.68	0.66	0.51
	High	0.17	0.19	0.19	0.20
$\mathcal{C}(min, AC, S)$	Low	0.51	0.47	0.31	0.34
	High	0.19	0.20	0.07	0.07
$\mathcal{C}(max, AC, S)$	Low	0.63	0.55	0.39	0.38
	High	0.19	0.20	0.10	0.10
$\mathcal{C}(agg, A, S)$	Low	0.77	0.72	0.80	0.54
	High	0.19	0.21	0.18	0.21
$\mathcal{C}(min, A, S)$	Low	0.74	0.57	0.74	0.59
	High	0.21	0.21	0.21	0.25
$\mathcal{C}(max, A, S)$	Low	0.77	0.64	0.78	0.58
	High	0.19	0.21	0.18	0.20

TABLE 6.5: Four different combinations as discussed in Sec. 3.2 are used for predicting the overall sentiment of reviews present in 23 different hotel data. The strength of the arguments present in the train data is computed using supporters only. Different coalitions aggregation methods (Methods) for choosing the coalitions from the train data are considered. Micro-averaged per-class accuracy of the predicted results for low rated reviews (Low) and high rated reviews (High) is reported.

		Acc-Acc,Acc-Grad		**Grad-Grad,Grad-Acc**	
Methods	**Category**	*agg*	*max*	*agg*	*max*
$\mathcal{C}(agg, AC, U)$	Low	0.65	0.62	0.57	0.44
	High	0.24	0.25	0.22	0.23
$\mathcal{C}(min, AC, U)$	Low	0.51	0.49	0.56	0.47
	High	0.25	0.24	0.22	0.21
$\mathcal{C}(max, AC, U)$	Low	0.65	0.57	0.58	0.43
	High	0.26	0.25	0.21	0.22
$\mathcal{C}(agg, A, U)$	Low	0.71	0.65	0.60	0.48
	High	0.25	0.27	0.22	0.25
$\mathcal{C}(min, A, U)$	Low	0.45	0.45	0.56	0.45
	High	0.30	0.30	0.21	0.26
$\mathcal{C}(max, A, U)$	Low	0.70	0.59	0.63	0.43
	High	0.25	0.26	0.21	0.25
$\mathcal{C}(agg, AC, S)$	Low	0.72	0.69	0.61	0.49
	High	0.18	0.19	0.15	0.17
$\mathcal{C}(min, AC, S)$	Low	0.53	0.49	0.33	0.36
	High	0.17	0.19	0.09	0.09
$\mathcal{C}(max, AC, S)$	Low	0.72	0.64	0.46	0.39
	High	0.19	0.20	0.15	0.17
$\mathcal{C}(agg, A, S)$	Low	0.80	0.74	0.76	0.59
	High	0.20	0.21	0.18	0.20
$\mathcal{C}(min, A, S)$	Low	0.59	0.58	0.71	0.56
	High	0.27	0.20	0.22	0.18
$\mathcal{C}(max, A, S)$	Low	0.79	0.69	0.79	0.54
	High	0.21	0.21	0.18	0.21

Table 6.6: Four different combinations as discussed in Sec. 3.2 are used for predicting the overall sentiment of reviews present in 23 different hotel data. The strength of the arguments present in the train data is computed using attackers only. Different coalitions aggregation methods (Methods) for choosing the coalitions from the train data are considered. Micro-averaged per-class accuracy of the predicted results for low rated reviews (Low) and high rated reviews (High) is reported.

First, the variations in the results across the three tables for *Acc-Acc* and *Grad-Grad* based methods is analysed and the observations are present below.

1. In Table. 6.4, *Acc-Acc* performs significantly better than *Grad-Grad* using $\mathcal{C}(*, AC, S)$

and the rest of the results remain the same for both. Amongst the different coalition methods, the results are significantly better using $\mathcal{C}(*, A, S)$. Considering the **strongest** coalitions is better than considering all the coalitions. The results also imply that aspects are a better way of constructing coalitions and an aspect being stronger(weaker) need not necessarily mean that the aspect category is stronger(weaker).

2. By comparing the results in Table. 6.4 and 6.6 for *Acc-Acc* method, we can find that except for the coalition methods that are based on *min*, the results in Table. 6.6 are either significantly better or the same compared to those in Table. 6.4. But, this is not the same for *Grad-Grad* based method where the results remain the same. This shows that the attack relation has an important role in the prediction process if the strength of the arguments are changed without gradually increasing or decreasing the impact of the attack relations on the arguments.

3. By comparing the results in Table. 6.4 and 6.5 for the *Acc-Acc* method, we can find the results remain the same. However, for the *Grad-Grad* method, the results are not consistent. In some methods, the support relation does make an impact and this means the strength of the arguments when changed gradually based on the support relations on the arguments can help in the prediction.

6.3.5 Fuzzy Logic-based aggregation

In the previous subsection, I report the results and observations for the different coalitions methods. However, the results in the previous subsection do not answer whether the different coalitions methods that have different ways of computing the strength of the coalitions are predicting the same set of reviews or are predicting a different set of reviews correctly. To analyse the different results based on the different coalition strength methods, I recall the three criteria explained in Section 5.1.2 that are used for aggregating coalitions and these are given below.

- Criteria 1: Strength of the coalitions
- Criteria 2: Topic-topic relation for coalition formation
- Criteria 3: Choosing coalitions

The first criteria represent the strength value of a coalition in three different ways namely *agg*, *min* and *max*. Here, I combine the three different values and this gives four different aggregation methods namely $\mathcal{C}(agg, min, max, AC, U)$, $\mathcal{C}(agg, min, max, A, U)$, $\mathcal{C}(agg, min, max, AC, S)$ and $\mathcal{C}(agg, min, max, A, S)$. In each of these methods, the number of correctly predicted reviews when using *agg*, *max* and *min* are converted into fuzzy based values in the range [0,1]. The final decision is the number of correctly predicted reviews that is chosen using the full reinforcement operator and the upward reinforcement operator over these fuzzy based values. The full reinforcement operator identifies whether a majority of the methods predict the overall sentiment correctly and the upward reinforcement operator identifies whether the correct sentiment is predicted by the method with the highest prediction score. By comparing the results obtained by the full reinforcement operator and the upward reinforcement operator, we can observe whether the correct sentiment is predicted by all the methods or by different methods. Hence, the full reinforcement and upward reinforcement operators are used and these are described below.

Prior work [131] investigates on reinforcement operators based on the fuzzy system modelling technique. One of the defuzzification techniques used in [131] is the Mean of Maximal (MOM) method and by adapting this, I define the two main aggregation operators as below:

Full Reinforcement operator For a given set of fuzzy values, if the values are low, then a t-norm aggregation is performed and, if the values are high, a t-concorm aggregation is performed.

$$\mathcal{FM} = \left\{ \begin{array}{ll} min(\mathcal{C}(max,*,*), \mathcal{C}(min,*,*), \mathcal{C}(agg,*,*)) & \Delta{>}\Omega \\ max(\mathcal{C}(max,*,*), \mathcal{C}(min,*,*), \mathcal{C}(agg,*,*)) & \Delta{<}\Omega \\ \frac{(min(\mathcal{C}(max,*,*),\mathcal{C}(min,*,*),\mathcal{C}(agg,*,*))+(max(\mathcal{C}(max,*,*),\mathcal{C}(min,*,*),\mathcal{C}(agg,*,*)))}{2} & \Delta = \Omega \end{array} \right\}$$

where

$\Delta = \min((1.0 - \mathcal{C}(max,*,*)), (1.0 - \mathcal{C}(min,*,*)), (1.0 - \mathcal{C}(agg,*,*)))$ and

$\Omega = \min(\mathcal{C}(max,*,*), \mathcal{C}(min,*,*), \mathcal{C}(agg,*,*))$

Upward Reinforcement operator $\mathcal{UM} = max(\mathcal{C}(max,*,*), \mathcal{C}(min,*,*), \mathcal{C}(agg,*,*))$

The sets of results that are present in Table. 6.7 follows the following procedure: (1) the strength of the arguments present in the train data are computed in three different ways: (a) supporters and attackers, (b) supporters only and (c) attackers only. These are then used to form coalitions and coalitions are aggregated using the 12 different methods. These 12 different methods are combined based on criteria 1 and we get four different combination methods. The results present in Table. 6.7 do not perform better than the corresponding individual coalitions aggregation methods. Among the individual coalitions aggregation methods, those with *agg* as Criteria 1 performs better than the rest of the methods. But there is no clear evidence to prove that these methods are able to predict all the answers correctly.

The next step is to use the upward reinforcement operator that considers the maximum number of correctly predicted reviews using *agg*, *max* and *min*. The results are present in Table. 6.8 and the results outperform those present in Table. 6.7. However, these results are not better than the results present in Tables. 6.4,6.5 and 6.6 obtained using the 12 different coalitions aggregation methods. Again, this does not present a clear idea of the aggregation methods and further analysis is carried out based on the following observation.

- Coalitions with criteria 3 as strongest (S) perform better than the corresponding results of coalitions with criteria 3 that considers all coalitions (U).

		Acc-Acc		Grad-Grad	
Methods	**Category**	*agg*	*max*	*agg*	*max*
Argument strength:h^{sup}_{agg},h^{att}_{agg}					
$\mathcal{C}((agg, max, min), AC, U)$	Low	0.61	0.55	0.60	0.51
	High	0.18	0.20	0.17	0.20
$\mathcal{C}((agg, max, min), A, U)$	Low	0.63	0.57	0.63	0.52
	High	0.15	0.16	0.15	0.15
$\mathcal{C}((agg, max, min), AC, S)$	Low	0.51	0.47	0.25	0.24
	High	0.18	0.19	0.15	0.16
$\mathcal{C}((agg, max, min), A, S)$	Low	0.79	0.75	0.80	0.63
	High	0.15	0.15	0.15	0.16
Argument strength: h^{sup}_{agg}					
$\mathcal{C}((agg, max, min), AC, U)$	Low	0.61	0.56	0.61	0.47
	High	0.18	0.20	0.20	0.19
$\mathcal{C}((agg, max, min), A, U)$	Low	0.66	0.60	0.70	0.50
	High	0.15	0.17	0.13	0.12
$\mathcal{C}((agg, max, min), AC, S)$	Low	0.49	0.47	0.43	0.28
	High	0.18	0.19	0.17	0.18
$\mathcal{C}((agg, max, min), A, S)$	Low	0.79	0.73	0.84	0.62
	High	0.15	0.17	0.14	0.12
Argument strength: h^{att}_{agg}					
$\mathcal{C}((agg, max, min), AC, U)$	Low	0.55	0.50	0.59	0.47
	High	0.18	0.18	0.18	0.20
$\mathcal{C}((agg, max, min), A, U)$	Low	0.71	0.65	0.64	0.50
	High	0.02	0.04	0.15	0.14
$\mathcal{C}((agg, max, min), AC, S)$	Low	0.52	0.50	0.36	0.30
	High	0.17	0.17	0.15	0.17
$\mathcal{C}((agg, max, min), A, S)$	Low	0.84	0.78	0.80	0.61
	High	0.03	0.04	0.15	0.15

Table 6.7: Full reinforcement operator used for combining the results obtained using coalitions that have the same Criteria 2 and 3. Coalitions combined based on Criteria 1 as *agg*, *max* and *min*. Micro-averaged accuracy of the results is reported.

The experiment is performed by narrowing down the different combinations to considering only those that have the criteria 3 as strongest (S). Results are obtained for using the upward reinforcement operator among three different sets of values: (1) number of correctly predicted reviews using *agg* and *max*, (2) number of correctly predicted reviews using *agg* and *min* and (3) number of correctly predicted reviews using *max* and *min*. The corresponding individual aggregation method that has *agg* as Criteria 1 outperforms the other methods that have *min* and *max* as Criteria 1 (Table. 6.4, 6.6 and 6.5). By comparison, it again outperforms (3) but (3) does perform better than the corresponding individual aggregation methods with *min* and *max* as Criteria 1. The next analysis that is carried out is to compare (1) and (2) where the results of (2) outperform that of (1). This does imply that combining the results of *min* with either *agg* or *max* improves their performance and because *agg* is the best among the rest, the combination of *agg* with *min* gives the best results (Table. 6.9).

		Acc-Acc		**Grad-Grad**	
Methods	**Category**	*agg*	*max*	*agg*	*max*
Argument strength:h_{agg}^{sup},h_{agg}^{att}					
$\mathcal{C}((agg, max, min), AC, U)$	Low	0.62	0.56	0.61	0.51
	High	0.18	0.20	0.18	0.20
$\mathcal{C}((agg, max, min), A, U)$	Low	0.63	0.57	0.63	0.52
	High	0.18	0.20	0.17	0.20
$\mathcal{C}((agg, max, min), AC, S)$	Low	0.75	0.71	0.63	0.55
	High	0.18	0.19	0.16	0.17
$\mathcal{C}((agg, max, min), A, S)$	Low	0.79	0.75	0.80	0.63
	High	0.18	0.21	0.17	0.21
Argument strength: h_{agg}^{sup}					
$\mathcal{C}((agg, max, min), AC, U)$	Low	0.62	0.56	0.65	0.53
	High	0.18	0.20	0.20	0.20
$\mathcal{C}((agg, max, min), A, U)$	Low	0.66	0.60	0.70	0.50
	High	0.19	0.20	0.20	0.20
$\mathcal{C}((agg, max, min), AC, S)$	Low	0.74	0.71	0.71	0.59
	High	0.18	0.19	0.18	0.19
$\mathcal{C}((agg, max, min), A, S)$	Low	0.79	0.73	0.84	0.62
	High	0.19	0.21	0.21	0.22
Argument strength: h_{agg}^{att}					
$\mathcal{C}((agg, max, min), AC, U)$	Low	0.67	0.63	0.61	0.48
	High	0.18	0.18	0.18	0.20
$\mathcal{C}((agg, max, min), A, U)$	Low	0.71	0.65	0.64	0.49
	High	0.19	0.19	0.17	0.20
$\mathcal{C}((agg, max, min), AC, S)$	Low	0.77	0.75	0.69	0.58
	High	0.17	0.18	0.15	0.17
$\mathcal{C}((agg, max, min), A, S)$	Low	0.84	0.78	0.80	0.61
	High	0.21	0.21	0.17	0.21

Table 6.8: Upward reinforcement operator used for combining the results obtained using coalitions that have the same Criteria 2 and 3. Coalitions are combined based on Criteria 1 as *agg*, *max* and *min*. Micro-averaged accuracy of the results is reported. Best results are highlighted in bold.

		Acc-Acc		Grad-Grad	
Methods	**Category**	*agg*	*max*	*agg*	*max*
Argument strength:h_{agg}^{sup},h_{agg}^{att}					
$\mathcal{C}((agg, min), AC, S)$	Low	0.75	0.71	0.62	0.52
	High	0.19	0.19	0.16	0.17
$\mathcal{C}((agg, min), A, S)$	Low	0.78	0.75	0.76	0.63
	High	0.19	0.22	0.19	0.22
Argument strength: h_{agg}^{sup}					
$\mathcal{C}((agg, min), AC, S)$	Low	0.74	0.71	0.71	0.56
	High	0.19	0.19	0.18	0.19
$\mathcal{C}((agg, min), A, S)$	Low	0.79	0.73	**0.84**	0.59
	High	0.18	0.21	**0.22**	0.22
Argument strength: h_{agg}^{att}					
$\mathcal{C}((agg, min), AC, S)$	Low	0.77	0.74	0.68	0.56
	High	0.17	0.18	0.15	0.16
$\mathcal{C}((agg, min), A, S)$	Low	**0.83**	0.78	0.77	0.61
	High	**0.22**	0.21	0.19	0.21

TABLE 6.9: Upward reinforcement operator used for combining the results obtained using coalitions that have the same Criteria 2 and 3. Coalitions are combined based on Criteria 1 as *agg*, *max* and *min*. Micro-averaged accuracy of the results is reported.

6.4 Conclusion

This chapter studies one of the steps of the argument mining pipeline that I propose for processing natural language arguments present in opinionated texts. In the previous chapters, the following steps of the argument mining pipeline are studied: (1) identifying statements as argumentative, (2) classifying opinions as implicit and explicit opinions and (3) studying the relations among explicit and implicit opinions for constructing Freeman-style arguments. In doing these steps, several research questions that are linked arguments, enthymemes and their relations are studied.

In this chapter, I study the final step of the argument mining pipeline by investigating the implicit and explicit opinions as abstract arguments, the support and attack relations among these abstract opinions, aggregating these arguments and coalitions and further computing the strengths of these arguments and coalitions for predicting the sentiment of reviews. There is no direct link between the previous steps of the argument mining pipeline and the work that is carried out in this chapter. However, what is studied in this chapter is an alternative approach to study the implicit and explicit opinions as arguments and the following research questions are answered.

Research Question 3: Can bipolar abstract argumentation help in computing the strength of the identified opinions present in the Freeman-style arguments?

Research Question 4: What kind of an argument structure can we build when the internal structure relating these opinions is ignored and how does "stance", "sentiment" and "topic" affect this?

To answer these questions, implicit and explicit opinions are considered as abstract arguments and bipolar argumentation graphs formed from these arguments is studied. The bipolar argumentation graphs are formed using two relations, the support and attack relations. To understand how the support and attack relations can be captured in opinionated texts, the linguistic properties "sentiment" and "topic" and semantic similarity measures are considered. The strength of an argument is computed using two existing functions that are proposed in the literature for arguments present in a bipolar argumentation graph — and I term these the accumulator function and the graduality function.

The bipolar argumentation graphs obtained using opinionated texts present in reviews contain similar opinions that talk about the same aspect or aspect category. The support relation present among similar opinions are used to convert bipolar argumentation graphs into coalitions of arguments where a coalition is a set of arguments supporting each other directly or indirectly. In this chapter, I proposed different ways of forming coalitions from the bipolar argumentation graphs and different ways of aggregating coalitions to support the conclusion of a review. Further, I proposed three different functions *agg, max, min* for computing the strength of a coalition using the strength of the individual arguments present in them. From here, the following research questions are answered.

Research Question 5a: Can converting bipolar argumentation graphs into coalitions of arguments represent the strength of combined arguments about a topic?

Research Question 5b: If coalitions of arguments represent the strength of combined arguments about a topic, can different coalitions represent the overall sentiment of a set of opinions in a review?

Evaluating the arguments and the strengths computed for them cannot directly depend on the formal argumentation models because the arguments come from natural language texts that may not be logically structured. Hence, a supervised learning approach is used where opinions from a set of reviews are taken to be training data and used for aggregating coalitions of arguments that are used for supporting reviews present in the test set. Sentiment prediction functions are proposed for predicting the overall sentiment of reviews in the test set and these make use of the strength of the coalitions that support arguments present in a test review.

Empirically, the different methodologies adopted in this work were studied and compared to understand the impact of using argumentation models for natural language texts. A comparison of the different results is performed using fuzzy based aggregation techniques and the overall conclusion that answers the above research questions can be drawn from aggregating the results and is as follows:

1. For the sentiment prediction task, aggregating arguments as coalitions and choosing the strongest coalitions, one constructed from a set of low rated reviews and other from a set of high rated reviews gives the best performance if they are weighted differently.

2. A coalition need not always be represented by all its arguments (strength of all the arguments) or the strongest argument (strength of the strongest argument) but may be represented by its weakest argument. The size of a coalition does not always influence its strength.

3. The strength of an argument computed using the accumulator function gives the best performance for the sentiment prediction task in comparison with the strength of an argument computed using the graduality function.

4. Argument strength represented by attackers and computed using accumulator function and argument strength represented by supporters and computed using the graduality function gives the best results (Table. 6.9). This suggests that attackers and supporters have to be handled separately.

The above observations also show that people weight the arguments differently and this has to be taken into consideration for representing combined arguments.

These results observed encourage belief in the use of formal argumentation methods for a deep understanding of the textual content beyond the current linguistic knowledge since the strength of the coalitions that are computed using the strength of the arguments present in the coalition, which again is computed using the support and attack relations, is able to predict the overall sentiment of a review. The combination of natural language methods and formal argumentation techniques strengthens the value of an argument in natural language texts and is represented as the strength values. The empirical evaluation for the overall sentiment prediction task by looking beyond the textual content of an opinion looks promising for integrating formal argumentation techniques for real-world tasks.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The main contribution of this thesis is to study the intersection of computational argumentation and natural language processing techniques for processing natural language arguments present in opinionated texts. An argument mining pipeline is proposed with the following different steps:

INPUT A set of reviews for some product/service.

1. Identifying opinions as argumentative based on their linguistic properties: sentiment, stance and topic.
2. Identifying explicit and implicit opinions based on how stance is expressed in the opinions.
3. Identifying relations among arguments for constructing Freeman-style [29] serial and linked argument structures in favour/against a decision.
4. Computing the strength of the opinions by identifying attack and support relation among opinions as arguments.
5. Aggregating opinions as coalitions of arguments and assessing their strength for the overall sentiment prediction task.

OUTPUT This gives the overall opinion of a set of reviews for the product/service.

The research questions answered are developed on the different steps present in the pipeline described above. Arguments in opinions are considered in the form of structured arguments as well as abstract arguments. For both the cases, three linguistic attributes present in an opinion: sentiment, stance and topic are considered for identifying arguments. The observations of each of the steps carried out strengthen the reason behind choosing these attributes.

First, I consider arguments in opinions in a structured form. As a first step in the process, a set of opinions extracted from hotel reviews are manually annotated as explicit opinions or implicit opinions based on how the stance is expressed in these opinions. This is done to answer the following questions:

Research Question 1a: How is implicit information identified in natural language arguments present within opinionated texts?

Research Question 1b: How does "stance" in opinions help as a means of filling the gap between a premise and a conclusion?

The first question is answered by considering the implicit/explicit opinion classification as a binary classification problem and a supervised based approach is used for automatically classifying the opinions. The manually annotated dataset is used as the training data. An SVM-based classifier is trained using three different feature sets: (1) surface-based, (2) embedding based and (3) hybrid method, of which, the hybrid method gives the best performance. The hybrid method combines average-based embedding with different surface-based features such as unigrams, bigrams and adjective-noun patterns. However, a drawback is the lack of a large annotated dataset that might help in modelling deep learning models. This is overcome by proposing different semi-supervised and weak supervised approaches for automatically labelling a large dataset of opinions. These automatically labelled opinions as training set are fed into an LSTM model and tested on the manually annotated data. Results show that these automatically labelled opinions, although noisy, are useful for modelling deep learning models. This classification helps to fit the opinions in the argumentation process as it helps to identify relations among opinions and thereby constructing different types of argument structures. It also provides a theoretical justification for the second research question by considering explicit opinions as arguments and implicit opinions as enthymemes where implicit opinions contain the stance left unexpressed which is otherwise present in explicit opinions.

The next step in the process is to identify relations that relate these implicit and explicit opinions and the following research question is answered:

Research Question 2: How does "stance", "sentiment" and "topic" help in relating opinions as premises supporting a conclusion and what kind of argument structures are obtained?

The above question is answered by exploring the support-based entailment relation identifies explicit and implicit opinions as either specific premises or generalised premises and the relation exists such that the specific premise supports as well as entails a generalised premise. This relation helps to construct linked argument structures where a premise infers another, leading to supporting a conclusion. A distant-supervision approach to automatically identify this relation is carried out by proposing different sets of rules that make use of sentiment, stance and domain-based ontology relations (subsumption, inclusion and equivalence). Three different datasets with text-hypothesis pairs satisfying support-based entailment relation is created. Experiments are carried out on these datasets for automatically identifying the entailment relation using an existing textual entailment algorithm and results show that the current textual entailment algorithm fails to capture the support-based entailment relation.

Another type of relation that exists is between an implicit and an explicit opinion where the explicit opinion rephrases the implicit opinion. Rephrase relation identifies the two opinions to express the same argument such that an explicit opinion can replace an implicit opinion without changing its meaning. This type of relation results in an argument structure of the convergent type, where different sets of premises together, support a conclusion. An unsupervised bipartite-graph based approach for identifying implicit-explicit opinions satisfying rephrase relation is proposed that considers three different features: sentiment, topic and similarity measures for computing the cost function. Different sentence embedding representations were investigated for computing the similarity measures. Results show that combining all three features for computing the cost function gives the best performance. The analysis of the relations among opinions to construct argument structures can also help in identifying enthymemes and arguments. The results of this work is a starting step for the reconstruction of enthymemes.

In the above step, one possible step to follow would be to find different methods to reconstruct the enthymemes and evaluate the reconstruction process. Instead, I explore a way in which the argument structures are evaluated and answers the following research questions:

Research Question 5a: Can converting bipolar argumentation graphs into coalitions of arguments represent the strength of combined arguments about a topic?

Research Question 5b: If coalitions of arguments represent the strength of combined arguments about a topic, can different coalitions represent the overall sentiment of a set of opinions in a review?

The different steps of the argumentation process is investigated for opinions as simpler structures of abstract arguments since arguments in abstract form gives an opportunity to use existing formal argumentation models for natural language texts. These arguments are related using two types of relations, support and attack to form bipolar argumentation graphs. Semantic textual similarity and sentiment are used for predicting the support and attack relations among opinions. Existing work on computing the argument strength are compared for computing the strength of the arguments. The support relation in a bipolar argumentation graph is used to convert it into a coalition of arguments, in which, a set of arguments support each other directly or indirectly. Different ways of computing the strength of the coalitions, whether choosing certain coalitions is important and how the strength of the coalitions influence an argument is investigated empirically for an NLP based task, which is to predict the overall sentiment of reviews. A supervised approach is carried out in which coalitions are formed, the strength of the arguments and coalitions are computed, aggregating coalitions using different criteria for opinions present in a training data. The aggregated coalitions are used to support arguments present in a test review and sentiment prediction functions are proposed and investigated for predicting the overall sentiment of a test review. The different sets of results are compared using a fuzzy based aggregation model and the observations show that coalitions formed using aspects are better represented.

This thesis answers several research questions that are useful for the work carried out by the argument mining community. The different steps carried out are used to strengthen the three linguistic attributes used for identifying arguments. It means that existing natural language techniques can enhance the argument identification task and it is shown for opinionated texts. Since most of the natural language texts are implicitly stated, the classification of opinions as implicit/explicit gives a new direction of looking at opinions as enthymemes and arguments, which cannot be detected using sentiment or stance only. Different types of relations using the textual content and argumentation properties help to identify argument structures that automatically help in the enthymeme reconstruction task. The argument structures that are created in this work are useful for the argumentation community to research on the implicit information present in natural language texts. Finally, fitting the main steps of the argumentation process using existing argumentation frameworks in combination with natural language techniques for an NLP-based task has given a strong objective to understand and reason on natural language texts beyond the current scope of the NLP community.

In the next sections, I explain the open issues of the work present in this thesis and the future work that can be carried out.

7.2 Open issues

There are several open issues that remain unanswered in this thesis.

7.2.1 Availability of labelled data

In the initial stage, there is no existing dataset available for automatically identifying opinions as implicit/explicit and a single annotator was asked to annotate the opinions. This was considered as less reliable and two annotators were asked to annotate a small dataset by creating a set of guidelines that were specific to the hotel domain, in order to avoid inconsistency. Due to the nature of the reviews, which contains opinions about several aspects, it is difficult to represent a generalised guideline that can be used across other domains. However, since aspect-based sentiment analysis in opinions is still an ongoing research topic in the NLP community, the guidelines can be adaptable for other domains within online reviews. To do this, we need to identify aspects, whether these aspects can be categorized and how opinions about an aspect influence the sentiment of the opinions. This implicit/explicit classification was further used to reconstruct enthymemes by considering implicit opinions as minor premises and explicit opinions as major premises with the overall star rating as the conclusion. This process requires an evaluation and using human annotators to evaluate it becomes a challenge since it is a highly subjective task. Instead, generalising these opinions as premises and working towards creating argument structures as described in argument diagramming was promising. Another issue was the lack of reviews from other domains for comparison. There are product reviews available, for which there are several tools available for identifying the sentiment and aspects. But, other problems arise such as manually annotating the opinions as implicit/explicit and distinguishing the aspects into different categories. The purpose of the work in this thesis is to investigate on identifying arguments and enthymemes among opinions and hence I do not focus on creating datasets for other domains. The work that has been carried out can help in identifying arguments, enthymemes and the different relations in other domains as well but has not been evaluated yet. The datasets that are created for the support-based entailment relation does not consider the implicitly implied aspects that are not stated explicitly in the opinions. This does not affect the proposed rules which will work for these implicitly implied aspects, but this has not been evaluated in this work.

The manually annotated dataset that was used for comparing the results of the rephrase relation depends heavily on manually relating aspects as well as related words or cues about these aspects with the aspect categories. By doing this, we are able to identify implicitly implied aspects but this has not been identified on a larger dataset. This means that the topic function that makes use of this information cannot be used beyond the dataset that is used for evaluation. The performance of the proposed methodology is good without this information but, identifying the information related to aspects of a large dataset can improve the performance.

7.2.2 Textual Entailment vs Argumentative relations

A general issue that was faced is the current NLP textual entailment algorithm and the sentence embedding representations that did not work for this argument mining problem. The textual entailment algorithm fails to capture the support relation and sentence embedding representations that work well for similarity tasks did not perform well for the rephrase relation. These are issues that need are to be addressed by the NLP community, in particular, by introducing better word embeddings model that can work well for identifying argument based components. But, the heterogeneous nature of data makes it difficult to propose a generic framework for identifying argument components across different domains which makes it a challenging task.

7.2.3 Abstract argumentation

The supervised approach for aggregating opinions as coalitions of arguments and further aggregating these as arguments on its own is empirically shown to work well for predicting the overall sentiment of reviews. The experiments were carried out on a small set of reviews which were not separated into train and test data separately but instead, a cross-validation based approach is used. Again, this particular work depends on identifying aspects and aspect categories in the opinions. The dataset that was used for the experiments contains the aspects and aspect categories manually annotated but does not contain enough reviews to separate them into training and test sets.

7.3 Future work

Earlier, I discussed some of the open issues that are not explored further due to time limitation and availability of resources but by solving these, it could help in improving the performance of the proposed methodologies. In this section, I discuss the scope of developing the work presented in this thesis as future work in argument mining tasks and stance classification. In natural language processing research, the definition of stance has been limited to identifying whether the author is for or against the given topic. The classification of opinions as implicit and explicit opinions based on the expressed stance is a novel approach that has not been tackled so far. This kind of a classification can be extended to other domains such as tweets, debates etc and can also help existing work in stance classification. The implicit/explicit classification is similar to sentence simplification problem and can be useful for such tasks. The three datasets created for identifying support-based entailment relation can be investigated for modelling deep learning models and analysis on cross-domain datasets can also be carried out. Current proposed rules for identifying the support-based entailment relation are heavily dependent on the domain-based knowledge base and modelling deep learning models that can learn features without the domain-based knowledge explicitly given can be useful. A comparison of the explicit and implicit opinions about a given topic can be investigated for an existing work [106] that identifies which argument is convincing among a pair of arguments. Such a comparison may require human annotators to annotate among implicit-explicit and explicit-explicit opinion pairs. One of the earliest work in mining arguments from reviews [88] considers a set of argumentation schemes for extracting arguments. The argument structures that are extracted in this thesis can also help these schemes and can be investigated as future work. Another line of future work is to investigate on aggregating opinions as coalitions for other domains such as debates, tweets etc. For instance, coalitions of pro-arguments and coalitions of con-arguments in debates can be investigated for predicting the overall conclusion of the debates.

Overall, the work carried out in this thesis is useful for carrying out further research in argumentation and opinion mining. Some useful applications are: (1) understanding whether the overall sentiment of a review can be justified argumentatively, (2) understanding how people implicitly express opinions which is not captured by sentiment analysis and stance detection in NLP, and (3) combining argumentation-based features such as coalitions of arguments along with traditional linguistic features as machine learning features for automatically predicting the sentiment.

Bibliography

[1] Stephen E Toulmin. *The uses of argument.* Cambridge university press, 2003.

[2] Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. A review corpus for argumentation analysis. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 115–127, 2014.

[3] Hugo Mercier and Dan Sperber. Why do humans reason? arguments for an argumentative theory. *Behavioral and brain sciences*, 34(2):57–74, 2011.

[4] Ralph H Johnson and J Anthony Blair. The recent development of informal logic. 1980.

[5] Robert J Fogelin and Walter Sinnott-Armstrong. *Understanding arguments: An introduction to informal logic*, volume 4. Harcourt Brace Jovanovich New York, 1978.

[6] Douglas N Walton and David N Walton. *Informal logic: A handbook for critical argument.* Cambridge University Press, 1989.

[7] Guillermo R Simari and Ronald P Loui. A mathematical treatment of defeasible reasoning and its implementation. *Journal of Artificial Intelligence*, 53:125–157, 1992.

[8] John L Pollock. A theory of defeasible reasoning. *International Journal of Intelligent Systems*, 6(1):33–54, 1991.

[9] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Journal of Artificial Intelligence*, 77:321–357, 1995.

[10] Leila Amgoud and Henri Prade. Using arguments for making and explaining decisions. *International Journal of Artificial Intelligence*, 173(3-4):413–436, 2009.

[11] Salem Benferhat, Didier Dubois, and Henri Prade. Argumentative inference in uncertain and inconsistent knowledge bases. In *Uncertainty in Artificial Intelligence*, pages 411–419. Elsevier, 1993.

[12] Claudette Cayrol. On the relation between argumentation and non-monotonic coherence-based entailment. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, volume 95, pages 1443–1448, 1995.

[13] Wiebe Van der Hoek and Michael Wooldridge. Multi-agent systems. *Foundations of Artificial Intelligence*, 3:887–928, 2008.

[14] Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and Pierre Livet. On bipolarity in argumentation frameworks. *International Journal of Intelligent Systems*, 23(10):1062–1093, 2008.

[15] Leila Amgoud, Philippe Besnard, and Anthony Hunter. Representing and reasoning about arguments mined from texts and dialogues. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 60–71, 2015.

[16] Raquel Mochales Palau and Marie-Francine Moens. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*, pages 98–107, 2009.

[17] Safia Abbas and Hajime Sawamura. A first step towards argument mining and its use in arguing agents and its. In *Proceedings of the International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 149–157, 2008.

[18] Andreas Peldszus and Manfred Stede. From argument diagrams to argumentation mining in texts: A survey. *Proceedings of the International Journal of Cognitive Informatics and Natural Intelligence*, 7(1):1–31, 2013.

[19] Kevin D Ashley and Vern R Walker. Toward constructing evidence-based legal arguments using legal decision documents and machine learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, pages 176–180, 2013.

[20] Ngo Xuan Bach, Nguyen Le Minh, Tran Thi Oanh, and Akira Shimazu. A two-phase framework for learning logical structures of paragraphs in legal articles. *Journal of ACM Transactions on Asian and Low-Resource Language Information Processing*, 12(1):1–32, 2013.

[21] Nancy L Green. Towards creation of a corpus for argumentation mining the biomedical genetics research literature. *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics*, page 11, 2014.

[22] Marco Lippi and Paolo Torroni. Argument mining: A machine learning perspective. In *TAFA*, pages 163–176, 2015.

[23] Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *Journal of ACM Transactions on Internet Technology*, 16(2):10, 2016.

[24] Nan Hu, Paul A Pavlou, and Jennifer Zhang. Can online reviews reveal a product's true quality?: empirical findings and analytical modeling of online word-of-mouth communication. In *EC*, pages 324–330, 2006.

[25] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

[26] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation*, pages 19–30, 2016.

[27] Douglas Biber and Edward Finegan. Adverbial stance types in english. *Discourse processes*, 11(1):1–34, 1988.

[28] Douglas Walton. Argument structure: A pragmatic theory (toronto studies in philosophy). 1996.

[29] James B Freeman. *Argument Structure:: Representation and Theory*, volume 18. Springer Science & Business Media, 2011.

[30] Frans H. van Eemeren, Rob Grootendorst, Francisca S. Henkemans, J. A. Blair, Ralph H. Johnson, Erik C. W. Krabbe, Christian Plantin, Douglas N. Walton, Charles A. Willard, John Woods, and David Zarefsky. *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments*. Lawrence Erlbaum Associates, 1996.

[31] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran. Generalizations of dung frameworks and their role in formal argumentation. *IEEE Intelligent Systems*, 29 (1):30–38, 2014.

[32] Leila Amgoud and Srdjan Vesic. On the role of preferences in argumentation frameworks. In *Proceedings of the International Conference on Tools with Artificial Intelligence*, volume 1, pages 219–222. IEEE, 2010.

[33] Trevor JM Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.

[34] Salem Benferhat, Didier Dubois, Souhila Kaci, and Henri Prade. Modeling positive and negative information in possibility theory. *International Journal of Intelligent Systems*, 23(10):1094–1118, 2008.

[35] Didier Dubois and Henri Prade. An introduction to bipolar representations of information and preference. *International Journal of Intelligent Systems*, 23(8): 866–877, 2008.

[36] Nir Oren. *An Argumentation Framework Supporting Evidential Reasoning with Applications to Contract Monitoring.* PhD thesis, University of Aberdeen, 2007.

[37] Sylwia Polberg and Nir Oren. Revisiting support in abstract argumentation systems. In *COMMA*, pages 369–376, 2014.

[38] Claudette Cayrol and M.-C. Lagasquie-Schiex. Coalitions of arguments: A tool for handling bipolar argumentation frameworks. volume 25, pages 83–109, 2010.

[39] Alexander Bochman. Collective argumentation and disjunctive logic programming. *Journal of logic and computation*, 13(3):405–428, 2003.

[40] Søren Holbech Nielsen and Simon Parsons. A generalization of dungs abstract framework for argumentation: Arguing with sets of attacking arguments. In *Proceedings of the International Workshop on Argumentation in Multi-Agent Systems*, pages 54–73, 2006.

[41] Guido Boella, Dov M Gabbay, Leon van der Torre, and Serena Villata. Support in abstract argumentation. In *Proceedings of the International Conference on Computational Models of Argument*, pages 40–51, 2010.

[42] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolarity in argumentation graphs: Towards a better understanding. *International Journal of Approximate Reasoning*, 54(7):pp–876, 2013.

[43] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Gradual valuation for bipolar argumentation frameworks. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 366–377, 2005.

[44] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Journal of Artificial Intelligence*, 128:203–235, 2001.

[45] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. *Journal of Logic and Computation*, 9(2):215–261, 1999.

[46] Joao Leite and Joao Martins. Social abstract argumentation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2287–2292, 2011.

[47] Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza. An argumentation-based approach for automatic evaluation of design debates. In *Proceedings of the International Workshop on Computational Logic in Multi-Agent Systems*, pages 340–356, 2013.

[48] Gerhard Brewka and Stefan Woltran. Abstract dialectical frameworks. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, pages 102–111, 2010.

[49] Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Assumption-based argumentation. In *Argumentation in Artificial Intelligence*, pages 199–218. 2009.

[50] Ho-Pun Lam, Guido Governatori, and Régis Riveret. On aspic+ and defeasible logic. In *Proceedings of the International Conference on Computational Models of Argument*, pages 359–370, 2016.

[51] Alejandro J García and Guillermo R Simari. Defeasible logic programming: Delp-servers, contextual queries, and explanations for answers. *Argument & Computation*, 5(1):63–88, 2014.

[52] Federico Cerutti, Nava Tintarev, and Nir Oren. Formal arguments, preferences, and natural language interfaces to humans: an empirical evaluation. In *Proceedings of the European Conference on Artificial Intelligence*, pages 207–212, 2014.

[53] Ariel Rosenfeld and Sarit Kraus. Providing arguments in discussions on the basis of the prediction of human argumentative behavior. *Journal of ACM Transactions on Interactive Intelligent Systems*, 6(4):30, 2016.

[54] Leila Amgoud and Henri Prade. Can ai models capture natural language argumentation? *International Journal of Cognitive Informatics and Natural Intelligence*, 6(3):19–32, 2012.

[55] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424, 2002.

[56] Swapna Somasundaran. *Discourse-level relations for Opinion Analysis*. PhD thesis, Doctoral dissertation, University of Pittsburgh, 2010.

[57] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation*, pages 1–18, 2016.

[58] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

[59] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, pages 1532–1543, 2014.

[60] Danushka Bollegala, Kohei Hayashi, and Ken-ichi Kawarabayashi. Learning linear transformations between counting-based and prediction-based word embeddings. *PloS one*, 12(9):184–544, 2017.

[61] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150, 2011.

[62] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2017.

[63] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

[64] Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 441–448, 2012.

[65] Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. Recognizing textual entailment: Models and applications. *Proceedings of the Synthesis Lectures on Human Language Technologies*, 6(4):1–220, 2013.

[66] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9, 2007.

[67] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Language Resources and Evaluation Conference*, pages 216–223, 2014.

[68] Lili Kotlerman, Ido Dagan, Bernardo Magnini, and Luisa Bentivogli. Textual entailment graphs. *Journal of Natural Language Engineering*, 21:699–724, 2015.

[69] Kenichi Yokote, Danushka Bollegala, and Mitsuru Ishizuka. Similarity is not entailment - jointly learning similarity transformations for textual entailment. In *Proceedings of the 26th Conference on Artificial Intelligence*, pages 1720–1726, 2012.

[70] Fabio Massimo Zanzotto, Maria Teresa Pazienza, and Marco Pennacchiotti. Discovering entailment relations using textual entailment patterns. In *Proceedings of*

the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment, pages 37–42, 2005.

[71] Parinaz Sobhani, Diana Inkpen, and Stan Matwin. From argumentation mining to stance classification. In *Proceedings of the Second Workshop on Argumentation Mining at the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 67–77, 2015.

[72] Marilyn A. Walker, Pranav Anand, Robert Abbott, and Ricky Grant. Stance classification using dialogic properties of persuasion. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 592–596, 2012.

[73] Swapna Somasundaran and Janyce Wiebe. Recognizing stances in ideological online debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, 2010.

[74] Kazi Saidul Hasan and Vincent Ng. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, 2014.

[75] Valerio Basile, Elena Cabrio, Serena Villata, Claude Frasson, and Fabien Gandon. A pragma-semantic analysis of the emotion/sentiment relation in debates. In *Proceedings of the 4th International Workshop on Artificial Intelligence and Cognition*, 2016.

[76] Elena Cabrio and Serena Villata. Combining textual entailment and argumentation theory for supporting online debates interactions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 208–212, 2012.

[77] Tom Bosc, Elena Cabrio, and Serena Villata. Tweeties squabbling: Positive and negative results in applying argument mining on social media. *Proceedings of the International Conference on Computational Models of Argument*, 287:21, 2016.

[78] Nefise Yaglikci and Paolo Torroni. Microdebates app for android: A tool for participating in argumentative online debates using a handheld device. In *Proceedings of the International Conference on Tools with Artificial Intelligence*, pages 792–799, 2014.

[79] Oana Cocarascu and Francesca Toni. Mining bipolar argumentation frameworks from natural language text. In *Proceedings of the Seventeenth Workshop on Computational Models of Natural Argument at International Conference on Artificial Intelligence and Law*, pages 65–70, 2017.

[80] 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*, pages 111–125, 2017.

[81] Theodore Patkos, Antonis Bikakis, and Giorgos Flouris. A multi-aspect evaluation framework for comments on the social web. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 593–596, 2016.

[82] Christian Stab and Iryna Gurevych. Parsing argumentation structures in persuasive essays. *Journal of Computational Linguistics*, 43(3):619–659, 2017.

[83] Khalid Al-Khatib, Henning Wachsmuth, Johannes Kiesel, Matthias Hagen, and Benno Stein. A news editorial corpus for mining argumentation strategies. In *Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers*, 2016.

[84] Ehud Aharoni, Anatoly Polnarov, Tamar Lavee, Daniel Hershcovich, Ran Levy, Ruty Rinott, Dan Gutfreund, and Noam Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 64–68, 2014.

[85] Marc Vincent and Grégoire Winterstein. Argumentative insights from an opinion classification task on a french corpus. In *Proceedings of the JSAI International Symposium on Artificial Intelligence*, pages 125–140, 2013.

[86] Isaac Persing and Vincent Ng. Modeling argument strength in student essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 543–552, 2015.

[87] Marco Lippi and Paolo Torroni. Context-independent claim detection for argument mining. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, volume 15, pages 185–191, 2015.

[88] Adam Wyner, Jodi Schneider, Katie Atkinson, and Trevor J. M. Bench-Capon. Semi-automated argumentative analysis of online product reviews. In *Proceedings of the 4th International Conference on Computational Models of Argument*, pages 43–50, 2012.

[89] Henning Wachsmuth, Martin Trenkmann, Benno Stein, and Gregor Engels. Modeling review argumentation for robust sentiment analysis. In *Proceedings of the*

25th International Conference on Computational Linguistics: Technical Papers, pages 553–564, 2014.

[90] Haijing Liu, Yang Gao, Pin Lv, Mengxue Li, Shiqiang Geng, Minglan Li, and Hao Wang. Using argument-based features to predict and analyse review helpfulness. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1374, 2017.

[91] Oana Cocarascu and Francesca Toni. Detecting deceptive reviews using argumentation. In *Proceedings of the 1st International Workshop on AI for Privacy and Security*, PrAISe, pages 1–8, 2016.

[92] Maria Paz Garcia Villalba and Patrick Saint-Dizier. A framework to extract arguments in opinion texts. *International Journal of Cognitive Informatics and Natural Intelligence*, 6(3):62–87, 2012.

[93] Mauro Dragoni, Celia da Costa Pereira, Andrea GB Tettamanzi, and Serena Villata. Combining argumentation and aspect-based opinion mining: The smack system. *Journal of AI Communications*, (Preprint):1–21, 2018.

[94] Debanjan Ghosh, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. Analyzing argumentative discourse units in online interactions. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 39–48, 2014.

[95] Filip Boltuzic and Jan Šnajder. Fill the gap! analyzing implicit premises between claims from online debates. In *Proceedings of the Third Workshop on Argument Mining at Annual Conference of the Association for Computational Linguistics*, pages 124–133, 2016.

[96] Filip Boltužić and Jan Šnajder. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 49–58, 2014.

[97] Ivan Habernal and Iryna Gurevych. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2127–2137, 2015.

[98] Rory Duthie, Katarzyna Budzynska, and Chris Reed. Mining ethos in political debate. In *Proceedings of the 6th International Conference on Computational Models of Argument*, pages 299–310, 2016.

[99] Marilyn A. Walker, Pranav Anand, Stephanie M. Lukin, and Steve Whittaker. Argument strength is in the eye of the beholder: Audience effects in persuasion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 742–753, 2017.

[100] Shereen Oraby, Lena Reed, Ryan Compton, Ellen Riloff, Marilyn A. Walker, and Steve Whittaker. And that's A fact: Distinguishing factual and emotional argumentation in online dialogue. In *Proceedings of the Second Workshop on Argumentation Mining at the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 116–126, 2015.

[101] Tom Bosc, Elena Cabrio, and Serena Villata. Dart: a dataset of arguments and their relations on twitter. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, pages 1258–1263, 2016.

[102] Joonsuk Park and Claire Cardie. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argument Mining, hosted by the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 29–38, 2014.

[103] Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the Second Workshop on Argumentation Mining at the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–11, 2015.

[104] Lucas Carstens and Francesca Toni. Towards relation based argumentation mining. *Proceedings of the Second Workshop on Argumentation Mining at the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 29–34, 2015.

[105] Oana Cocarascu and Francesca Toni. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379, 2017.

[106] Ivan Habernal and Iryna Gurevych. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[107] 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*, pages 938–948, 2015.

[108] Vanessa Wei Feng and Graeme Hirst. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 987–996, 2011.

[109] Nancy Green. Manual identification of arguments with implicit conclusions using semantic rules for argument mining. In *Proceedings of the Fourth Argument Mining Workshop at 2017 Conference on Empirical Methods in Natural Language Processing*, pages 73–78, 2017.

[110] Olesya Razuvayevskaya and Simone Teufel. Finding enthymemes in real-world texts: A feasibility study. *Argument & Computation*, 8(2):113–129, 2017.

[111] Maria Becker, Michael Staniek, Vivi Nastase, and Anette Frank. Enriching argumentative texts with implicit knowledge. In *Proceedings of the International Conference on Applications of Natural Language to Information Systems*, pages 84–96, 2017.

[112] Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1930–1940, 2018.

[113] Douglas N Walton. Enthymemes, common knowledge, and plausible inference. *Philosophy and rhetoric*, 34(2):93–112, 2001.

[114] Christopher Potts. Presupposition and implicature. *The handbook of contemporary semantic theory*, 2:168–202, 2015.

[115] Sir William Hamilton. New York: Harper and Brothers, 1861.

[116] Barbara Konat, Katarzyna Budzynska, and Patrick Saint-Dizier. Rephrase in argument structure. *Proceedings of the Foundations of the Language of Argumentation Workshop at the 6th International Conference on Computational Models of Argument*, pages 32–39, 2016.

[117] John Fox and Simon Parsons. Arguing about beliefs and actions. In *Applications of uncertainty formalisms*, pages 266–302. 1998.

[118] Bart Verheij. Accrual of arguments in defeasible argumentation. In *Proceedings of the Second Dutch/German Workshop on Nonmonotonic Reasoning*, pages 217–224, 1995.

[119] Henry Prakken. A study of accrual of arguments, with applications to evidential reasoning. In *Proceedings of the 10th international conference on Artificial intelligence and law*, pages 85–94, 2005.

[120] Khalid Al Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. Cross-domain mining of argumentative text through distant supervision. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1395–1404, 2016.

[121] Foster Provost. Machine learning from imbalanced data sets. In *AAAI 2000 Workshop on Imbalanced Data Sets*, 2000.

[122] I. Tomek. Two modifications of cnn. *IEEE Transactions on System, Man and Cybernetics*, 6:769 – 772, 1976.

[123] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[124] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[125] Vincent Ng and Claire Cardie. Weakly supervised natural language learning without redundant views. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 94–101, 2003.

[126] Zhiguang Liu, Xishuang Dong, Yi Guan, and Jinfeng Yang. Reserved self-training: A semi-supervised sentiment classification method for chinese microblogs. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 455–462, 2013.

[127] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[128] Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Gnter Neumann, Tae-Gil Noh, Sebastian Pado, Asher Stern, and Omer Levy. The excitement open platform for textual inferences. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 43–48, 2014.

[129] Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. *CoRR*, abs/1702.01417, 2017.

[130] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1556–1566, 2015.

[131] Ronald R Yager and Alexander Rybalov. Full reinforcement operators in aggregation techniques. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(6):757–769, 1998.