DETECTION OF CLAIMS AND SUPPORTING EVIDENCE IN WIKIPEDIA ARTICLES ON CONTROVERSIAL TOPICS

by

Waleed Mebane

APPROVED BY SUPERVISORY COMMITTEE:

_____

Dan Moldovan, Chair

_____

Chris Irwin Davis

_____

Vincent Ng

DETECTION OF CLAIMS AND SUPPORTING EVIDENCE IN WIKIPEDIA ARTICLES ON CONTROVERSIAL TOPICS

by

WALEED MEBANE, BS

THESIS

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN

COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

May 2017

ACKNOWLEDGMENTS

I would like to thank my supervising professor, Dan Moldovan, for his guidance and support, enabling me to take up this research.

I would also like to thank Christian Stab for answering my questions about his research, in personal correspondence. And I would like to thank Marco Lippi for helping me to find the available software for kernel learning with support vector machine models.

November 2016

DETECTION OF CLAIMS AND SUPPORTING EVIDENCE IN WIKIPEDIA ARTICLES ON CONTROVERSIAL TOPICS

Waleed Mebane, MS
The University of Texas at Dallas, 2017

Supervising Professor: Dan Moldovan, PhD

This thesis presents the task of argument mining, finding arguments within natural language texts, and reports on experiments combining techniques previously applied in disparate but related domains to the tasks of detecting claims and evidence and predicting the relationship of support from evidence to claim. A large corpus built and labeled at IBM Research, which has been made freely available to other researchers, was used. This thesis demonstrates the usefulness of that resource for argument mining experiments by applying a combination of techniques tried on other data from a different domain together with insights from discourse processing and machine learning. Features from discourse processing applications and a kernel method from machine learning which were expected perform well on the argument mining tasks were tested and compared. In a first published application, the subset tree kernel used with a support vector machine model was found to perform well for all three tasks. Previously other researchers detected claims using a similar tree kernel. The subset tree kernel was augmented with feature vectors as tried for claims by previous researchers and further improved performance was shown.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The topic of this thesis is human arguments found in monological natural language texts. A monological text, as opposed to a dialogical one, is a text characterized by expression flowing in one direction, from the writer or writers to an (often implied) audience. Dialogues or changes of perspective may be embedded within the text. Although argumentation is usually thought of as a dialogue based activity in which arguments are exchanged, with one party seeking to influence the opinions or beliefs of the other or of a third party, arguments are also present in text which involve no interaction with second or third parties; the arguments are being presented to the reader.

A minimal definition of argument, given by argumentation theorist Douglas Walton, in an introductory paper on argumentation is, "a set of statements (propositions) made up of three parts, a conclusion, a set of premises, and an inference from the premises to the conclusion" [1, p. 2]. Such a description is especially amenable to the tasks of this thesis. Given a set of monological natural language texts, we wish to identify key parts of arguments, the conclusions (which we will refer to as claims) and the premises, or some premises (which we will refer to as evidence). In addition we wish to recognize pairs of claim and evidence text fragments which are related such that the evidence acts as a premise for the specific claim; that is, we would like to find links.

Computational approaches to argumentation refer to, besides the simple model of claim, premise, and inference, richer models. For example, besides supporting a claim, propositions can be found which attack claims or which undermine whole arguments. In the experiments of this work we will be finding only the support relation between evidence and claims.

Although there have been decades of research on computational models of natural argument, in the last few years there has been particular interest in "mining" arguments from natural language

texts using the techniques of natural language processing. This means especially the extraction of the arguments from the text as well as other processing. Once arguments have been identified in and extracted from the text and fitted to a formal model, mature computational tools could be used to analyze or evaluate those arguments in order to gain a deeper understanding of the text, for example, or to provide rich textual summaries or to act as a teaching aid for the study of argumentative reasoning in some domain.

The interest in this task has led to the development of some corpora and a body of work exploring methods of mining arguments.

In a particular application relevant to this thesis, IBM Research has started a multiyear project, the Debater Project, with the aim of creating an artificial debate assistant. In the style of their artificial question answering system, Watson, that defeated top players in the question answering game, Jeopardy!, the debate assistant should be capable of competition together with human debaters where the next moves in the debate are to find, not fixed facts, but best arguments.

For their work on their project, IBM has prepared large datasets with text fragments manually labeled as claims or evidence together with the support relations between them. They have made these datasets freely available to other researchers. These datasets have a number of useful properties. They have been taken from Wikipedia articles which are written in a somewhat formal and standard register of English and which have been curated in a sense (i.e., by the Wikipedia user community) so as to have relatively small numbers of spelling and syntactic errors. Several articles on each topic have been selected so that there is enough text for comparisons within and between topics. These datasets are also perhaps the largest available that have been developed for argument mining tasks.

IBM researchers have used these for two argument mining tasks, finding what they have called context dependent claims, the content dependent claim detection (CDCD) task, and finding what they have called context dependent evidence, the context dependent evidence (CDED) task. And

they have reported on their work and results in two papers [2], [3]. These datasets, therefore, offer a reasonable benchmark for the claim detection and evidence detection tasks. Although IBM researchers have successfully completed these tasks, it remains interesting to discover whether there are other methods and freely available tools and insights and methods from other domains of application which could be employed to produce similar results, especially since some of their methods exploited in-house tools that are not open to outside researchers at present. The task is a very challenging one. For example, the reported average precision on the CDCD task using IBM's pipeline approach was 9% (with recall of 73%).

Therefore, it is the position of this thesis, that it is worthwhile to explore alternative approaches. In particular, this research is intended to address the potential benefits of some methods from discourse processing which have worked in other domains or on other similar tasks and the potential benefits of kernel machines. The hypothesis asserted is that discourse processing methods, in particular supervised learning classification with word-pair features and parse tree fragment features and also the tree kernel analog of those approaches will perform well in the IBM dataset.

CHAPTER 2

BACKGROUND

2.1 Argumentation Theory and Artificial Intelligence

Argumentation theory is an area of study including formal, semi-formal, and informal methods for the identification, analysis, evaluation, and production of human arguments, methods which generally go beyond formal logic (see [1, p. 2]). Argumentation theory has its roots in classical rhetoric and in the classical study of human reasoning and human arguments. In the mid-twentieth century, a group of researchers departed from the standard methods and perspectives in the related fields in order to improve teaching about argument and reasoning. These efforts spawned the closely related modern disciplines of informal logic and argumentation theory [1, p. 2]; (see also [4, pp. 33–35]). They developed rich theories of argument, and later collaboration with computer scientists accelerated that progress in some respects by adding new tools and methods which can inform a deeper understanding of the structures of arguments.

A minimal definition of argument, given by argumentation theorist Douglas Walton, in an introductory paper on argumentation is, "a set of statements (propositions) made up of three parts, a conclusion, a set of premises, and an inference from the premises to the conclusion" [1, p. 2].

The conclusion of one argument may take on the role of premise in another argument or a premise of one argument may also be a premise of another argument, so that arguments about a particular topic are interconnected and their interactions may be viewed as a graph, with nodes representing argument components and directed arcs representing relationships between them, such as forms of inferences.

Not all argument components are always explicitly present. For example, an argument with conclusion, "There is no direct link between violent video games and their influence on

children", and with a premise (or evidence), "A meta-analysis by psychologist Jonathan Freedman, who reviewed over 200 published studies and found that the 'vast and overwhelming majority' did not find a causal link also reached this conclusion"[1], does not yet provide an explicit indication of the inference from premise to conclusion. It is clear to the human reader, however, that the pair is able to be straight forwardly connected by an inference step. One simple choice could be something like, "If many published studies agree with the conclusion, then the conclusion is likely valid", which is an additional premise which licenses the inference.

Similarly, there are arguments which omit other premises or which omit the conclusion: (1) "If, on Groundhog day, Punxsutawney Phil emerges from his hole and sees his own shadow, then we can expect six more weeks of winter"; (2) "Today, Punxsutawney Phil emerged from his hole and saw his own shadow." In this case (if "today" is Groundhog day), the conclusion is clear to the human reader though not explicitly stated. Missing argumentative elements are traditionally called enthymemes.

Considering the form of inference suggested in the foregoing examples, we see what looks like a form of modus ponens in propositional logic:

$p \rightarrow q$

p

_____

q

The exception is that we have forms such as "likely" in the formulation of the inferential step in the first example, and "expect" in the second example which qualify the statements (i.e., the if p then q of these examples may not be a universal license). Thus, while the (implicit) conclusion "we can expect six more weeks of winter" fits the form (with expect being embedded in the

[1] These examples were taken from an aforementioned IBM corpus.

proposition) and is, therefore, warranted by its corresponding premise, "There is no direct link between violent video games and their influence on children" appears as too strong a conclusion given its inference warranting premise.

Notice that we chose to write "is likely valid", but that we could have instead simply chosen to write "is valid"; then we would satisfactorily fit the deductive scheme. But we would deviate from the real meaning of the argument, which we understand from our background knowledge from our human experience. Empirical studies do not fit the schema of deductive proofs; inference from empirical evidence is inductive. This highlights the need for an argumentation model that goes beyond formal (deductive) logic, and such deficiencies in formal logic were noticed by philosopher Stephen Toulmin who wrote his "Uses of Argument" in 1958. It was a counter to the then current optimism regarding positivistic application of deductive methods to all kinds of arguments, and in fact he objected to reserving the word deductive only for formal arguments.

Toulmin's model

Toulmin noticed that most useful arguments outside of mathematics are not in the form of formal deductive proofs. Useful conclusions can be drawn, but they do not hold unconditionally. They often need to be qualified.

To account for this and other realities of ordinary human argumentation, in place of the two argumentative elements for the premise-conclusion model, Toulmin gives us six elements: data, qualifier, claim, warrant, backing, and rebuttal. The schematic diagram for this model is shown in Fig. 1.

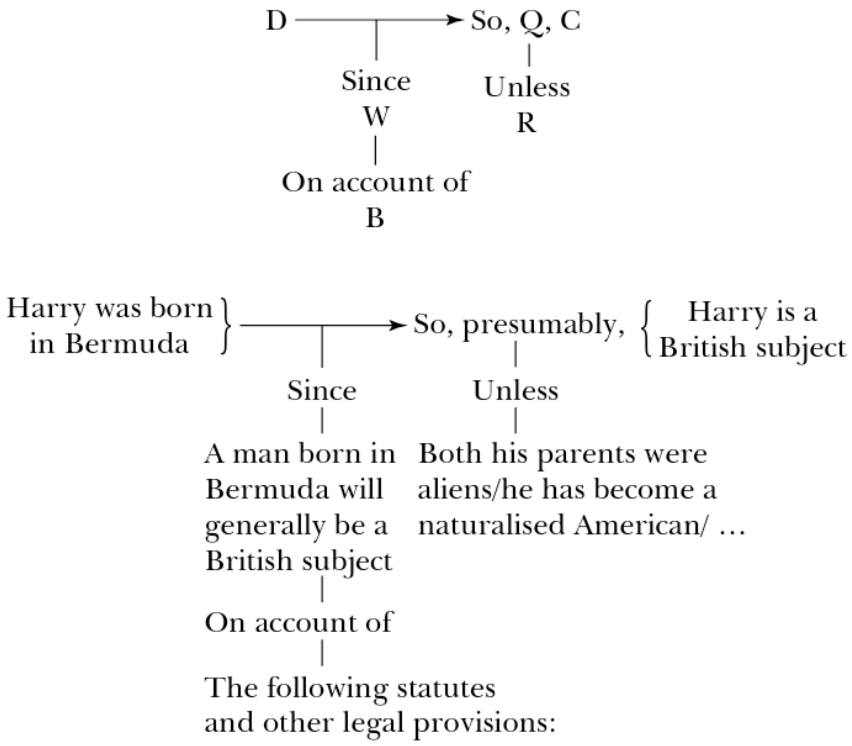

Fig. 1. Diagram of an argument according to Toulmin, from "The Uses of Argument, Updated Edition" [5, p. 97]

Toulmin gives an example in the context of an analogy with legal reasoning [5, pp. 88–89]. In such a setting, when one makes a claim, a challenger may ask for additional information which could establish the truth of the claim; and the same situation is common in other fields of argumentation. Therefore, to meet the challenge, we need some facts which support the claim. In Toulmin's example, shown in Fig. 1, the claim is that Harry is a British citizen. A fact to support that claim is, the *datum*, "Harry was born in Bermuda". From this fact together with background knowledge about laws and statutes and the like relating to citizenship we are able to agree that the fact lends some weight to the claim [5, p. 90]. Toulmin calls the, usually implicit, general rules that license the inference from datum to claim, *warrants*. But Toulmin points out that the warrant might not be a universal grant of license for the inference [5, pp. 91–93]. There may be some circumstances in which the general rule does not hold; examples of these are given as *rebuttals* such as shown in the figure: both of Harry's parents may have been aliens or he has

become a naturalized American citizen. As a result, the warrant needs to be qualified. The qualification given here is "presumably".

With only datum, claim, and warrant, we might think we see the familiar form of syllogism discussed earlier, in which the datum is the minor premise, the warrant is the major premise, and the claim is the conclusion. Toulmin presents his model in order to point out the elements of ordinary arguments that are hidden in the simple form of the syllogism [5, pp. 89, 114–118]. The usual arguments, outside of mathematics, do not hold unconditionally. Therefore, even the warrant can be challenged, and in addition to additional data to support the claim against rebuttals, a challenge to the warrant can be met by *backing*, information which establishes the authority of the warrant. Crucially, the kind of information that can be given to support each type of warrant varies with the domain. So in the case of laws, we can point to various statutes or judicial rulings. On the other hand, in the case of biology, Toulmin points out that a warrant such as, "a whale will be (i.e., classifiable as a) mammal" will be "defended by relating it to a system of taxonomical classification [5, p. 96]."

Therefore, to engage in natural language argumentation without discarding a great deal of its meaning and purpose, we may need to consider forms of representation that go beyond classical logic and which specify elements beyond only conclusions and premises.

Defeasible reasoning and nonmonotonic logic

The Stanford Encyclopedia of Philosophy gives the following explanation of the meaning of defeasible reasoning:

> Reasoning is defeasible when the corresponding argument is rationally compelling but not deductively valid. The truth of the premises of a good defeasible argument provide support for the conclusion, even though it is possible for the premises to be true and the conclusion false. In other words, the relationship between premises and conclusion is a tentative one, potentially defeated by additional information [6, Para. 1].

Forms of reasoning, for example, which fit this description are inductive generalizations (such as in scientific literature), abduction, analogical reasoning, and inferences on the basis of expert opinion [7, Para. 1], [6, Para. 1].

Such methods of reasoning are ubiquitous in all argumentation domains outside of mathematics. In spite of the apparent inappropriateness of formal *deductive* logical inference to capture the meaning and uncertainties of most human arguments (and including forms of human reasoning), researchers have been inspired to develop formal nondeductive accounts of human argumentation. Approaches developed in philosophy and in argumentation theory include, Hamblin's "formal dialectic" (1970) [4, p. 32] and Pollock's (from 1967) whose concepts of rebutting defeaters and undercutting defeaters have been influential in the study of defeasible reasoning in artificial intelligence [8, p. 229], [6, Sec. 5]; (see also [9]). (Indeed Pollock has applied these ideas to artificial intelligence himself.) In the field of artificial intelligence, there has been intense interest in defeasible reasoning and much study including the study of nonmonotonic logics [6, Para. 1].

A nonmonotonic logic is a formal logic which does not have the property of monotonic logics that the truth values of statements in the logic are preserved in all supersets of such statements. That is, if we include new statements into our set under consideration (i.e., we consider a superset), if the new set is inconsistent, we have ways to accommodate the inconsistency, such as by retracting some statements that were previously supposed to be true. In a nonmonotonic logic, some statements (or inferred statements) could become invalid due to the introduction of new statements.

Thus, we might want to say:

(1) (All) birds can fly.
$\forall x$ (bird(x) $\rightarrow$ fly(x)).

(2) A penguin is a bird
bird(penguin).[2]

Now due to our rules of natural deduction, we can infer, fly(penguin). But we might want to insert an exception:

(3) $\forall x$ (flightless_bird(x) $\rightarrow \neg$fly(x)).

(4) flightless_bird(penguin).

Now we can infer a contradiction: $\neg$fly(penguin).[3]
The obvious problem here is the quantifier, $\forall$, in (1). But if we replace it with $\exists$, then it becomes less useful since it does not license an inference in the general case.

A highly visible case of nonmonotonic logic in computer science is the Closed World Assumption in database theory and in logic programming. In their encyclopedia article, Strasser and Antonelli give as an example a travel agent looking up flights from Oshkosh to Minsk in a database and then claiming that there are no direct flights. "How does the travel agent *know*?", they ask [7, Sec. 3.1] [my emphasis]. In fact, the travel agent does not know in a strong sense,

---

[2] A penguin example very similar to this is given as an example of nonmonotonic formalisms such as circumscription, negation as failure, and default negation in Strasser and Antonelli [7, Sec. 3]. It is an example used by other authors, elsewhere, as well.

[3] We could likewise arrive at the contradiction if we had $\forall x$ (flightless_bird(x) $\rightarrow$ bird(x)) instead of (2).

but the claim is based on the assumption that the database is *complete*. New information entered into the database could result in a different, contradictory claim based on the same query.

Work on nonmonotonic logic in artificial intelligence was done by McCarthy and Hayes (1969, situation calculus), McCarthy (1977, circumscription), Reiter (Closed World Assumption and negation as failure in logic programming; 1980, default logic), and others (see [7], [6], [10]).

There are argument based approaches, and of particular interest is abstract argumentation, due to Dung [12]. Dung's approach may be especially interesting in the context of arguments extracted from natural language texts because such arguments will typically not be expressed in terms of logical primitives due to the current limitations of the state of the art of natural language processing techniques. Although there exist domain specific ontologies and broad coverage semantic lexicons, natural language processing techniques do not in general permit full understanding of the semantics of processed texts.

In Dung's approach, a whole argument, and not its components (for example, premises and conclusion), is taken as the smallest unit of consideration. One experimental system, for example, obtains arguments from a debate forum. The texts are assumed to be arguments and natural language processing is used to determine the attack relations between them; Dung's formalism is exploited to summarize the state of the debate, as it stands, so far [11].

Abstract argumentation

Dung introduced the *argumentation framework* (AF) [12]. An AF contains only a set of what are called arguments and a binary relation on that set, called the attack relation.

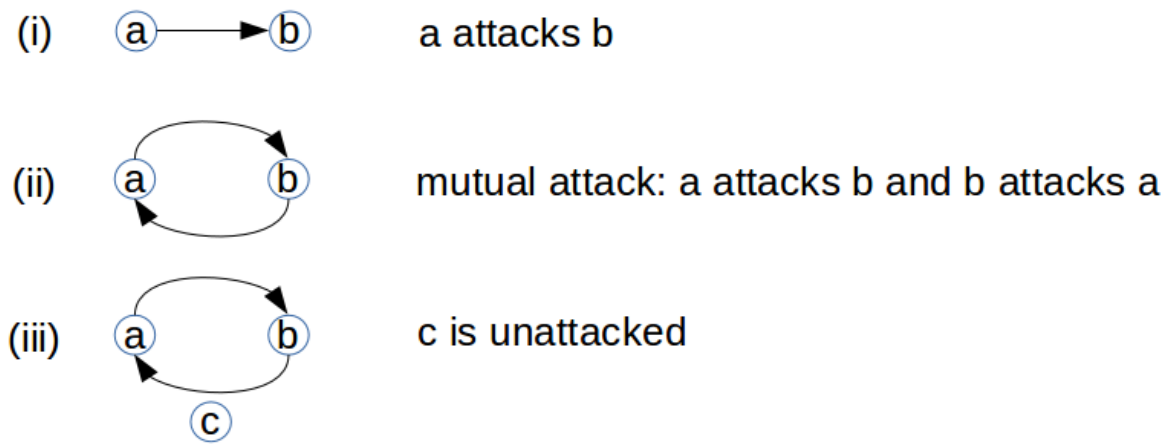

Fig. 2. Simple argument frameworks (AFs)

An AF can be formally represented as a pair <*A*, *R*> with *A* a set of arguments and *R* the attack relation [13, p. 25]. Another way of representing an AF is as a directed graph, with nodes representing arguments and edges representing the attack relation (drawn in the direction from attacker to attacked) [13, p. 25]. Fig. 2 shows some simple argumentation frameworks.
Arguments, so called, are anything that can be represented as attacking or being attacked by other arguments. They are not assumed to take any (e.g., rhetorical or logical) form. So Dung's theory can generalize to domains outside of argumentation, per se, and in fact, he shows how it can be applied to game theory.

Dung formally sets out different types of subset extensions of the sets of arguments in argument frameworks. Each such subset of any of the types corresponds to the intuitive idea that the arguments together in a set are without conflict (i.e., they do not attack each other), and that they in a sense solve external conflicts by collectively attacking their attackers [13, pp. 27, 29–34].

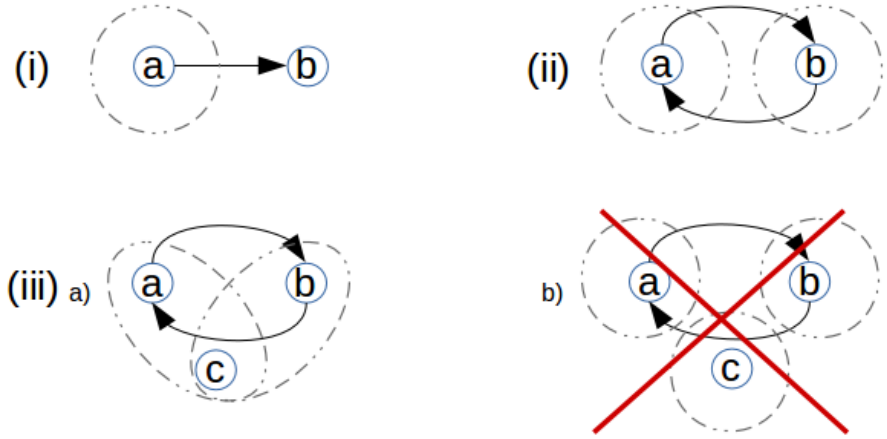

Fig. 3. *Stable semantics* for argumentation frameworks

This can be illustrated by examples. Consider first some simple examples of argumentation frameworks in Fig. 2. In Fig. 3, the extensions given by "stable semantics" (stable extensions) are shown as enclosed in dashed boundaries. A stable extension consists first of all of only arguments none of which attack others in the set. Thus (a) and (b) cannot be in an extension together in any of the three argumentation frameworks from Fig. 2. Secondly, there is an attack from an argument in the set to all arguments outside of the set. (This second requirement leads to larger sets; for example, in AF (iii), (c) will not be in an extension by itself although it is not attacked, but instead will join in an extension with either (a) or (b).)

Two possible notions of justification (but not the only two) are skeptical justification and credulous justification. An argument is skeptically justified if it is a member of all extensions of an AF (under a given semantics), for example, argument c. An argument is credulously justified if it is a member of at least one such extension (e.g., arguments (a) and (b) in AFs (ii) and (iii)) [13, p. 32].

Abstract argumentation can obviously allow us to draw interesting conclusions without knowing the contents of arguments but only knowing the attack relation on the set of arguments. And, further, it seems more or less to correspond to our intuitions regarding how we might evaluate arguments if we are given such limited information about them. The precise formal descriptions

and explanations of the mathematical properties given by Dung and others have enabled the development of computer programs to do such evaluations.[4] A previously mentioned example using natural language processing will be looked at in more detail in the next chapter.

Argumentation frameworks have been extended, in particular, to bipolar abstract argumentation systems, which include, in addition to the attack relation, a support relation (see [14]). Such systems have been formally specified and implemented in software. So that if support relations are also available, a formal model with greater expressiveness is also available for use.

Defeasible logic programming (DeLP), which takes advantage of negation as failure and defeasible rules, is also available. Primitives in this formalism are the premises, conclusion, and inference rules as in logic programming, expressing *and*, *or*, *not*, *material implication* (expressed by "rules" in logic programming, called "strict rules" in DeLP), and *negation as failure*, and with the addition of *defeasible rules*. Success of a query is decided according to dialectical analysis (using argumentation) to determine whether the statement is *warranted* (i.e., if it involves defeasible rules, then it survives possible defeats) [15]. So, if one has been able to identify the internal components of arguments, their premises, conclusions and warrants (i.e., defeasible inference rules), then a system employing DeLP is available for evaluating those arguments.[5]

Argumentation schemes

Argumentation schemes are abstract argument forms that capture patterns of argumentation commonly found in various argumentative contexts, such as legal and scientific argumentation

[4] As an example, such a method of evaluation is available in the AIFdb online argument visualization interface via the "Dung-o-Matic" subsystem.

[5] Evaluation of arguments using DeLP is made available in the AIFdb online argument visualization tool through an online version of software for computational logic, called Tweety.

[1, p. 7]. An example of an argument scheme, "Argument from Example", is given by Cabrio, Tonelli, and Villata [16, p. 3]:

> Argument from Example
> Premise: In this particular case, the individual *a* has property *F* and also property *G*.
> Conclusion: Therefore, generally, if *x* has property *F*, then it also has property *G*.

Of this argumentation scheme, they write:

> This scheme is one of the most common types of reasoning in debates since it is used to support some kinds of generalization.
> The Argument from Example is a weak form of argumentation that does not confirm a claim in a conclusive way, nor associates it with a certain probability, but only gives a small weight of presumption to support the conclusion [16, p. 3].

It is particularly interesting to take note of two things here. One is that this is not a deductive argument. It is based on the premise-conclusion model and not the Toulmin model; even a warrant to license the inference from premise to conclusion is missing. Proceeding from there, the second thing to note is that it is a weak form of argument. It is not necessarily correct. The conclusion is qualified by the qualifier, "generally", but even so, it is not clear that it is generally correct. In fact, that is not what is being expressed by the scheme.

Instead what we have in argumentation schemes is a semi-formal description of actual patterns of argument which can be (in fact, often are) used to provide presumptive support for conclusions. Bench-Capon and Atkinson illuminate the motivation and purpose behind argumentation schemes in the following way:

> Walton's notion of argumentation schemes developed out of his long standing interest in fallacies. In particular there is a need to account for the fact that many of the well known fallacies often seem to be used quite properly to support positions in everyday argumentation. Thus, although fallacies such as the Argument from Ignorance, Argument from Expert Opinion and various forms of abductive argument, are strictly speaking logical fallacies, they also seem, in the right circumstances, to be accepted as justifying their conclusions. Thus the fallacy can

> be seen not so much in the form of the argument, but rather in the improper use of the argument. So, the notion of argumentation schemes was developed in order to explain the proper use of such arguments: argumentation schemes represent stereotypical patterns of reasoning which can presumptively support conclusions when used properly but which also have the possibility of being fallacious when improperly used [17, p. 103].

Walton's argumentation schemes are accompanied by *critical questions*. Some examples selected by Cabrio, Tonelli, and Villata, for Argument from Example, are as follows:

CQ1: Is the proposition presented by the example in fact true?
CQ2: Does the example support the general claim it is supposed to be an instance of?
CQ3: Is the example typical of the kinds of cases that the generalization ranges over [16, p. 3]?

The critical questions suggest ways in which the argument (fitting the scheme) could be supported or rebutted.

The usefulness of argumentation schemes to the task of argument extraction from natural language texts is that they catalog patterns likely to be discovered in argumentative texts. Notice that much is left implicit in the form of the argumentation scheme given as an example, and, in fact, that is so for others as well. Such missing elements could likely complicate the task of identifying arguments according to more or less straight forward attempts to fit the premise-conclusion model or the Toulmin model, expecting to reconstruct examples of a kind of defeasible modus ponens. But if one can detect text that fits the form of an argumentation scheme, due to the presence of elements such as the name of an expert, syntactic indications of reported speech, the name of a subject domain, or other scheme specific features, then without explicitly finding defeasible rules, warrants, backing, or the like, one may be able to detect an argument. Furthermore, one may infer the missing elements based on the semantics of the argumentation scheme. Feng and Hirst [18, p. 987] propose to do exactly that, and to that end, they experiment with classifying arguments by scheme.

Argumentation schemes were not exploited for the empirical potion of this work, but this author expects that the most successful approaches to detecting argument components will make use of argumentation schemes. Studies using argumentation schemes with the goal of extracting arguments will be outlined in the next chapter.

2.2 Argumentation and Natural Language Processing

Argument mining

A good definition of argument mining, also called argumentation mining is given by Lawrence and Reed: "Argument Mining is the automatic identification of the argumentative structure contained within a piece of natural language text [19, p. 127]." Possible applications suggested for argument mining, as listed by Peldszus [20, p. 88] include: "improving document summarization (Teufel and Moens, 2002) [21], retrieval capabilities of legal databases (Palau and Moens, 2011) [22], opinion mining for commercial purposes, or also as a tool for assessing public opinion or political questions." Simon Wells [23] suggests a broad application, to comparing arguments from recently digitized non-fiction works from, e.g., the Google Books Project, with past and current arguments on the topic in order to enhance the knowledge discovery process.

Lawrence and Reed [19, p. 127], and also [24, p. 80] point to Moens et. al. [25] as a start to research into argument mining (in this case using supervised machine learning) and Palau and Moens [22] is often cited. Both of these investigations were targeted primarily at the legal domain. Other early work was in the domains of online reviews and debate [26], [11]. The 2011 work by Palau and Moens is rare in that it attempted to demonstrate a more or less complete argument mining pipeline [27, p. 13]. Therefore, a description of their work can be informative as to the tasks generally considered as constituting a complete system as well as to an example of approaches taken.

They used a premise-conclusion model, considering arguments to be composed of one conclusion supported by one or more premises; however, they considered compounds of

arguments such that one argument supports the next related argument in a tree structure embodying forms from the pragma-dialectic theory of argument by van Eemeren and Grootendorst [28] (see Fig. 4). I.e., arguments can be considered which act as premises to other arguments.

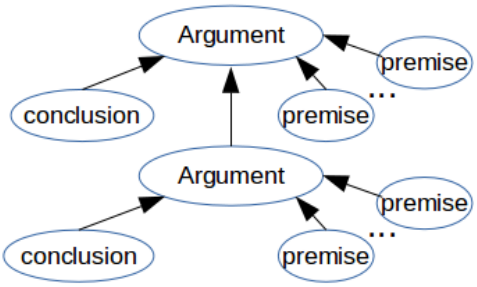

Fig. 4. Compound Argument [22, Fig. 1]

The first step typical of argument mining is *argument sentence detection* [22, p. 11], [27, p. 6][6]. In this step, the authors use machine learning classifiers in an attempt to classify sentences as argumentative or non-argumentative. Once sentences have been so classified, it is still not known what the boundaries of *specific* arguments are; that is, in which part of the text does one whole argument reside (e.g. all of its premises and its conclusion), where is the next argument, and so on. This is the *argument segmentation* problem. There is no inherent limit to the distance within a document between argument components or the length of a complete argument in the document [22, p. 15]. A whole argument in this case means a conclusion together with all of its (explicit) premises. A segmentation step is not explicitly implemented in all approaches. For example, one could attempt to find all conclusions in a text, then find all premises in a text, then, finally, try them in pairs to discover the argumentative support relation that applies between them (or no relation), sidestepping the segmentation problem, as in [29] and [30]. If the

[6] Lippi and Torroni, who wrote a paper [27] seeking to define the major tasks of argument mining and encourage a common language in writing about it, consider this task as a sub-task of "argument component detection".

document is not short, there may be many pairs to compare. We will return to Palau and Moens' approach to the problem after considering the next step, *argument component detection*, which they call "argument proposition classification" [22, p. 13], [27, p. 6].

They use two machine learning classifiers operating at the level of clauses of sentences (separated using a parsing tool), to classify each (argumentative) clause as either a premise or a conclusion. They use a list of features, including some indicators specific to the legal domain. Another segmentation problem, not tackled by Palau and Moens, is called (by Lippi and Torroni), *argument component boundary detection*, the detection of the locations of the exact start and end points of argument components (e.g., conclusion or premises). By classifying at the clause granularity, Palau and Moens seem to sidestep the problem, but note that conclusions or premises can be longer than a single clause, and could also possibly be shorter by excluding some phrase(s) of the clause. Palau and Moens solve this, in part, together with the argument segmentation problem and together with the next step, *argument structure prediction*[7].

In this step relationships between argument components are sought in order to detect or predict the whole internal structure of each argument. In the premise-conclusion model, which they use, this means the argumentative support relations between premises and their conclusion. Since their model also contains compound arguments, it will include the tree structure of the resulting compounds. Palau and Moens use a corpus of legal texts of the European Court of Human Rights (ECHR). These documents are written using a standardized (by convention) legal language, forms of reasoning, and structure of argumentation [22, p. 9]. Palau and Moens adopt a solution especially appropriate to the legal domain as represented by the ECHR corpus. By manually analyzing a held-out section of the corpus, they constructed a context-free grammar productive of all forms of argumentation structure they noticed present in the texts using premise

[7] Called "detection" by Palau and Moens, "prediction" by Lippi and Torroni. Lippi and Torroni explain that they use "prediction", as is traditional in machine learning, because it refers to something that is not tangibly present in the text, that is, a link or relationship between parts [27, pp. 9–10].

and conclusion clauses detected in the last step as two kinds of token (among others). In this way, after having tagged the elements of the document with premise and conclusion tags and with various other features of argumentative discourse in the legal domain, they could parse the document to create a complete tree of the argument structure predicted for all parts of the document.

This parsing approach is very interesting in that it solves elements of two segmentation problems and the argumentative relation prediction problem. Palau and Moens report 60% accuracy when using their grammar to parse texts from the ECHR corpus [22, p. 18]. Unfortunately it would likely be very challenging to produce similar successful grammars for parsing less standardized texts. It is possible that such a grammar would not generalize even within the same domain to texts not using similar argumentative and linguistic conventions, but it is obviously a promising approach, especially for the legal domain. The similarity with the goals of discourse parsers is also notable. Observations from a general theory of argumentative discourse together with statistical methods could be hoped to eventually enable the development of general-purpose (or broad coverage) argumentative discourse parsers.

In addition to the work with the ECHR legal corpus, Palau and Moens also experimented on a corpus of heterogeneous texts from 19 newspapers, 4 parliamentary records, 5 court reports, 6 magazines, and 14 online sources such as discussion boards – material from a wide variety of genres [22, pp. 8–9]. The arguments of these texts were analyzed and labeled using the Araucaria argument diagramming tool and formed the initial Araucaria corpus.

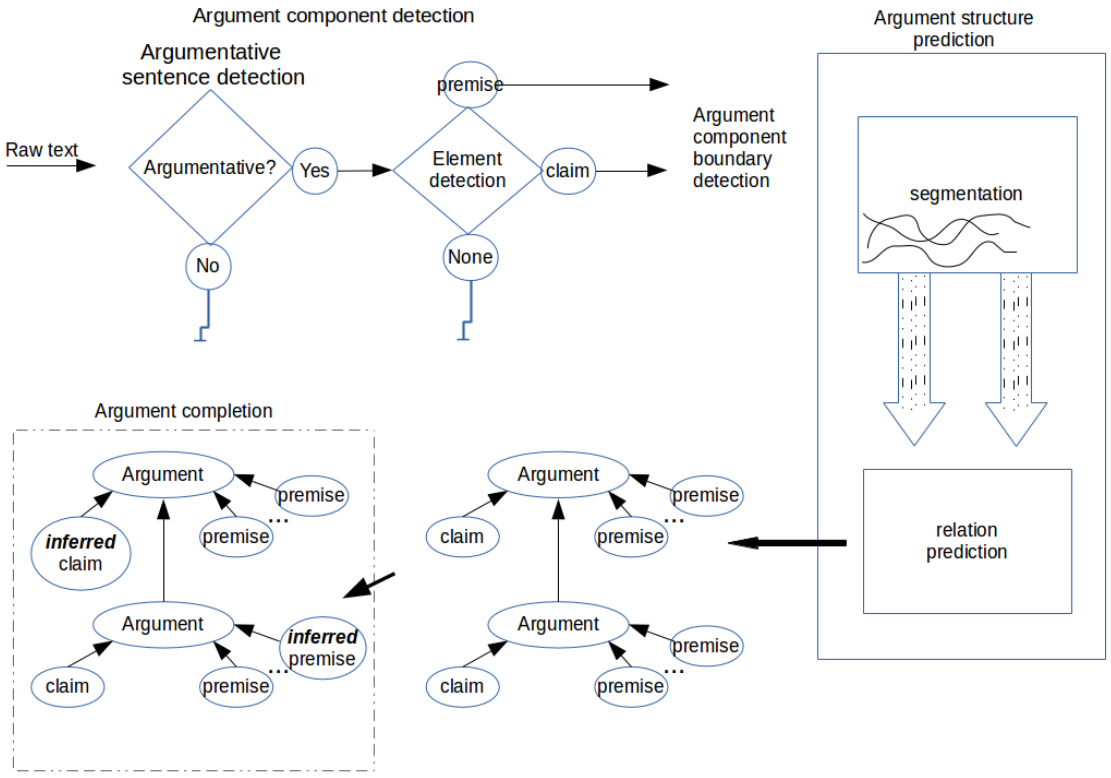

Fig. 5. Illustration of an argument identification and analysis pipeline, derived from [27, Fig. 2] and [22, Fig. 1]

They applied their argumentative sentence detection classifiers to texts from this corpus achieving 73% accuracy (vs. 80% for the legal corpus) and demonstrating that many of their features and their machine learning classifiers generalize to domains beyond law.

Shown in Fig. 5 is the prototypical model of an argumentation pipeline as described by Lippi and Torroni [27], and augmented from my descriptions. This shows the steps of *argument component detection* and *argument structure prediction*, where component detection is divided into two sub-tasks, *argumentative sentence detection* and *argument component boundary detection*. The argumentative sentence detection task may consist of two steps, as in the discussed work by Palau and Moens [22], where first sentences are classified as either argumentative or non-argumentative and then specific components (such as conclusion and

premises) are identified; or both parts of the task may be completed as one step, for example, with a multiclass classifier as in [29] or with separate classifiers as in [2] (claims/conclusions) and [3] (evidence/premises) (see [27, p. 6]). A final possible task could be argument *completion*: filling in missing/implicit elements, *enthymemes* [27, p. 4], such as in a previously mentioned approach, by Feng and Hirst [18], using argument schemes.

Most argument mining architectures use the premise-conclusion model, but Habernal, et. al. [31] recently used the Toulmin model to annotate a corpus. Lippi and Torroni [27, p. 10] and Palau and Moens [22, p. 5] suggest that the premise-conclusion model is generally more suitable (than the Toulmin model) due to the likely difficulty of fitting more elements (i.e., warrants, qualifiers, backing, and rebuttals) when many of these elements are usually left implicit in natural language texts.

Argument mining – Related areas – Discourse processing

Cabrio, Tonelli, and Villata [16, p. 5] provide the following definition: "In Linguistics, discourse analysis is a broad term used to cover linguistic phenomena occurring beyond the sentence boundary, usually emerging from corpus evidence." Therefore, argument mining is a discourse analysis, or discourse processing, task, as it aims to detect linguistic phenomena occurring beyond the sentence boundary. Approaches taken by researchers of discourse processing have also been tried by argument mining researchers, especially the use of discourse markers as features for segmentation and as indicators of the presence of argumentative relations. In fact, argument relations are reasonably considered as types of discourse relations.

In one example of overlap in the research programs of argument mining and of discourse processing, a study by Cabrio, Tonelli, and Villata [16] investigated the hypothesis that there is a direct mapping from some previously defined discourse relations annotated in the Penn Discourse Treebank to equivalent argument schemes.

The Penn Discourse Treebank (PDTB) is a corpus created from the million word Wall Street Journal corpus by the additional manual annotation of discourse relations. Relations are organized into a hierarchy with broad classes of relations at the top level, and the fine grained relations themselves at the bottom level, such that some relations differ only in the order of their arguments. For example, the relation *Cause*, under the class CONTINGENCY, has sub-types, *reason* and *result*. The relation, *reason*, applies when the second text span, Arg2, describes a situation that is the cause of the first, Arg1; and *result* applies when the situation of Arg2 is the result of Arg1.

Cabrio, Tonelli, and Villata considered the five argument schemes: Example, Cause to Effect and Effect to Cause, Practical Reasoning, and Inconsistency. They hypothesized a mapping of the five argument schemes on to PDTB discourse relations [16, p. 2], based on apparently equivalent or very similar semantics as understood from their definitions. For example, they hypothesize that the argument scheme, Argument from Cause to Effect, corresponds to the PDTB discourse relation, *reason*, and Argument from Effect to Cause, to the relation, *result*.

Argument from Cause to Effect is described as "a predictive form of reasoning that reasons from the past to the future, based on a probabilistic generalization." Argument from Effect to Cause is described as, "from the observed data to a hypothesis about the presumed cause of the data (abductive reasoning)" [16, p. 9].

As an example (together with mapping on to the premise-conclusion structure of the hypothesized argument scheme):

(reason)
CONCLUSION: (Arg1) She pleaded guilty
PREMISE: **because** (Arg2) she was afraid of further charges

(result)
PREMISE: (Arg1) Producers were granted the right earlier this year to ship sugar and the export licenses were expected to have begun to be issued yesterday
CONCLUSION: **as a result** (Arg2) it is believed that little or no sugar from the 1989-90 crop has been shipped yet [16, p. 9][their emphasis[8]]

As mentioned previously, much can be left implicit in the form of argument schemes. The form of inference here is implicit. Human annotators were employed to judge the fit of a randomly selected set of text pairs for the considered discourse relations with the hypothesized matching argument schemes.

In this case (and also by a measure of inter-annotator agreement) 80% of the selected samples labeled as *reason* and *result* in the PDTB were annotated as positive instances of the corresponding argument scheme [16, p. 10]. For the other argument schemes, Argument from Example, Practical Reasoning, and Argument from Inconsistency, the correspondences were, respectively, 100%, 70%, and 60% [16, pp. 8–12].

This high level of correspondence suggests that a large corpus, the PDTB, could be used as a source of examples of some argument schemes for future research. It shows a close relationship between one aspect of discourse theory and argumentation theory.

Another research experiment strongly influenced by work from discourse processing was carried out by Stab and Gurevych [29]. It is one of four sources (including two papers by IBM researchers) most informing the empirical portion of this research and, therefore, it will be covered in more detail in order to provide background for the design of the experiments done for this thesis.

---

[8] The highlighted phrases are discourse markers which were either present in the original WSJ text or which were added by PDTB annotators as a best guess at how to make the implicit relation explicit with such a marker.

Stab and Gurevych aim to identify argumentative discourse structures in persuasive essays written by students, as an aid to teaching, for example: enabling automatic feedback about the argumentative quality of the writing as compared with the limited feedback available in current systems such as spell-checking, grammar checking, and the checking of some stylistic properties [29, p. 46]. They use a premise-conclusion model. The persuasive essay writing style leads to a certain typical organization of argumentative elements in the text. A class of claims called major claims, of which there is one per essay, always appears in the introduction or conclusion of an essay. Claims, in general, are frequently at the beginning or ends of paragraphs [29, p. 50]. Their unit of analysis is the clause. And a 'covering sentence' is the sentence containing a clause.

Therefore they define features such as the position of the covering sentence in the essay and also boolean features indicating whether the argument component is in the introduction, conclusion, and first or last sentences of a paragraph [29, p. 50]. Using a clause as their unit of analysis (which was also done by Palau and Moens [22]), enables them to take account of words or punctuation marks before the clauses (when the clause does not begin the sentence; first tokens, otherwise) that may act as discourse markers. These structural features were shown to perform well for the classification of argument components. They do not do any boundary detection, so argument components detected by their system will be single clauses. (Although boundaries potentially different from a single clause were also labeled for each argument component.)

In addition to structural/position-based features, they also use n-grams, verbs, adverbs, and modals. Of these features, they write, "… certain verbs like 'believe', 'think' or 'agree' often signal stance expressions which indicate the presence of a major claim [29, p. 50]." For the same purpose they include features denoting reference to the first person which might coincide with statements of the personal stance of the author and indicate the presence of the major claim. These features may be less useful (or not at all useful) in domains in which writing is rarely in the first person, such as in Wikipedia articles. But the mentioned verbs could still occur in reported speech indicating the stance of an expert or of the authors of a scientific study. Of the

other of these features, they write, "…adverbs like 'also', 'often' or 'really' emphasize the importance of a premise" and "[m]odal verbs like 'should' and 'could' are frequently used in argumentative discourse to signal the degree of certainty when expressing a claim."

They include production rules from the parse trees as Boolean features, which were proposed by previous researchers, [32], as a way to capture syntactic characteristics of a component [29, p. 50]. Similar to other discourse approaches, and as with Palau and Moens, they include the presence of discourse markers as Boolean features. While Palau and Moens used some domain specific discourse markers which signaled argumentative transitions in ECHR documents, according to the writing conventions typical of such documents, Stab and Gurevych use general markers derived from the Penn Discourse Treebank annotation manual. They exclude from their list those discourse markers which do not indicate argumentative discourse (as examples, they give, "markers indicating temporal discourse" [29, p. 50]).

They find that discourse markers are not useful for separating argumentative from non-argumentative clauses, but they help distinguish claims and premises. The sense of such markers is often ambiguous. These are words and phrases such as because, since, although, however, therefore, for example, and, but, and with.
Jurafsky and Martin give a useful example of such ambiguity, including the use of the word 'with'. It appears as a discourse marker in the first sentence in the following example and in an unrelated use in the second:

[1] **With** its distant orbit, Mars exhibits frigid weather conditions
[2] We can see Mars **with** an ordinary telescope [33, p. 693]

Likewise Stab and Gurevych note that "'since' indicates temporal properties as well as justifications, whereas 'because' also indicates causal links [as opposed to giving justification] [29, p. 51]."

In addition, in general, the vast majority of instances of discourse relations are not marked with explicit discourse markers. Marcu and Echihabi found that for a relation relevant to argumentation, the CAUSE-EXPLANATION-EVIDENCE relation, only 79 out of 307 (26%) were marked by a discourse marker in a commonly used discourse processing resource, the corpus of Rhetorical Structure trees (RST Discourse Treebank) [34, p. 368].

Furthermore, Stab and Gurevych find that in their corpus, written by students, "discourse markers are either frequently missing or misleadingly used." They see this as evidence that students could benefit from systems that would assist them in using discourse markers correctly and contrast the essays in their corpus with the documents in the ECHR corpus used by Palau and Moens, with which discourse markers were used in a consistently predictable way [29, p. 54]. However, in spite of the limitations found by Stab and Gurevych with regard to discourse markers as indicators of the presence of argumentative content, they have shown usefulness in other domains besides the legal domain where although they may be frequently missing, they show high precision as indicators (see [19, p. 129], where they are used for finding connections between propositions; and [2]; and [3]).

In addition to finding premises and claims, Stab and Gurevych also attempt to predict the support relationships from premises to claims. They again use a machine learning classifier for supervised learning. The features they use on claim-premise (and also premise-premise) pairs are similar to their features used for the earlier component classification, but one other very interesting feature, word-pairs, is added. The use of word-pairs is an attempt to capture implicit discourse relations. Marcu and Echihabi [34] pioneered the use of word-pairs for this purpose, together with a Naïve-Bayes classifier, and were able to distinguish the (argumentatively relevant) CAUSE-EXPLANATION-EVIDENCE relation (a relation class including several, related, more fine grained relations) with accuracies of 74%+ [34, pp. 372–373].

Marcu and Echihabi give an example of their hypothesis for the CONTRAST discourse relation as:

John is *good* in math and sciences.
Paul *fails* almost every class he takes [34, p. 371].

Our background knowledge tells us that *good* and *fails* appearing as a pair in adjacent text/discourse units is a good indicator of a CONTRAST relation. Thus by pairing words from each two text spans from a large corpus in which the relations are already known, a supervised (in the case of their study, semi-supervised) learning algorithm could be used to learn the needed 'background knowledge' (i.e., which pairs of words are indicative for a given discourse relation). Stab and Gurevych find that word pairs perform well as features for classifying support/non-support argument component pairs, with a macro F1-score of 0.68 [29, p. 53].

The number of argument component pairs in their training data was only 6,330 pairs; whereas, Marcu and Echihabi used up to 4,771,534 examples (theirs were imperfectly labeled, noisy, examples) [34, p. 372]. Therefore several discourse features have been shown to be useful for argument mining tasks, the position of argument units within the text, discourse markers, and word-pairs for the discovery of implicit discourse relations (in this case the argumentative support relation).

Argument mining – Related areas – Debating Technologies

Debating Technologies is an emerging, closely related area being pioneered by IBM Research[9], and it was also, for example, the topic of a special track ("Special Track in Debating Technologies") at the 3rd Workshop on Argument Mining[10] at the Association for Computational Linguistics 2016 conference. A multi-year research project at IBM Research aims to develop an artificial debate assistant. The core technology used for tasks associated with the project, described in recent papers, is argument mining, and, as previously mentioned, large corpora

[9] See http://researcher.watson.ibm.com/researcher/view_group.php?id=5443

[10] http://argmining2016.arg.tech

suitable for some argument mining tasks have been compiled and annotated by IBM and made freely available for the benefit of the research community.

In two recent papers, [2] and [3], the authors describe the identification of, respectively, ('context-dependent') claims, and premises (called evidence, or more precisely 'context-dependent evidence') using a pipeline of tools and machine learning classifiers. In the latter paper, context-dependent evidence (CDE) is defined only in relation the claims it supports; therefore, although candidate CDE is identified separately during a step of the pipeline (corresponding to *argument component detection* in the typical argument pipeline), the prediction of the argumentative support relation between claims and (candidate) pieces of evidence is considered as integral to the task of identifying CDE.

A complete system for extracting arguments from (either of the two) IBM corpora would, therefore, detect context dependent claims (CDC) and context dependent evidence (CDE). These are the two tasks tackled, respectively, in the two mentioned papers.

The empirical portion of this thesis addresses the same two tasks and uses the second of two corpora built by IBM as a resource for these tasks, the one which was also used in the work described in [3], and it builds on the work described there and in the first of the two mentioned papers and in a third work, [35], by an independent set of researchers (at the University of Bologna), all of which will be described in this section in order to provide background for the empirical work of this thesis.

There are obvious equivalences between the typical argument mining tasks described earlier and the tasks taken up for the IBM research. The work clearly uses the premise-conclusion model. They use pipeline approaches. There is an argument component detection step, including, as we will see a boundaries detection step, and there is a relation detection step, among others. Their work differs from other argument mining works primarily in the field of their concerns, but the tasks taken up are clearly the tasks typical of argument mining.

They are interested in finding argumentative material on a particular topic while ignoring arguments irrelevant to the topic at hand, even though the system should perform as well over a broad range of topics. They are not necessarily interested in complete arguments or the structures of such arguments. It is sufficient to be able to adequately (or preferably better than adequately) support claims which are relevant to debating on a specific topic. Enthymemes need not be made explicit. The argument schemes used or the form of inference may be, likewise, of low immediate importance. Evidence supporting a claim does not have to be a part of the same argument in a discourse representation of the text. In fact, it could be found in other documents (possibly by other authors) related only by topic (but in the current corpora, context-dependent evidence, CDE, are only labeled within the same document).

So although discourse processing techniques are still relevant to the tasks, at a top conceptual level, the tasks more closely resemble information retrieval than discourse processing. With that in mind, consider the definitions of context dependent claim and context dependent evidence:

Context Dependent Claim (CDC) – a general, concise statement that directly supports or contests the given topic.
Context Dependent Evidence (CDE) – a text segment that directly supports a claim in the context of the topic. [36, pp. 64–65]

Levy, et. al [2] reported on their work on the context dependent claim detection (CDCD) task. For their task, they used a dataset described, in overview, earlier. Thirty-two topics were selected at random from among debate motions at Debatabase[11] and so were varied in terms of domain of knowledge [2, p. 1491].

[11] http://idebate.org/debatabase

TABLE I
EXAMPLE SET OF CLAIMS DETECTED BY PIPELINE OF LEVY, ET. AL., TAKEN FROM [2, P. 1490]

Topic: The sale of violent video games to minors should be banned		
S1	Violent video games can increase children's aggression	V
S2	Video game addiction is excessive or compulsive use of computer and video games that interferes with daily life	X
S3	Many TV programmers argue that their shows just mirror the violence that goes on in the real world	X
S4	Violent video games should not be sold to children	X
S5	Video game publishers unethically train children in the use of weapons	V

TABLE I shows an example given by Levy, et. al., together with statements which should and should not be considered as CDCs. They explain that S2 simply defines a relevant concept, S3 is not relevant enough to the given topic, and S4 merely repeats the given topic in different words [2, p. 1490].

Their system consisted of a pipeline of parts: a sentence component, with the task of selecting the 200 best sentences; a boundaries component, in two parts, with the task of finding exact boundaries of candidate CDCs within or across the previously selected sentences; and a ranking component, with the task of selecting the 50 best CDCs using information obtained from both previous components. They used a supervised learning approach with machine learning classifiers.

Features for the sentence component were of two types: *Context features*, which in this case means indicators of relation to the *topic*; and *Context-free features*, which they define as relying "solely on the context of the candidate sentence, aiming to capture the probability it includes a 'claim like' statement." Context features were similarity scores. Context-free features were a subjectivity score, a sentiment score, and features obtained from an in-house English Slot Grammar (ESG) parser, which was also used to identify subjects of each sentence for computing one of the similarity scores. A mixed type feature (set of features) was generated by an extension of the Sequential Pattern Matching algorithm [37], which found highly relevant patterns of attributes from parsers, topic words, and an automatically learned lexicon of "claim words" [2, p. 1493].

The boundaries component aims to filter out the non-claim portions of sentences by finding the most likely exact boundaries, which is done in a coarse and then a fine-grained filtering step. In the final step, candidate CDCs passing through the filter are ranked using a machine learning classifier based on scores from the previous components and by again applying some of the features from the first step, with the expectation that the classification will be more accurate after the non-claim material (i.e., noise) has been filtered out.

By this approach of successive refinement, they were able to address two major challenges of their data, (1) the subtlety of the distinctions between claim and non-claim material, and (2) the imbalance in the amount of positive and negative examples (on average only 2% of sentences include claims) [2, p. 1490]. On average (across topics, which were separately classified as folds of cross-validation), they were able to achieve a precision of 0.12, where average maximal precision should be about 0.6 (because there are only 30 CDCs per topic on average, and they select the best 50 candidates). Random selection precision was 0.00008. (Although the task is different, inter-rater agreement on acceptability of candidate CDCs during the document labeling was 0.39). Therefore, they presented a good result for a very challenging task and have also left room for improvement.

For the context dependent evidence detection (CDED) task, *context-dependent* is defined as, "considering the relation of the candidate to the claim and topic." [3, p. 443]. The researchers used supervised learning for all components. CDE were detected by a pipeline of 4 components: *a coherence component*, with the task of scoring spans of 1-3 sentences (considered as initial candidate CDE) as to whether they can stand alone as coherent; an *evidence characteristics component*, with the task of scoring the initial candidate CDE according to whether it has characteristics of evidence of one of the specific types (i.e., it is an 'evidence like' span); a *context-dependent component*, with the task of classifying [topic; claim; candidate CDE] triplets as to whether the candidate CDE could be used to support the claim in the context of the topic; and a *claim selection component*, with the task of selecting claims which are likely to have evidence of the relevant type [3, p. 443].

For the CDED task, the researchers considered 3 types of evidence, which were those types labeled in the data, namely *Study* evidence, *Expert* evidence, and *Anecdotal* evidence. Each of these types were expected to have different statistical signatures. For example, Study evidence is likely to include numerical values that are the reported results of the study, while Expert evidence will generally include the name of an expert and may include a title, role, or name of an organization. Because of the expected differences, each type of evidence was considered separately (i.e., separately classified by a separate instance of the pipeline) [3, p. 444].

Scores from the first two components (context-free components) are combined and the best scoring candidates are chosen from among a non-overlapping selection. Features used to score coherence are such as: unresolved anaphora and incomplete quotes [3, p. 444]. Context-free features used in the evidence characteristics component were manually and automatically compiled in-house lexicons of words characterizing the evidence type, named entities from the Stanford Named Entity Recognizer (NER) and finer grained named entities from an in-house named entity recognizer, patterns from simple and complex regular expressions, and the output of a subjectivity classifier. The context-dependent component relied on semantic relatedness.

They evaluated their pipeline against two baselines. The first treats the task as a purely semantic relatedness task using Word2Vec. The second treats the task as a purely information retrieval task using BM25 [38] [3, p. 446].

By using their pipeline they were able to achieve significant performance improvements over those baselines, and they also demonstrated the necessity of each element of their pipeline toward achieving their result. Their results were expressed in terms of the rank of the top scoring candidate which was indeed a match. So, for example, on average across claims for Study evidence a match was found at the 4th highest ranked candidate, and a match for Expert evidence was found at the 3rd ranked. (For comparison, by a less stringent standard, not penalizing certain misses, Word2Vec could classify an about 10th ranked true match on average across claims for Study evidence.)

In an independent study, using the first of the two IBM datasets (the one used by IBM researchers for the context dependent claim detection work), Lippi and Torroni [35] tested their belief that the most powerful machine learning techniques appropriate to the argument component detection task were not yet being exploited (see [39, pp. 172–173]). They used a support vector machine (SVM) classifier with a partial tree kernel to learn a model for classifying sentences as claims or not claims using portions of the IBM dataset and training and test data. They used a single 'context-independent'[12] feature, the constituency parse tree obtained from the Stanford CoreNLP parser. For their method they used the same evaluation as was used by the IBM researchers for the sentence component of their CDCD pipeline. That is, they used the proportion of the true positives in the top 200 sentences as ranked according to the scores from the classifier, as the measure of precision. Although their work was done on a later version of the IBM dataset that contained one additional topic, so that a proper direct comparison cannot be made in a principled way, they appear to have achieved similar results to the IBM researchers, 9.8% precision and F1=16.8 compared with 9.0% precision and F1=16.0 for the earlier IBM experiment.

Promising directions

A very promising direction seems to be the use of tree kernels with machine learning classifiers such as support vector machines as used by Marco Lippi and Paolo Torroni in the work mentioned above. They tackled the task of claim detection using an IBM dataset similar to the one used in this study (compiled and labeled in the same way, but produced earlier than the one used here) [35]. They used a support vector machine classifier and a single context independent feature, the constituency parse tree of the sentence. They obtained their parses tree using Stanford CoreNLP but modified them by replacing the word forms (at the tree's terminal nodes) with their stemmed versions for greater generality. Although they could not make a proper direct comparison with the results reported by IBM, since the data they received was slightly different (a later version of the dataset) than the data used by IBM in their system, their result was

[12] The same as what the IBM researchers denote as *context-free*.

comparable and possibly even a little better than the IBM system result which used very many highly engineered features and sophisticated in-house tools.

Their approach leverages deep syntactic comparisons made possible by using a tree kernel function to partition the space of training examples with a support vector machine. The kernel "trick" allows for the evaluation of a potentially exponential number of partial tree combinations efficiently.

Fig. 6 shows common elements between two parse trees which could be compared using the kernel method used by Lippi and Torroni in [35]. Note the matches at the words 'that' and 'is', for example. Other researchers have suggested that the verbs in the present tense might often be an indication of claims (while verbs in the past tense might often indicate the presence of evidence) (see [29, p. 50], [40]), and IBM researchers used the presence of the word 'that', when labeled in a specific way by their English Slot Grammar parser, as a feature indicating the presence of a claim [2, p. 1493]. The figure suggests the way in which the kernel method could automatically learn other useful features embedded within parse trees.

Lippi and Torroni used the partial tree (PT) kernel for their experiments [35, p. 187], which is due to Moschitti [41]. This is the most general form of tree kernel, which considers tree fragments consisting of single nodes and also nodes together with only some of their immediate child nodes. By contrast, the subset tree (SST) kernel, due to Collins and Duffy [42], considers only fragments that contain at least a node and all of its immediate child nodes (i.e., they may be all interior nodes with no leaves); and the subtree or syntactic tree (ST) kernel considers nodes together with their children down to the leaves.

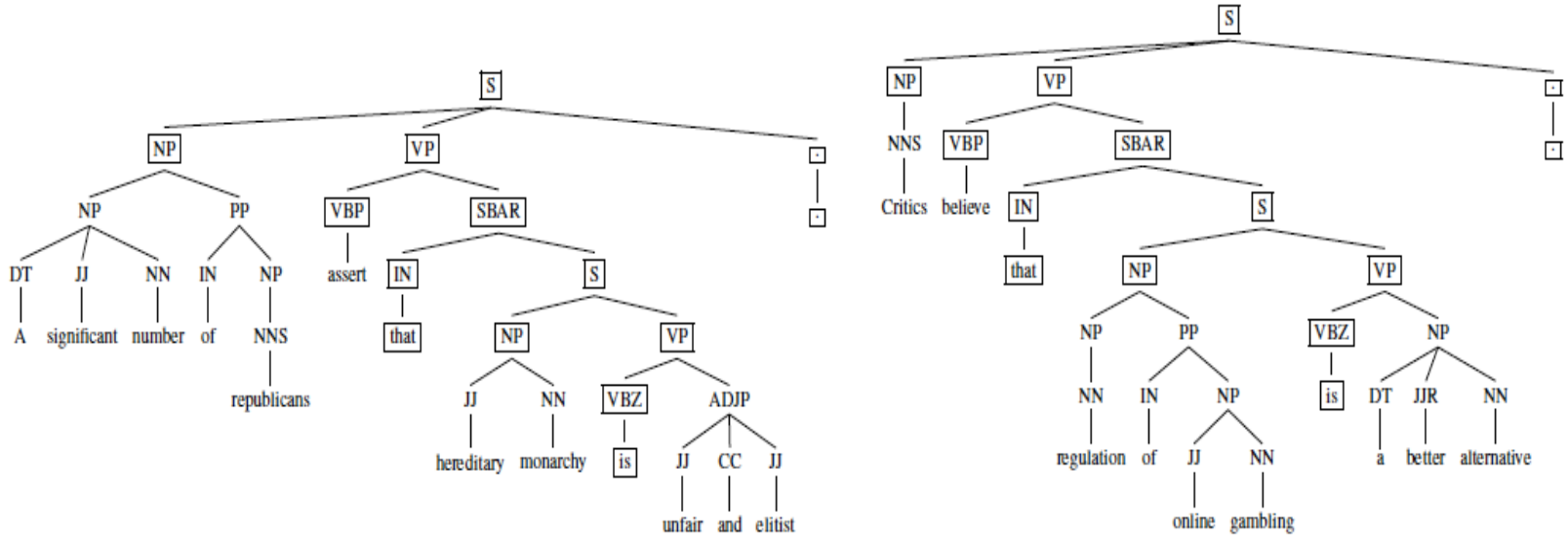

Fig. 6. Common elements between two constituency parse trees, as an example of the effect of the tree kernel method used by Lippi and Torroni, taken from [35, Fig. 1]

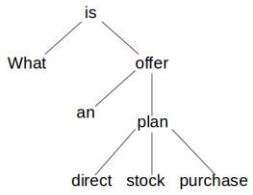

Fig. 7. Dependency parse tree, example from [41, Fig. 4]

Although the PT kernel is more general, Moschitti's empirical results show that, "[t]he new PT kernel is slightly less accurate than the SST one on constituent trees but much more accurate on dependency structures [41, p. 319]." Consider Fig. 7, Moschitti writes, "[i]t is clear that the SST and ST kernels cannot fully exploit the representational power of a dependency tree sinces from subtrees like [plan [direct stock purchase]], they cannot generate substructures like [plan [stock purchase]] or [plan [direct purchase]] [41, p. 325]."

In another application, Filice, Da San Martino, and Moschitti [43], report on the use of their tree kernels for learning relations between pairs of texts. They created kernels which incorporated semantic information from trained word vectors, from WordNet, and elsewhere, and they applied them to the Microsoft Research Paraphrase task and recognizing textual entailment task using the RTE3 dataset. For RTE3, their results were close to those of the best performing solution, and their classifier performed well on the paraphrase task as well. Such tasks resemble the relation learning task of predicting relations between claims and their supporting evidence, and so appear to be very interesting as possible approaches to apply to this domain.

CHAPTER 3

EXPERIMENTS

3.1 Model of Argumentation

The model of argumentation adopted for this thesis is a simple claim-premise model. This choice was driven primarily by the available datasets. Finding argumentation components for a more elaborate model might have also required many more computational elements or more specially targeted features. Using a claim-premise model can allow for a narrower focus on a smaller number of elements which can be considered in correspondingly greater depth.

Text fragments containing evidence that is suitable to support a claim are considered as premises in the model; and the argumentative support relation is, therefore, integral to the definition. A piece of evidence is only candidate evidence until the relation linking it to a specific claim is recognized.

The part of the model considered, therefore, consists of claims, evidence and support relations. No additional information to support inference is considered in this study. That is, the information identified in the text for each argument may not be enough for a complete or justified argument. Argument components are being identified.

3.2 Tasks

The argument mining tasks considered and evaluated for this thesis are two argument component detection tasks, claim detection and evidence detection; and a relation prediction task, finding the support relations that hold between claims and pieces of evidence that could be used to the support each specific claim.

There is sufficient labeled data so that we can use supervised machine learning for all three tasks, selecting a feature set based on considering the data and considering the similar data and approaches to similar tasks on such data found in the literature.

3.3 Experimental design

3.3.1 Procedure

The experiments use a corpus created by IBM for the debate technology project [3]. The corpus contains 547 documents. First, 58 topics were selected at random from among debate topics at Debatabase (http://idebate.org/debatabase); then Wikipedia articles relevant to those topics were selected and manually annotated. Claims found in each article which were relevant to the topic were labeled as such. Then evidence which could support each claim was found (limited to evidence in the same article as that containing the claim) and labeled [3].

Claims annotated in this corpus are, therefore, what are called 'context dependent claims' in that they are only those claims relevant to the topic. Any other claims in the article were ignored by annotators and not labeled. Likewise, evidence is 'context dependent evidence', which matches a specific claim or matches specific claims.

The model for these experiments is as follows. In the first step all sentences in the corpus are tokenized, split, part-of-speech (POS) tagged, and parsed; and the dependency relations and coreferences are found using natural language processing tools. In the next step features, as described below, indicative of claims or evidence or a match between claims and evidence are extracted using a variety of computer programs.

These features are used to train classifiers using supervised learning. A maximum entropy model[13] is trained to classify claims and separate models are trained to classify evidence. Evidence can be classified as one of three types, each of which have separate characteristics, study evidence, expert evidence, and anecdotal evidence.

Separate support vector machine models are trained using tree kernels instead of n-gram, parse tree fragment, and word-pair features, to automatically generate a large number of syntactic features based on constituency parse trees of each sentence, as described below.

A separate support vector machine model is trained for matching claims and evidence. All pairs of claim sentences and only evidence sentences containing complete evidence for a given article are used, with those corresponding to matches labeled as +1 and those not corresponding to matches labeled as -1; and they were tested using cross-validation.

For testing the explicitly designed features, maximum entropy model classifiers were used. For the experiments with tree kernels the subset tree (SST) kernel was used rather than the more general (but also more computationally expensive) partial tree (PT) kernel used by Lippi and Torroni [35]. Also, rather than using the traditional support vector machine (SVM), a fast approximation to the SVM model computed by the cutting plane algorithm was used, embodied in the software package, uSVM, by Aliaksei Severyn and derived from SVM Light with tree kernels by Alessandro Moschitti [44], [45], [46]. For CDCD task classifiers were created that correspond to the *sentence component* of the pipeline system of Levy, et. al. [2, p. 1493] and to the PT kernel classifier of Lippi and Torroni [35, p. 187]. For this experiment finding the exact boundaries of the claims was not attempted. For the CDED task, each of three types of evidence, study, expert, and anecdotal evidence were separately classified; each of these types are expected to have different characteristics, as Rinott et. al. [3, p. 444]. As was done by Rinott et. al., those

[13] This was implemented using the "Maximum Entropy Modeling Toolkit for Python and C++", by Zhang Le (see http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit.html).

topics from among the 39 topics in the test set of the IBM corpus that contained at least 3 positive examples were used [3, p. 443]; this resulted in 30 topics for expert evidence, 25 for study evidence, and 14 for anecdotal evidence. For these experiments, the CDED classifiers correspond to the *context-free, evidence characteristics component* and to the *context-dependent component,* and to "basic claim selection" of Rinott et. al. [3, pp. 445–446]. Only classification of evidence of one sentence in length was attempted, rather than the up to 3 sentence text spans classified by Rinott et. al. [3, p. 444].

Cross-validation as described in [2, p. 1495] was used, testing each topic separately using models trained on all of the remaining topics, and we report on the average of those results. From this corpus, training and test sets consisting of 39 topics of about 60,000 total sentences were used, with about 2,200 sentences containing claims and about 2,600 pieces of evidence spanning varying numbers of sentences, the same sets used by Rinott et. al [3].

3.3.2 Features for classification

Used in the experiments are some features described by Levy et. al. [2] and Rinott et. al. [3]: similarity scores, subjectivity scores, sentiment scores, named entity types, the presence of "that" at the beginning of a subordinate clause, presence of quotes, presence of a reference.

Used in the experiments are some features described by Stab and Gurevych [29]: position of the sentence in the paragraph and in the document, 1-3 n-grams, parse tree fragments corresponding to production rules, the depth of the parse tree, the tense of the verb, the presence of a modal verb, discourse markers selected from the Penn Discourse Treebank Manual 2.0; for classification of pairs, also used are: word pairs, relative positions in document and in the paragraphs.

TABLE II

FEATURES, BY CATEGORY, FOR MAXENT MODEL CLAIM AND EVIDENCE CLASSIFICATION

Lexical	n-grams	Presence of 'to be'	Presence of modal verb	Presence of (any) verb	Contains quote
Syntactic	Parse tree fragments	Presence of ccomp	ccomp 'that'		
Semantic	Similarity scores	Sentiment score	Subjectivity score		
Positional	Paragraph position in document	Sentence position in document	Position of sentence in paragraph		
Other	Presence of '[REF]'	Presence of named entities			

Features for detecting claims and evidence:

Lexical features

n-grams: n-grams of length 1 to 3 as a bag-of-words vector; unigrams are without stop words or punctuation, except for single word discourse markers, which are explicitly included.[14] Only those n-grams appearing in at least 10 sentences in the entire corpus are included.

Presence of any form of the verb, "to be": Claims are often declarative sentences, expressing a state of affairs often with a form of the verb 'to be'.[15] In the IBM corpus as a whole, about 63% of claim sentences contain the verb 'to be' versus 50% for all sentences. Features are derived from whole sentences which are to be classified as containing or not containing claims, and each sentence may have one or more clauses, each with its own verb.

[14] N-grams, discourse markers, and punctuation were all separately used in the experiments of Stab and Gurevych [29].

[15] In [35], the verb 'is' is given emphasis in the example of their Figure 1.

Presence of a modal verb: These are verbs such as 'can', 'could', 'should'. Stab and Gurevych [29, p. 50] report that they are often used to signal the degree of certainty when expressing a claim.

Boolean feature indicating that the verb is in the present tense and a boolean feature indicating that the verb is in the past tense: claims are usually in the present tense and describe a current or enduring state of affairs, i.e., how things *are*. Whereas, evidence is sometimes reported speech or text describing facts previously established (see [29, p. 50], [40]). Longer sentences sometimes contain both past tense and present tense verbs.

A Boolean feature that is true when a sentence does not contain a verb, and false otherwise: A sentence not containing a token tagged by the parser as a verb is generally not a claim since it is most likely not a complete sentence. In the experiment done by Lippi and Torroni [35] such sentences were removed from consideration although a small number contained claims due to mis-tagging by the parser. In the IBM corpus uses for these experiments only 4 sentences were labeled as containing claims, but were not tagged with a verb. These appear to be part-of-speech tagging mistakes. An example is, "Safeguards the constitutional rights of the individual", which was a section heading, part of a list of claims, each one a heading, with brief elaboration below each of them. "Safeguards" was tagged as a noun by the POS tagger.

Discourse markers are included as part of n-gram features. Discourse markers are words or phrases such as "because", "consequently", "thus", which indicate a transition in a discourse. The list of discourse markers used as features for this work are taken from the Penn Discourse Treebank 2.0 Annotation Manual [47], but some markers which do not indicate argumentative discourse, such as markers indicating temporal discourse, have been removed. The edited list was obtained from the supplementary materials to the paper, "Identifying Argumentative Discourse Structures in Persuasive Essays" [29]. All of the multiword markers are already included as bigrams or trigrams. Unigram markers, which might have been excluded because they are stop words, were explicitly added. A limitation of these markers is that they often indicate transition from stating a conclusion to stating a premise within the same sentence although the marker can also sometimes be found at the beginning of a new sentence. In Stab

and Gurevych's experiment [29], features were extracted for each clause of a sentence and clauses were considered as to whether they contained claims or premises. If a marker occurs within a sentence and indicates the presence of both claims and evidence in the same sentence, it is, however, still of use because claims and evidence are separately classified. A sentence can be classified as containing both claims and evidence.

Syntactic features

Parse tree fragments: For the Maxent classifiers, inferred production rules are used as features. In this approach, each subtree root (down to pre-terminal nodes) and its children one level deep are extracted from the constituency parse tree as a feature. For example, 'S -> N VP', could be one binary feature, and expanding the VP branch we could have 'VP -> V NP PP' as a second binary feature. Features extracted from parse trees in this way were used by Stab and Gurevych [29, p. 50]. Lemmas are used rather than word forms for better generality. Only those fragments meeting the frequency requirement, that they are features of at least 10 sentences in the corpus, are included. The depth of the parse tree is also given as a feature.

Sentence contains the "ccomp" dependency relation: A sentence may have dependent clauses marked by the Stanford dependency relationship, "clausal complement" (ccomp) [48, p. 4]. An example drawn from the IBM corpus is, "The heading of the advertisement asks, ``Have you seen this girl?''" The relation is ccomp(asks, seen), where the verb of the main clause is 'asks' and the verb of the dependent clause is 'seen'. This structure sometimes indicates the presence of evidence, especially when the word following the verb of the main clause is "that", as seen in this example: "Terry Flew writes that generally representations of gender in digital games are stereotyped." (Also notice that the presence of the named entity, "Terry Flew" may help to indicate that this is expert evidence.) Levy et. al. encode a feature indicating whether the sentence contains the token "that" which fills a particular role identified by their English Slot Grammar parser [2, p. 1493]. Similarly, Lippi and Torroni [35, p. 186] discuss an example of a claim contained in a subordinate clause of a sentence and beginning with the word "that": "A significant number of republicans assert **that** *hereditary monarchy is unfair and elitist*" [my

emphasis; claim is delineated by italics and was originally found in the IBM corpus]. The same structural relationship is encoded by the use of the "ccomp" dependency relation identified by the Stanford parser [49]. Claims are often introduced by the word "that", but also by other words or by a colon (':') [35, pp. 186–187]. A binary feature is included that indicates the presence of the ccomp relation and a separate binary feature indicates the presence of a ccomp relation with the first verb followed by the word, "that".

Sentence contains the "ccomp" dependency relation, where the dependent starts with "that" (other common words in this position are "with", or punctuation); examples:

"The heading of the advertisement asks**,** 'have you seen this girl'."

"He was quick to mention **that** the case would probably do well with or without his presence."[16]

Contains a quote: The presence of a quote might sometimes indicate the presence of evidence, for example, the quoting of a study or expert or of reported speech of an anecdote. This was used as a feature for detecting evidence by Rinott et. al. [3, p. 445].

Semantic features

Similarity scores: (between the sentence and the topic): pairwise averages and maximums of: WordNet similarity score[17]; Word2Vec similarity score[18] [3, p. 445], which is the cosine similarity of the word vectors for each pair of words. In a separate version, in order to improve the results, multi-word terms found in WordNet[19] were considered as single words.[20] For example, in the topic, "make physical education compulsory", 'physical education' is a WordNet

---

[16] These examples are taken from the IBM corpus.

[17] WordNet similarity scores were obtained using the WNSim package from the University of Illinois [50], [51], [52].

[18] The Word2Vec similarity scores were obtained using the Gensim python package [53].

[19] The Freeling parser library was used for matching multi-word WordNet concepts [54].

[20] Levy, et. al. [55] propose a multi-word term relatedness task and provide a classifier-based method as a benchmark.

concept. The words physical alone or education alone could be matched in sentences discussing very different themes other than physical education. This seems to account for some failures in classification when similarity scores are computed from all individual words, as discussed below. Multi-word terms were grouped together for computing both WordNet and Word2Vec similarity scores. The version using this method is compared, in the results, to the version not considering multi-word terms.

In addition, pronominal anaphora were first resolved before computing the similarity scores. This improved the results as shown in the results tables.

Sentiment score: A real valued measure of the sentiment polarity expressed by the sentence, based on a list of sentiment words, which takes negating words such as 'not' into consideration.[21] A sentiment score was also used by Levy et. al. [2, p. 1493].

Subjectivity score: A real valued measure of the subjectivity of the sentence, based on rules and a list of words. The presence of first person subjects and words usually indicating opinions increase this score. A classifier-based subjectivity score was used by Levy et. al. [2, p. 1493] and by Rinott et. al. [3, p. 445].

Positional/Structural features

This kind of feature was used in and experiment described in the paper on persuasive essays by Stab and Gureyvch [29, pp. 49–50]. In a persuasive essay, claims are often introduced at the beginning of a document or the beginning of a paragraph. This is not necessarily a characteristic of documents in the domain of Wikipedia articles, which are more expository; however, it can be seen that claims frequently begin paragraphs. In this corpus, more than a third of all labeled claims start a paragraph. One feature, position of the sentence within its paragraph, was given as the ratio, number of the sentence within the paragraph over total number of sentences in the paragraph. This produces characteristic numbers for claims or evidence which can be distinguished by being at or near the last sentences of a paragraph, whereas the value of this

[21] The TextBlob python package was used for sentiment scores and also for subjectivity scores.

feature for first sentences will vary depending on the length of the paragraph. The absolute position within the paragraph is, therefore, used as well.

Position in the document measured in paragraphs: The paragraph number, the number of paragraphs from the beginning of the document to the paragraph in which the sentence is found is encoded, and the ratio of paragraph number to total paragraphs in the document is encoded. In the corpus of Stab and Gurevych [29, p. 50], some claims are always found in the beginning or ending paragraphs of a document.

Position in the document measured in sentences: As above, the sentence number is encoded, and the ratio to total number of sentences in the document is encoded.

Position of the candidate claim's sentence in its paragraph: The sentence number, counting from the start of the paragraph it is found in is encoded, and the ratio of the sentence number to the total number of sentences in the paragraph is encoded. Stab and Gurevych [29, p. 50] found that paragraphs frequently begin with a claim. In the IBM corpus, about one third of claims are found in the first sentence of a paragraph.

Other features

Has a reference: (In this corpus, sentences with references contain or end with the text, "[REF]"). This is often an indication of the presence of study evidence. Claims and evidence sometimes appear in the same sentences; so, in that case, the presence of the reference might signal the presence of evidence, but not the absence of claims.

A Boolean feature for each named entity type which can be recognized by the Stanford Named Entity Recognizer: person, location, organization, misc, money, number, percent, date, time, duration, and set. The presence of a person or organization can sometimes indicate the presence of expert evidence and numbers may sometimes indicate the presence of study evidence, which reports statistics from a study [3, p. 445].

In total 38 individual binary or numeric features plus unigrams, bigrams, trigrams, and parse tree fragments were used to classify claims and evidence with the Maxent model classifier.

TABLE III

FEATURES, BY CATEGORY, FOR THE TREE KERNEL MODELS FOR CLAIM AND EVIDENCE CLASSIFICATION

(a) Features, by category, for SSTK model claim and evidence classification

Syntactic	Constituency parse tree

(b) Features, by category, for SSTK+ model claim and evidence classification

Lexical	Presence of 'to be'	Presence of modal verb	Presence of (any) verb	Contains quote	
Syntactic	Constituency parse tree	Presence of ccomp	ccomp 'that'		
Semantic	Similarity scores	Sentiment score	Subjectivity score		
Positional	Paragraph position in document	Sentence position in document	Position of sentence in paragraph		
Other	Presence of '[REF]'	Presence of named entities			

Features for classifying claims and evidence with the SST kernel:

All features as above except that the n-grams, and parse tree fragments are replaced by the constituency parse tree with lemmas as the terminal nodes rather than word forms.

The use of tree kernels with support vector machines was proposed and tested by Lippi and Torroni [35], [39] as a way to automatically generate a large number of syntactic features for classification, which can be either all sub-trees of a (parse) tree given as a kernel feature, or all subset trees of such a tree. The subset tree kernel is to be used as a feature for classification of claims and evidence in these experiments. The tree to be used is the constituency parse tree with words (leaf nodes) replaced by their lemmas for better generalization. This is the same procedure used by Lippi and Torroni [35], except that they used partial tree kernels rather than subset tree kernels, and stemmed rather than lemmatized words (i.e., tokens at leaf nodes). A partial tree kernel is a more general structure than a subset tree kernel in which tree fragments are not constrained to contain all child nodes at any given level [41, pp. 319–320]. Although

subset tree kernels capture a great amount of detail there exists the potential problem of overfitting to irrelevant features.

The motivation for incorporating syntactic features is the observation that some sentences (or clauses) 'look' like claims due to their syntactic structure. Lippi and Torroni [35, p. 186] give the following examples. They write that, "'The prototypical delegative democracy has been summarized by Bryan Ford in his paper, Delegative Democracy' 'sounds like' a factual statement, whereas 'The difficulty and cost of becoming a delegate is small' 'sounds like' a claim, independently of the topic under discussion." What seems obvious in the example is the presence of the present tense form of the verb 'to be' in the second example sentence, and the past tense in the first example sentence. Both of those indicators are given as separate features; however, less obvious distinguishing characteristics embedded in the syntactic structure may be discovered by including syntactic features such as those described here.

TABLE IV

FEATURES, BY CATEGORY, FOR MAXENT MODEL PAIR CLASSIFICATION

Lexical	n-grams			
Syntactic	Parse tree fragments			
Discourse	Word pairs	Presence of shared coreferences	Similarity between sentences	
Positional	Section distance between sentences of pair	Paragraph distance between sentences of pair	Sentence distance between sentences of pair	Claim precedes candidate evidence

Features for classifying claim and evidence pairs:

Lexical features

n-grams: 1-3 grams including discourse markers, as described above.

Syntactic features

Parse tree fragments: as described above, for claim and evidence classification.

Discourse features

Word pairs: pairs of words, one each selected from the claim sentence and one each selected from the candidate evidence sentence, where each word is one of either a noun, verb, or discourse marker, which are those categories described by Marcu and Echihabi [34], in their research on using word pairs to detect implicit discourse relations, as 'most representative words'. These are included when they are features of at least 10 sentences in the entire corpus. Marcu and Echinabi [34, p. 373] found that if one uses only most representative word one can achieve very good performance while training with only about 100,000 training examples. Stab and Gurevych benefited with only 6,330 pairs [29, p. 52]. About 30,000 training examples (claim and evidence sentence pairings) are available in the IBM dataset (although some of those are in the heldout set), so it is expected to be an interesting test.

The presence of shared coreferences: A Boolean feature is included to indicate the presence of any coreferences shared between the pair of sentences. This should be a strong indication of a relationship between the two sentences. If the coreferences have the same or complementary roles in the sentence it could also be a strong indication that the sentences are *about* the same thing and be a good way to establish shared context between them.

Similarity scores (between the two sentences): Same as above, but coreferences shared between the two sentences, as identified by the Stanford coreference annotator, were extracted and considered as exact matches.

Positional/Structural features

Distance between the two sentences as measured in sections: Wikipedia articles are sometimes divided into sections, which are characterized by having section-heading text. Regular expression matching was used to find likely section headers and assigned each section a sequential number. It seems likely that when there are separate sections, many arguments would be concluded within a single section or within closely related adjacent sections. This distance is the absolute value of the difference between the two sentences' section numbers.

Distance between the two sentences as measured in paragraphs: Paragraphs are one way that a natural language discourse is organized. Changes in topic may occur at paragraph boundaries.

In some cases an author may fully exhaust a line of argument, including all of the evidence intended to support a particular claim, before moving on to the next line of argument[22]. The move to the next argument may take place after some number of paragraphs and at a paragraph boundary.

Distance between the two sentences as measured in sentences: The absolute value of the difference between the sentence numbers of the two sentences.

Boolean feature indicating that the first sentence precedes the second: Claims frequently precede the evidence that supports them [19, p. 130].

TABLE V

FEATURES, BY CATEGORY, FOR THE TREE KERNEL MODELS FOR PAIR CLASSIFICATION

(a) Features, by category, for SSTK model pair classification

Syntactic	Constituency parse trees (one for each of the claim and the candidate evidence)

(b) Features, by category, for SSTK+ model pair classification

Syntactic	Constituency parse trees			
Discourse	Presence of shared coreferences	Similarity between sentences		
Positional	Section distance between sentences of pair	Paragraph distance between sentences of pair	Sentence distance between sentences of pair	Claim precedes candidate evidence

Features for classifying claim and evidence pairs with the SST kernel:

All features as above except that word pairs are replaced with the constituency parse trees.

Note about similarity scores

[22] This discourse phenomenon is discussed by Lawrence, et. al. [24], in the context of extracting arguments from natural language text.

Semantic similarity between the target sentence and the debate topic corresponding with the article is used to distinguish claims and (candidate) evidence from non-labeled sentences because claims and evidence are "context dependent". Manual annotators of the corpus who labeled sentences as context dependent claims or context dependent evidence were instructed only to label claims which were relevant to the topic, and evidence which was evidence which could be used to support a particular labeled claim. Therefore, it is not enough to identify sentences which could contain a claim independent of the context. The claim must also be relevant to the topic in order to benefit from the manual labels in this supervised learning task. For evidence, it must be able to support a particular claim, but in the first step in the pipeline for classifying evidence, sentences are only classified as candidate evidence where subsequently they will be classified as context dependent evidence if a matching claim can be found.

Semantic similarity between claim and topic is taken as an indication of the claim's relevance to the topic and is therefore considered useful as a feature. The IBM researchers who created the corpus also used a number of similarity based features to classify claims and evidence [2, p. 1493], [3]. Semantic similarity between evidence and topic is considered as a way to narrow the field of acceptable candidate evidence since evidence that supports a claim that is relevant to the topic is likely to be recognizably similar to the topic as well.

Two similarity measures are used for these experiments. WordNet similarity counts the distance between words in the WordNet taxonomies of words related by hypernym-hyponym relations and meronym relations. It also considers synonym and antonym relations (with antonym relations producing a negative distance value). The pairwise average of WordNet similarity values for words in the target sentence and the topic is computed and used as a feature. Only words tagged as adjectives, nouns, or verbs are compared, and stop words are excluded. Word2Vec similarity is the cosine similarity between vectors representing words where the vectors were previously learned as containing values indicating the frequency with which other words show up in the vicinity of the target word in a large corpus. A connection between the meanings of words that are frequently used in the context of the same other words is supposed.

The pairwise average of the Word2Vec similarity between the target sentence and the topic, exclusive of stop words, is computed and used as a feature.

A problem with this approach is that it can aggregate similarity scores for many irrelevant pairings of words. In fact, a similarity score (especially the WordNet similarity) computed using the words of a sentence compared with the words of the same sentence can be smaller than the score for two different sentences. This could happen if there are a large number of words in a sentence and only a small number are similar or a second sentence contains words such that each word is similar to more than one word in the first sentence. At least one example for WordNet similarity scores has been found in the data.

The maximum similarity of all pairs of words is, therefore, also used as two additional features (a score using WordNet and a score using Word2Vec).

Only words having the same part of speech can be compared using WordNet; therefore, some similar words, such as "violent" and "violence" which can be found in sentences will not be counted as similar as part of the WordNet similarity score feature. (They are also not counted as dissimilar; such pairings are not used.) Such similarities are captured, however, by the Word2Vec similarity.

Some alternative approaches are possible. An option could be to pair words according to their function in the sentence, which would be according to the dependency relations or semantic role relations in the sentence. A third option, also used with WordNet by Lawrence and Reed [19, p. 131], is to pair words according to maximum similarity and then take the average of the scores for all pairings.

A separate feature could be used to capture the similarity of the sentence subjects potentially indicating that the sentences are about similar things. An approach similar to this was used by Levy, et. al [2, p. 1493], in addition to a WordNet based similarity score feature. In these

experiments a feature indicating whether a sentence pair contains coreferences shared between them as found by the Stanford coreference annotator may work as an indicator just for such pairs, that they are about the same thing.

The similarity feature is further limited by the presence of unresolved anaphora. If the word 'it' in the target sentence could be substituted with its coreferent, for example, 'violent video games' then it could be matched with the topic, '…violent video games should be banned'. This was done for the experiments labeled as Maxent 2 and Maxent 3 in the results. Another concern in the above example, is the presence of compound nouns and nouns with modifiers. 'Video games' is really one term and appears frequently with and without the modifier 'violent' in claims and evidence for this topic. An approach to producing similarity scores taking compound nouns into consideration improves the results somewhat. This was tried in experiments labeled as Maxent3 and SSTK2 in the results. A method of capturing text similarity by pairing words with maximal scores and taking compounds into consideration was explored in detail by Bhagwani, Satapathy, and Karnick [56], using maximal weighted bipartite matching. Their method was reported to perform significantly better than some simpler methods, and the use of compounds seemed capable of making a contribution to its success (when incorporated in a certain way) [56, pp. 583–584] but was not attempted for these experiments.

3.4 Implementation

3.4.1 Setup

All articles in the dataset were first tokenized, segmented into sentences, tagged with part-of-speech tags, and parsed using the Stanford CoreNLP parser and were also annotated with lemmas, dependency relations, and coreference mentions using the same suite of tools. All sentences together with the information added by Stanford CoreNLP pipeline tools were then loaded into a relational database in order to make feature extraction easier. The database used for these experiments was MySQL.

Many sentences in the dataset contained the notation "[REF]" to indicate the presence of a reference. Since this text was irrelevant to the parsing, it was first removed and replaced with white space using regular expression matching. After the parsing, it was added back into the sentence during the sentence reconstruction step, so the reconstructed (i.e., original) sentence could be matched to the list of claim text fragments and/or to the list of evidence text fragments.

Tab delimited files relating claims, topics, articles, and evidence were provided with the articles as part of the dataset. These were also loaded into the database as separate tables. The tables containing parsed sentences, together with article numbers could then be joined with article, topic, and claims tables to extract the claim labels; joined with article, topic, and evidence tables to extract the evidence labels; or joined with article, topic, claim, and evidence tables to extract the label indicating a match between claim and evidence (i.e., an argumentative support relation).

Other features were extracted using SQL queries or other programmatic methods discussed, in greater detail, in the appendix.

Most of these features were also stored in the database. Therefore, in addition to the information gained by the use of the machine learning classifiers, ad hoc queries could be performed to get corpus statistics such as, for example, how many sentences containing text fragments labeled as claims are also first sentences of a paragraph, or how many sentences containing text fragments labeled as evidence were also annotated with the ccomp dependency relation. These statistics together with inspection of the texts and intuition could help to guide the feature engineering process.

3.5 Results

Results are presented in tables. TABLE VI shows the results on the CDCD (claim detection) task. Rows are labeled: Maxent(1-3), for the maximum entropy model classifiers; SSTK+ for the tree kernel classifiers augmented with a feature vector as described above; SSTK(1-2) for the

classifiers using only the tree kernels; n-grams for classification using only n-grams; and Random, for random scoring. Maxent1 shows the results without resolving pronominal anaphora before computing similarity scores. Maxent2 shows results using pronominal anaphora resolution. Maxent3 shows the results using pronominal anaphora resolution and multi-word WordNet concepts in the input to the similarity score computation. SSTK1 shows the results using the trees as described above. For SSTK2 multi-word terms at leaf nodes were combined if their nodes were siblings. For this task, in addition to multi-word concepts from WordNet, multi-word named entities detected by the Stanford named entity recognizer were also combined. All other classifiers (i.e. those with results shown in the following tables, for evidence and for sentence pairs) correspond to the feature sets of Maxent2 (for feature vectors) and SSTK1 for trees. Rows at the bottom show previous results achieved by Levy et. al., the results from the partial tree kernel model of Lippi and Torroni (labeled PTK), and the results from Lippi and Torroni's TK+Topic model, which combines contributions from a partial tree kernel and a vector containing one feature, the cosine similarity between the candidate claim sentence and the topic sentence, via their bag-of-words vectors [35, p. 189]. TABLE VIII, TABLE IX, and TABLE X show the results for detecting evidence (corresponding to the *context-free, evidence characteristics component* of the pipeline of Rinott, et. al. [3, pp. 444–445]). TABLE XI, TABLE XII, and TABLE XIII show the results for matching claims with candidate evidence, in order to distinguish context-dependent evidence.

Scoring for the results is according to F1, precision, and recall scores when taking the top ranked 200 sentences to be positive predictions, according to a common scoring scheme in information retrieval, and the same method used by Levy et. al. [2] and by Lippi and Torroni [35] for the claim detection task. Since results from one step may be pipelined into a next step, high recall is especially of interest. Small values for precision may be misleading since topics do not have a full 200 claims on average; and the same is true for evidence and for claim and evidence pairs. For claims, the topic containing the maximum number of positive examples contains 113 claims. The average across all topics used was 42.1 positive examples. So we should not expect a precision value higher than 42.1/200, or 0.211. Since the scores are computed on a topic by

topic basis and then aggregated, the number of top ranked sentences to consider should be at least as many as the maximum number of positive examples in a topic, 113, but 200 was used because it corresponds to procedure used in the very similar experiments by Levy et. al. [2] and Lippi and Torroni [35].

The right side of TABLE VI shows the result when selecting the top 50 sentences. Precision improves, and, of course, recall is reduced. The maximal average precision as shown in a table below the result tables is 0.842 for the case of selecting the top 50 sentences; so although the precision results are better for selecting 50, they are not close to the maximal value.

TABLE VI

RESULTS FOR CLASSIFYING SENTENCES AS CONTAINING OR NOT CONTAINING CLAIMS

	F1@200	P@200	R@200
SSTK+	0.183	0.107	0.614
SSTK1	0.163	0.095	0.561
SSTK2	0.165	0.096	0.568
n-grams	0.048	0.027	0.198
Random	0.046	0.026	0.198
Maxent1	0.150	0.087	0.527
Maxent2	0.160	0.094	0.539
Maxent3	0.163	0.096	0.552

claims detected on earlier versions of the dataset[23]:

PTK(L&T)	0.168	0.098	0.587
TK+Topic	0.180	0.105	0.629
Levy et al	0.160	0.09-[24]	0.73-

	F1@50	P@50	R@50
SSTK+	0.232	0.193	0.290
SSTK1	0.191	0.157	0.243
SSTK2	0.198	0.164	0.248
n-grams	0.045	0.041	0.051
Random	0.041	0.303	0.062
Maxent1	0.192	0.155	0.252
Maxent2	0.208	0.174	0.259
Maxent3	0.201	0.166	0.253

claims detected on earlier versions of the dataset[25]:

Levy et al	0.248	0.18-	0.40-

Positive examples per topic and maximal average precision

avg.	42.10
max.	113
min.	6
max P@200	0.211
max P@50	0.842

[23]The earlier versions of the dataset contained fewer topics and fewer than half as many claim containing sentences to use as positive examples in the classifications; therefore, the results are not directly comparable.

[24] The thousandth place is shown replaced with a hyphen because it was originally reported to the hundredth place.

[25]Results for Lippi and Torroni's [35] partial tree kernel models are not included in this table because they only reported experimenting with selecting the top 200 claims, not also with selecting the top 50.

TABLE VII

RESULTS FOR SELECTING SMALLER NUMBERS OF RELEVANT CLAIMS (BEST MATCHES)

	P@50	P@20	P@10	P@5
SSTK+	0.193	0.247	0.277	0.323
Levy et al RC[26]	0.12-	0.16-	0.20-	0.23-
Random	0.030	0.029	0.030	0.021

In TABLE VII, the results for selecting smaller numbers of claims are shown. In a real application, we might want to present the user with the top 5 matches, with the hope that they would all or nearly all be relevant. An example is in TABLE I. As can be seen from the table, in the experiment of Levy, et. al., they were able to return one correct result in five on average. Using the classifier that performed best for selecting 200 and 50 claims, SSTK+, one in five on average can also be achieved, but it is getting closer to two. Note, again, that the dataset used for these experiments was much larger than the one used by Levy, et. al.; so, that most likely accounts for some of the improvement gain. One way to give an interesting target for future research could be in terms of numbers of correct results on average when selecting five. Thus, a full 2 claims or more could be the target result for a next study with improved features and methods.

The average number of claims from among the top ranked 200 sentences that were correctly classified by the SSTK was 18.8, while the average for Maxent was 17.9. The tree kernel was

[26] Ranking Component from Levy et. al. [2, p. 1495] (see Table 5 in Levy et. al. [2, p. 1497]): this is the result of taking the 200 best sentences from the previous step and passing them through the Boundaries Component, a next step in the pipeline, which attempts to find the exact (sub-sentence) boundaries of the claims, filtering out likely non-argumentative surrounding material; then reclassifying using scores from previous pipeline steps and additional features; and finally selecting the best ranked 50 sentences. Again these results are from work on an earlier version of the dataset than used for the other models in this table. The other models correspond the first step of the pipeline of Levy et. al. [2], the *sentence component*.

able to correctly classify one more claim on average. It is impressive considering that very little feature design is necessary for that model, but it is also encouraging with regard to the usefulness of the explicitly designed features which were able to do nearly as well. The average difference in number of claims correctly classified (from among the 200) between SSTK+ and Maxent is almost 3 claims.

All classifiers significantly outperformed random, where random means selecting a random set of 200 sentences to consider as positive predictions; and all other classifiers also outperformed the classifier using only the n-gram bag-of-words vector, consisting of n-grams of length 1-3 and discourse markers as described in the feature description above. Maxent model and SSTK model classifiers performed comparably to the models used by Lippi and Torroni and by Levi et. al., although they are not directly comparable due to this experiment having the advantage of much additional data. The similar performance can be said to suggest that the features tested here are similarly promising.

The SSTK+ model was clearly superior, especially given the small numbers of claims in the training and test data. Only about 3.5% of sentences contain claims. This shows that additional information is present in the constituency parse trees which when better understood should help to inform good feature design.

Examining some high ranked false positives from the model before grouping multi-word terms together, we see that they appear to indeed look like claims. Some examples are shown below. The examples were selected from the topic, "This house would make physical education compulsory". A true positive result appears as the top ranked example:
"Data suggests that students who lack opportunities for play do not grow into happy, well adjusted adults, [REF] and although schools are now focusing their attention on test scores while eliminating recess/physical education, studies show that recess
and/or P.E. actually increase test scores as the students produce dopamine, a neurotransmitter involved in memory and problem solving [REF]."

Two illustrative false positive examples:
"Studies show that exercising in middle age leads to physical ability later in life [REF]."
"Obesity is a medical condition in which excess body fat has accumulated to the extent that it may have an adverse affect on health [REF]."

In these two examples, the sentences received high scores for similarity with the topic. We see the word 'physical' in the first and 'health' in the second which are related to the topic words, but which are unsuccessful as indicators that these sentences are *about* the same thing as the topic. This illustrates the importance of the similarity measure to these context-dependent tasks and also the subtlety of the task.

In addition, we often see that claims and evidence are found in the same sentences. A claim may be made in one clause and evidence provided in another, or the claim is presented as part of evidence introduced with words such as "studies show" or "experts say". Classifying clauses rather than sentences may, therefore, improve the results somewhat.

For classifying sentences as evidence and classifying sentence pairs as instances of claims together with topical supporting evidence, only candidate pieces of evidence of one sentence in length (or shorter) were used. (For comparison, in the study by Rinott, et. al. [3], all spans of one to three sentences in length were considered as potential candidate evidence; the maximum length of a piece of evidence in the dataset was 25 sentences [3, p. 444]). Considering spans of 1 - 3 sentences in length covers 90% of the evidence in the dataset [3, p. 444]. Considering only one sentence spans results in coverage of only about a third of the evidence, but simplifies the experiment.

TABLE VIII

RESULTS FOR CLASSIFYING SENTENCES AS CONTAINING EVIDENCE OR NOT CONTAINING EVIDENCE, WHEN CONSIDERING THE 200 TOP RANKED PAIRS AS POSITIVE PREDICTIONS

If the sentence is contained within a longer (multi-sentence) piece of evidence, it is considered a positive result in the overlap scoring.

STUDY evidence:				overlap		
	F1@200	P@200	R@200	F1@200	P@200	R@200
Maxent	0.056	0.030	0.474	0.122	0.069	0.523
SSTK	0.078	0.041	0.671	0.117	0.066	0.520
SSTK+	0.085	0.045	0.726	0.126	0.071	0.552
n-grams	0.007	0.004	0.075	0.028	0.015	0.129
Random	0.015	0.008	0.137	0.034	0.019	0.164

Positive examples per topic and maximal average precision

STUDY		overlap
avg.	13.84	30.16
max.	41	85
min.	3	5
P@200<	0.07	0.15

EXPERT evidence:				overlap		
	F1@200	P@200	R@200	F1@200	P@200	R@200
Maxent	0.081	0.043	0.638	0.140	0.082	0.495
SSTK	0.072	0.038	0.626	0.107	0.062	0.422
SSTK+	0.090	0.048	0.736	0.133	0.077	0.501
n-grams	0.005	0.003	0.063	0.025	0.015	0.100
Random	0.015	0.008	0.141	0.035	0.021	0.119

Positive examples per topic and maximal average precision

EXPERT		overlap
avg.	15.93	44.77
max.	64	180
min.	4	4
P@200<	0.08	0.22

ANECDOTAL evidence:				overlap		
	F1@200	P@200	R@200	F1@200	P@200	R@200
Maxent	0.037	0.019	0.456	0.088	0.049	0.457
SSTK	0.033	0.017	0.492	0.062	0.034	0.348
SSTK+	0.046	0.024	0.621	0.075	0.041	0.429
n-grams	—	0.000	0.000	0.013	0.008	0.045
Random	0.010	0.005	0.139	0.025	0.014	0.107

Positive examples per topic and maximal average precision

ANECDOTAL		overlap
avg.	7.79	22.71
max.	22	58
min.	3	5
P@200<	0.04	0.11

TABLE IX

RESULTS FOR CLASSIFYING SENTENCES AS CONTAINING EVIDENCE OR NOT CONTAINING EVIDENCE, WHEN CONSIDERING THE 50 TOP RANKED PAIRS AS POSITIVE PREDICTIONS

If the sentence is contained within a longer (multi-sentence) piece of evidence, it is considered a positive result in the overlap scoring.

STUDY evidence:				overlap		
	F1@50	P@50	R@50	F1@50	P@50	R@50
Maxent	0.104	0.063	0.286	0.168	0.123	0.264
SSTK	0.130	0.078	0.386	0.153	0.110	0.247
SSTK+	0.154	0.094	0.440	0.182	0.131	0.297
n-grams	0.013	0.008	0.034	0.026	0.021	0.035
Random	0.010	0.006	0.026	0.039	0.030	0.059

Maximal average precision

STUDY		overlap
P@50<	0.27	0.60

EXPERT evidence:				overlap		
	F1@50	P@50	R@50	F1@50	P@50	R@50
Maxent	0.145	0.090	0.369	0.159	0.125	0.218
SSTK	0.097	0.059	0.265	0.113	0.087	0.162
SSTK+	0.152	0.095	0.383	0.161	0.125	0.229
n-grams	0.006	0.004	0.011	0.014	0.015	0.014
Random	0.017	0.011	0.048	0.025	0.021	0.033

Maximal average precision

EXPERT		overlap
P@50<	0.32	0.90

ANECDOTAL evidence:				overlap		
	F1@50	P@50	R@50	F1@50	P@50	R@50
Maxent	0.047	0.029	0.130	0.082	0.061	0.122
SSTK	0.027	0.016	0.097	0.049	0.357	0.078
SSTK+	0.062	0.036	0.246	0.095	0.064	0.185
n-grams	—	0.000	0.000	0.007	0.006	0.010
Random	0.007	0.004	0.022	0.018	0.013	0.028

Maximal average precision

ANECDOTAL		overlap
P@50<	0.16	0.45

Therefore, in the case of scoring for evidence (TABLE VIII and TABLE IX) and for claim and candidate evidence pairs (TABLE XI and TABLE XII), some sentences might not fully contain a piece of evidence, but rather might be contained within a multi-sentence piece of evidence. A more liberal scoring showing the performance of the classifiers at detecting sentences overlapping with evidence is shown as the rightmost sets of columns. The good performance of these classifiers on the overlap measure suggests that they might also perform well at classifying the longer text spans, such as two or three sentences.

Because we selected only single sentences for the classification of evidence, we excluded a large number of evidence in the documents. The results are, nevertheless, informative. And it is also noteworthy that the tree kernel classifiers performed significantly better than the maximum entropy model classifiers on the evidence detection (and link prediction, shown in TABLE VIII and TABLE XI) tasks and by a much larger margin than on the claim detection task. All classifiers again significantly outperformed the random and the n-grams bag-of-words vector classifiers. These results could not be compared with Rinott, et. al. because results for this stage of their pipeline were not reported. Pretty good recall, especially, for the SSTK+ classifiers for STUDY and EXPERT evidence (0.73 and 0.74, respectively) suggests that these classifiers could be a useful part of a pipeline model, which could be tried in the future. Classification of ANECDOTAL evidence may have performed poorly due to limited data, which is mentioned by Rinott, et. al. [3, p. 447]. The average number of positive examples is less than half that of either of the other types (considering only topics for which the number is at least three) and the total number of positives across topics is only 161, compared with 953 for EXPERT evidence and 619 for STUDY evidence.

With regard to interpreting the results, it is important to note, again, that due to the small number of claim sentences, evidence sentences, or claim and evidence candidate sentence pairs relative to the total numbers of sentences or sentence pairs for each topic, values for precision averaged over topics have low expected maximums. For STUDY evidence, the average number of 1-sentence pieces of evidence per topic is 13.8 with a maximum of 41. Therefore, when

considering the top 200 results as though positive, a maximum of 41/200 precision is possible (for those single topics with the maximum number of positive examples), and 0.07 is the maximal average precision. For EXPERT evidence, the maximal average precision is 0.08, and for ANECDOTAL, it is 0.04. For overlap scoring, the maximal average precision values are, respectively, 0.15, 0.22, and 0.11.

Although outperformed by the tree kernels, the maximum entropy model classifiers performed comparably well in all cases except the case, shown in TABLE VIII and in TABLE IX, of the classifier for EXPERT evidence, which is outperformed by both the n-grams classifier and the random classifier. The reason for this poor performance is currently unknown.

Again, as with claims, the results for selecting the top 50 candidate evidence sentences is shown for comparison, and the maximal average precision is shown in tabular form to the right.

TABLE X shows the results for selecting smaller numbers of candidate evidence sentences. As expected and desired, the precision increases as fewer candidates are selected. For selecting 5 candidates, the overlap scoring reveals that one selection on average, for STUDY or EXPERT evidence types can be selected to overlap with a larger (usually multi-sentence) piece of evidence. As, in other case, likely due to the limited amount of data, the results for ANECDOTAL evidence are relatively poor.

TABLE X

RESULTS FOR SELECTING SMALLER NUMBERS OF RELEVANT EVIDENCE (BEST MATCHES)

STUDY evidence:					overlap			
	P@50	P@20	P@10	P@5	P@50	P@20	P@10	P@5
SSTK+	0.094	0.126	0.140	0.176	0.131	0.162	0.188	0.200
Random	0.006	0.012	0.008	0.016	0.030	0.012	0.020	0.032

EXPERT evidence:					overlap			
	P@50	P@20	P@10	P@5	P@50	P@20	P@10	P@5
SSTK+	0.095	0.120	0.140	0.160	0.125	0.160	0.187	0.207
Random	0.011	0.008	0.007	0.000	0.021	0.025	0.030	0.020

ANECDOTAL evidence:					overlap			
	P@50	P@20	P@10	P@5	P@50	P@20	P@10	P@5
SSTK+	0.036	0.043	0.036	0.014	0.064	0.071	0.079	0.057
Random	0.004	0.004	0.000	0.000	0.013	0.004	0.007	0.029

For classifying pairs of claims and candidate evidence as instances of claims with their topical supporting evidence, the results in for selecting 200 appear similar to classifying candidate evidence, in general. For EXPERT evidence, the SSTK+ classifier significantly outperforms the random and n-gram bag-of words classifiers. It also does well in all other cases except for the cases of ANECDOTAL evidence, for which positive examples are very limited in the data.

TABLE XI

RESULTS FOR CLASSIFYING PAIRS OF SENTENCES AS A CLAIM CORRECTLY OR INCORRECTLY PAIRED WITH CANDIDATE EVIDENCE, WHEN CONSIDERING THE 200 TOP RANKED PAIRS AS POSITIVE PREDICTIONS

If the claim of the first sentence has as evidence a longer run of sentences of which the candidate evidence sentence is a part, it is considered a positive result in the overlap scoring.

Claims paired with STUDY evidence:				overlap		
	F1@200	P@200	R@200	F1@200	P@200	R@200
Maxent	0.125	0.070	0.621	0.221	0.137	0.579
SSTK	0.119	0.065	0.656	0.206	0.125	0.580
SSTK+	0.133	0.074	0.676	0.237	0.147	0.618
n-grams	0.097	0.054	0.514	0.192	0.119	0.509
Random	0.076	0.042	0.444	0.142	0.087	0.398

Positive examples per topic and maximal average precision

STUDY		overlap
avg.	34.76	58.48
max.	117	215
min.	3	6
P@200<	0.12	0.29

Claims paired with EXPERT evidence:				overlap		
	F1@200	P@200	R@200	F1@200	P@200	R@200
Maxent	0.042	0.023	0.195	0.116	0.076	0.239
SSTK	0.140	0.079	0.627	0.228	0.150	0.472
SSTK+	0.169	0.097	0.694	0.314	0.217	0.566
n-grams	0.083	0.046	0.434	0.175	0.113	0.387
Random	0.071	0.039	0.358	0.179	0.119	0.354

Positive examples per topic and maximal average precision

EXPERT		overlap
avg.	31.77	99.7
max.	177	543
min.	3	4
P@200<	0.16	0.50

Claims paired with ANECDOTAL evidence:				overlap		
	F1@200	P@200	R@200	F1@200	P@200	R@200
Maxent	0.047	0.025	0.577	0.132	0.075	0.541
SSTK	0.056	0.029	0.599	0.134	0.077	0.510
SSTK+	0.055	0.029	0.651	0.127	0.072	0.509
n-grams	0.041	0.021	0.488	0.133	0.077	0.487
Random	0.028	0.015	0.267	0.088	0.050	0.356

Positive examples per topic and maximal average precision

ANECDOTAL		overlap
avg.	11.5	46.07
max.	39	237
min.	2	6
P@200<	0.06	0.23

TABLE XII shows the results for selecting 50 top matches. The precision improves noticeably, but, again, does not come close to the maximal values.

TABLE XII

RESULTS FOR CLASSIFYING PAIRS OF SENTENCES AS A CLAIM CORRECTLY OR INCORRECTLY PAIRED WITH CANDIDATE EVIDENCE, WHEN CONSIDERING THE 50 TOP RANKED PAIRS AS POSITIVE PREDICTIONS

If the claim of the first sentence is has as evidence a longer run of sentences of which the candidate evidence sentence is a part, it is considered a positive result in the overlap scoring.

Claims paired with STUDY evidence:				overlap		
	F1@50	P@50	R@50	F1@50	P@50	R@50
Maxent	0.190	0.129	0.357	0.255	0.223	0.299
SSTK	0.166	0.110	0.339	0.221	0.180	0.286
SSTK+	0.220	0.150	0.414	0.299	0.261	0.350
Random	0.068	0.043	0.153	0.119	0.098	0.151

Maximal average precision

STUDY		overlap
P@50<	0.50	1.0

Claims paired with EXPERT evidence:				overlap		
	F1@50	P@50	R@50	F1@50	P@50	R@50
Maxent	0.053	0.035	0.110	0.088	0.085	0.090
SSTK	0.187	0.127	0.353	0.222	0.206	0.241
SSTK+	0.311	0.223	0.516	0.382	0.411	0.357
Random	0.082	0.053	0.180	0.161	0.155	0.169

Maximal average precision

EXPERT		overlap
P@50<	0.64	1.0

Claims paired with ANECDOTAL evidence:				overlap		
	F1@50	P@50	R@50	F1@50	P@50	R@50
Maxent	0.068	0.040	0.228	0.126	0.097	0.182
SSTK	0.064	0.037	0.233	0.142	0.114	0.190
SSTK+	0.080	0.046	0.313	0.151	0.111	0.237
Random	0.023	0.014	0.061	0.069	0.054	0.097

Maximal average precision

ANECDOTAL		overlap
P@50<	0.23	0.92

TABLE XIII

RESULTS FOR SELECTING SMALLER NUMBERS OF RELEVANT PAIRS (BEST MATCHES)

Claims paired with STUDY evidence:					overlap			
	P@50	P@20	P@10	P@5	P@50	P@20	P@10	P@5
SSTK+	0.150	0.264	0.372	0.480	0.261	0.390	0.496	0.592
Random	0.043	0.066	0.036	0.072	0.098	0.102	0.096	0.080

Claims paired with EXPERT evidence:					overlap			
	P@50	P@20	P@10	P@5	P@50	P@20	P@10	P@5
SSTK+	0.223	0.352	0.470	0.600	0.411	0.578	0.683	0.793
Random	0.053	0.058	0.067	0.053	0.155	0.162	0.113	0.133

Claims paired with ANECDOTAL evidence:					overlap			
	P@50	P@20	P@10	P@5	P@50	P@20	P@10	P@5
SSTK+	0.046	0.068	0.079	0.086	0.111	0.150	0.179	0.229
Random	0.014	0.039	0.014	0.057	0.054	0.061	0.079	0.043

For classifying smaller numbers of pairs, the classifiers do better, and note, in particular the very strong result for precision for classifying claims paired with EXPERT evidence. With the SSTK+ classifier, three of five pairings are correct on average when selecting five; with overlap scoring, four of five pairings are correct, indicating that the evidence sentence contains part of a (usually multi-sentence) piece of relevant supporting evidence for the given claim. For STUDY evidence, nearly three of five pairings are correct on average using overlap scoring.

This classifier starts with known claims and pairs them with actual evidence sentences. What is unknown is whether the evidence is a piece of evidence that could support the claim (or even of the correct type). This experiment proceeds such as if the previous step had performed perfectly (except that the type of the evidence need not have been correctly detected). Therefore this task is likely easier than the previous ones. Although there are a large number of negative pairings, the skew between negative and positive is not as great as for the other cases. Therefore, a next step is to use the evidence candidates found in the previous experiment to pair with claims in this

step. If the evidence selection performs well, then we can expect that the pair classification will also perform well. The performance in the case of using low quality pairs, containing many sentences without evidence of any type or that could be matched with any claim is not established, yet, but could be the subject of future research.

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

In this thesis experiments taking up the CDCD and CDED tasks and which extended previous work were described. Two categories of classifier, classifiers using manually designed feature vectors, and ones using tree kernels were tested. As in Lippi and Torroni's [35] experiment, the tree kernel classifiers performed very well, even without any other features. The classifiers using the manually designed features also performed nearly as well, and when some of these features were combined with the tree kernel, the performance was boosted. Because of the demonstrated strong performance of the tree kernel, it can serve as a reasonable benchmark, together with other baselines. The results are encouraging with regard to the relevance and influence of our chosen features which had previously been used successfully on data from other domains or on different but plausibly related tasks.

Kernel machines, such as the support vector machines (SVM) used by Lippi and Torroni, are generally regarded as producing models less easy to interpret than some other types of machine learning classifiers such as logistic regression. Levy et. al. report that their choice of a logistic regression classifier was partly motivated by the model interpretability [2, p. 1493]. However, there exists at least one method for discovering some of the most influential tree fragments from tree kernel models as developed by Pighin and Moschitti (see [57]). (This was attempted, unsuccessfully, due to the current limitations of the available software. Though software has been made available by Pighin, it seems unable to process as much data as was used in these experiments.) Therefore, in future research some of the tree fragments contributing to the success of the model may be discovered, which may advance the study of discourse argumentation theory and the practice of argument mining. As the tree kernels can potentially result in models that overfit the data due to the large number of features considered (many of which may be irrelevant) [45, p. 113], knowing which tree fragments are of most value to the classification should also enable the development of improved models.

All models could be better tuned by cross-validation in the held-out data set in order to produce an optimal comparison, and feature selection could be used to avoid overfitting. For the CDED task, pieces of evidence of up to 3 sentences could be included. That step, together with some additional work would also make it possible to compare the performance with that reported in [3].

It can be noted that although noticeably outperformed by the kernel machines, the maximum entropy model classifiers also did well. These models were significantly less time consuming to train and also to use for classification, even when compared to the fast cutting plane approximation algorithm that was used to train the kernel machine models. As an example, on the same hardware, training maximum entropy models took on the order of minutes, while training kernel machine models took on the order of hours.

Because of this, it is possible to do more experimentation with different feature vectors using maximum entropy models.

Furthermore, the contributions of individual features can be determined from the sum of the corresponding terms in the learned regression equation. This was attempted. In initial attempts, some n-grams seem to have been given large weights in the model due to appearing, by chance, only in positive examples (even though they might have only appeared in one sentence in the entire dataset). Therefore, the model was changed to use only those n-grams appearing at least 10 times in the whole dataset, as noted in an earlier section describing features. Checking the top weighted features for a few topics reveals that n-grams still appear as the top 100 weighted features for making positive predictions (and word pairs for claim and evidence sentence pair classification); however some parse tree fragments containing past tense or past participle verbs appear as highly weighted features for making negative predictions for claims. These results need to be checked and could be examined in detail in future work. Some of the top weighted features are shown in Appendix A. With feature selection and feature reduction, such information also should be expected to be more readily interpretable and usable.

When technology such as this has been better developed, one could expect it to be useful at assisting humans with debate or when researching controversial topics, to discover arguments which have been made (at this point *for*, but not *against* a topic). In addition, such technology could be useful for document summarization. Although it does not attempt to completely map out arguments in the context of the discourse, the amount of argumentative material, its location and its topic (from lists of topics) could be detected, and might be useful in characterizing a document. Likewise, insights into the quality of arguments in essays might be obtained at a superficial level, by comparing the amounts and locations of argument components to those considered effective by experts and by determining how many claims have been supported and with how much evidence. If additional materials are available, more supporting evidence matching the desired claims might be found, which could be used by the author to improve the quality of his or her arguments.

APPENDIX A

FEATURE IMPORTANCE IN THE MAXENT MODEL

The following tables show the top weighted features in the maximum entropy models. The predominance of n-grams and word pairs (when classifying pairs of claims with candidate evidence sentences) seems to suggest some overfitting in these models. The name of the class is shown in the header row, followed by the top weighted features according to their contributions toward classifying an example as an example of the given class; features are in order from largest weighted (at the top) to smallest weight in the corresponding columns.

TABLE XIV

TOP FIVE MOST HIGHLY WEIGHTED FEATURES IN THE MAXIMUM ENTROPY MODEL FOR CLAIMS

0 (negative)	**1 (positive)**
VP->VBD NP	genocide crimes humanity
America	Eugenic
VBD->form	Diet .
VBD->introduce	Protection Law
S->PP NP VP .	to commit crime

TABLE XV

TOP FIVE MOST HIGHLY WEIGHTED FEATURES IN THE MAXIMUM ENTROPY MODEL FOR STUDY EVIDENCE

0 (negative)	**1 (positive)**	**2 (overlap)**
RB->sometimes	Manchu	Attributed
ADVP->RB	Practicing	These issues
``->``	Infringing	religions . ''
IN->by	VBG->infringe	a sacred
IN->while	Disruption of	The constitution

TABLE XVI

TOP FIVE MOST HIGHLY WEIGHTED FEATURES IN THE MAXIMUM ENTROPY MODEL FOR EXPERT EVIDENCE

0 (negative)	**1 (positive)**	**2 (overlap)**
release	the code	W. Bush Administration
euros	attack from	The Who's
ensure	Human Rights Law	Training in
additional	Countries , such	in Human
12	of International Human	World Health Assembly

TABLE XVII

TOP FIVE MOST HIGHLY WEIGHTED FEATURES IN THE MAXIMUM ENTROPY MODEL FOR ANECDOTAL EVIDENCE

0 (negative)	**1 (positive)**	**2 (overlap)**
, Women	gender selection	the Governments
Deliver	layers	of health and
NNP->Adolescent	NNS->chromosome	Western Nations
International Federation of	weight of the	Nations are
International , The	the X	`` Too

In the tables below, showing data for sentence pair classification, features are prefixed with labels: 'WPair' for word pairs; 'Unigram', 'Bigram', or 'Trigram' for n-grams; 'Parse' for parse tree fragments. For n-grams and parse tree fragments, the feature might be a feature from the first or second sentence of the pair. Which case applies is indicated by the use of '1st' or '2nd' in parentheses next to the labels.

TABLE XVIII

TOP FIVE MOST HIGHLY WEIGHTED FEATURES IN THE MAXIMUM ENTROPY MODEL FOR CLAIM AND STUDY EVIDENCE PAIRS

0 (negative)	**1 (positive)**	**2 (overlap)**
[WPair:] governments world	[Bigram (2nd):] , Amnesty	[Bigram (1st):] several years
[WPair:] also ranked	[Bigram (1st):] results .	[Unigram (1st):] 1976
[WPair:] also went	[WPair:] has acts	[Unigram (1st):] forth
[WPair:] corruption Population	[WPair:] has city	[Unigram (1st):] Mao
[WPair:] also Population	[WPair:] has history	[Unigram (1st):] reluctant

TABLE XIX

TOP FIVE MOST HIGHLY WEIGHTED FEATURES IN THE MAXIMUM ENTROPY MODEL FOR CLAIM AND EXPERT EVIDENCE PAIRS

0 (negative)	**1 (positive)**	**2 (overlap)**
[Parse (2nd):] VB->damage	[WPair:] UN City	[WPair:] City as
[Parse (2nd):] VBP->predict	[WPair:] Population policies	[WPair:] City as
[Parse (2nd):] NN->visitor	[WPair:] Population led	[WPair:] City Health
[Parse (2nd):] NN->tourism	[WPair:] Population countries	[WPair:] City rights
[Parse (2nd):] NN->observation	[WPair:] Population attack	[WPair:] Conference Health

TABLE XX

TOP FIVE MOST HIGHLY WEIGHTED FEATURES IN THE MAXIMUM ENTROPY MODEL FOR CLAIM AND ANECDOTAL EVIDENCE PAIRS

0 (negative)	**1 (positive)**	**2 (overlap)**
[WPair:] as forests	[Bigram (2^{nd}):] states many	[WPair:] children people
[WPair:] as atmosphere	[Unigram (2^{nd}):] variety	[WPair:] thousands have
[WPair:] life President	[Bigram (2^{nd}):] ’’ by	[WPair:] thousands Population
[WPair:] life Peter	[Bigram (2^{nd}):] ’s natural	[WPair:] affected population
[WPair:] as Says	[Bigram (2^{nd}):] colon an	[WPair:] children reports

APPENDIX B

IMPLEMENTATION DETAILS

Model Training Parameters

The maximum entropy model classifiers were trained using the "Maximum Entropy Modeling Toolkit for Python and C++" by Zhang Le. The training parameters given to the program were, "-g 0.5 -i 100", where *-g* gives the value of the Gaussian prior, and *–i* gives the value for the number of iterations.

The support vector machine model classifiers were trained using "uSVM-TK2.0" ('uniform sampling' with the cutting plane algorithm, as a fast approximation to the support vector machine model), by Alaiksei Severyn and Alessandro Moschitti [44], and based on SVM-*light* by Thorsten Joachims [46]. The training parameters given to the program for the SSTK+ model (model combining trees and vectors) claims or evidence were, " -t 5 -C + -j 25", where: the *-t* parameter with a value of '5' selects a combination of tree kernels and vectors according to the settings of parameters such as *-C*; the *-C* parameter with a value of '+' combines trees and vectors by adding their contributions; the *-j* parameter selects the "cost factor, by which training errors on positive examples outweigh errors on negative examples (default 1)."[27]

For training the SSTK (trees only model), the parameter *-C* was given the value 'T', meaning use trees only.

For training the SSTK+ model for pairs of claims and candidate evidence sentence, the additional parameter, *-W*, was given with the value of 'A', which instructs the program to apply tree kernels to all pairs of trees, rather than applying to only corresponding tree pairs and then summing, e.g., sentence 1 of example 1 can be compared with both sentence 1 and sentence 2 of

[27] The quoted text is from the software's command line help message.

other examples. The parameter, *-j*, was given the value of 10 for this model and for the corresponding trees only model.

The parameter 'C', the soft-margins parameter of the SVM model, was left at the default value, which is one over the square root of the average of the values of training examples.

Tools Developed

Custom tools were developed to extract features and store them in the database and to fetch features from the database and output them to features files.

`reconstruct_sentences`

Running this tool with a list of articles to process reconstructs the original sentences from articles and stores them in the database. The Stanford CoreNLP parser provides only a list of tokens, but includes the offsets in the original file, so that the original sentence can be reconstructed. The original sentences also contain "[REF]" tokens which were removed prior to parsing. In addition to reconstructing the original sentences, this tool also finds paragraph boundaries, which are indicated in this dataset by newlines. Newlines in this corpus generally only occur at paragraph boundaries, but they also delimit headers and items in lists. No attempt is made in this tool to distinguish those uses, but they are distinguished when extracting the feature, "section numbers" (see below).

`store_features`[28]

Given a list of articles to process, any of: n-grams, lemmatization of the parse tree, positional features, word2vec similarity with topic, WordNet similarity with topic, and section numbers can be extracted and stored in the database.

The n-grams stored are described under features in the previous section. The lemmatized parse tree is a modification of the parse tree provided by the Stanford parser, in which the word forms (which are at the tree leaves) are replaced by their respective lemmas. Positional features are the sentence position in a paragraph and paragraph position in each document. The reconstruct_sentences tool must be used first in order to store the paragraph boundaries needed by this tool. Section numbers are relative to section boundaries. A new section is considered to start when a header is found and a first section is started at the beginning of the document whether or not there is a header. Header boundaries are discovered using an SQL query described in the appendix. The header boundary feature must be stored in the database before using this tool. Word vectors are computed using the word2vec component of the python tool, gensim. It is provided with a precomputed model learned from a 6 billion word Wikipedia corpus. Sentiment and subjectivity scores are computing using the rule based, TextBlob, python tool. It provides a polarity based on basic indicators of negation, such as the presence of the word "not" or of words with negative polarity. WordNet similarity is computed using the WNSim tool from the University of Illinois, which is a graph distance based measure. It combines distance scores from the WordNet hypernym hierarchy and meronym hierarchies and also considers antonyms in order to possibly change the sign of the match. These two similarity

[28] This tool is currently in the form of 6 separate programs: store_ngram_and_lemmatized_parse, store_positional_features, store_section_numbers, store_w2v_and_sentiment, store_wn_similarity, and store_claim_cross_evidence_similarity_scores. Each of these extracts and stores some subset of the total feature set, as implied by its name.

tools are used to provide similarity scores for the comparison between claims and topics (where each article is a member of a set of articles on a particular topic), and separately between claims and supporting evidence.

`extract_with_claim_labels`
`extract_with_evidence_labels` (currently part of the previous tool)

Extracts all features from the database relevant, respectively, to classifying claims or evidence and puts them in one file per article and in one separate directory per topic. It also gives heldout topic directories distinctive names (prefixed with the word "heldout"). The lemmatized parse tree stored in the database is converted into a bag of tree production features; that is, a feature for each expansion of a node down to one depth, for all nodes except leaves was generated. These features correspond to parse production rules (though they may not have been the actual rules used by the parser).

`extract_claims_cross_evidence_features`

Extracts all features from the database relevant to matching claims and supporting evidence. And it also separately generates word pair features.

REFERENCES

[1] D. Walton, “Argumentation Theory: A Very Short Introduction,” in *Argumentation in Artificial Intelligence*, G. Simari and I. Rahwan, Eds. Boston, MA: Springer US, 2009, pp. 1–22.

[2] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim, “Context Dependent Claim Detection,” in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 2014, pp. 1489–1500.

[3] R. Rinott, L. Dankin, C. Alzate Perez, M. M. Khapra, E. Aharoni, and N. Slonim, “Show Me Your Evidence - an Automatic Method for Context Dependent Evidence Detection,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 2015, pp. 440–450.

[4] F. H. Van Eemeren, Handbook of argumentation theory, 1st edition. New York: Springer, 2014.

[5] S. E. Toulmin, The Uses of Argument, Updated Edition. Cambridge, MA, USA: Cambridge University Press, 2003.

[6] R. Koons, “Defeasible Reasoning,” in The Stanford Encyclopedia of Philosophy, Spring 2014., E. N. Zalta, Ed. 2014.

[7] C. Strasser and G. A. Antonelli, “Non-monotonic Logic,” in The Stanford Encyclopedia of Philosophy, Fall 2015., E. N. Zalta, Ed. 2015.

[8] B. Verheij, “The Toulmin Argument Model in Artificial Intelligence,” in Argumentation in Artificial Intelligence, G. Simari and I. Rahwan, Eds. Boston, MA: Springer US, 2009, pp. 219–238.

[9] J. L. Pollock, “A Recursive Semantics for Defeasible Reasoning,” in Argumentation in Artificial Intelligence, G. Simari and I. Rahwan, Eds. Boston, MA: Springer US, 2009, pp. 173–197.

[10] C. Reed and F. Grasso, “Recent advances in computational models of natural argument,” Int. J. Intell. Syst., vol. 22, no. 1, pp. 1–15, Jan. 2007.

[11] E. Cabrio and S. Villata, “Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interactions.,” in The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers, 2012, pp. 208–212.

[12] P. M. Dung, “On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games.,” Artif Intell, vol. 77, no. 2, pp. 321–358, 1995.

[13] P. Baroni and M. Giacomin, “Semantics of Abstract Argument Systems,” in Argumentation in Artificial Intelligence, G. Simari and I. Rahwan, Eds. Boston, MA: Springer US, 2009, pp. 25–44.

[14] C. Cayrol and M.-C. Lagasquie-Schiex, “Bipolar abstract argumentation systems,” in Argumentation in Artificial Intelligence, G. Simari and I. Rahwan, Eds. Boston, MA: Springer US, 2009, pp. 65–84.

[15] A. J. García and G. R. Simari, “Defeasible logic programming: An argumentative approach,” Theory Pract. Log. Program., vol. 4, no. 1+ 2, pp. 95–138, 2004.

[16] E. Cabrio, S. Tonelli, and S. Villata, “From Discourse Analysis to Argumentation Schemes and Back: Relations and Differences.,” in Computational Logic in Multi-Agent Systems - 14th International Workshop, CLIMA XIV, Corunna, Spain, September 16-18, 2013. Proceedings, 2013, pp. 1–17.

[17] T. Bench-Capon and Katie Atkinson, “Argumentation Schemes: From Informal Logic to Computational Models,” in Dialectics, Dialogue, and Argumentation: An Examination of Douglas Walton’s Theories of Reasoning and Argument, London, UK: College Publications, 2010, pp. 103–113.

[18] V. W. Feng and G. Hirst, “Classifying arguments by scheme,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, 2011, pp. 987–996.

[19] J. Lawrence and C. Reed, “Combining Argument Mining Techniques,” in Proceedings of the 2nd Workshop on Argumentation Mining, Denver, CO, 2015, pp. 127–136.

[20] A. Peldszus, “Towards segment-based recognition of argumentation structure in short texts,” in Proceedings of the First Workshop on Argumentation Mining, Baltimore, Maryland, 2014, pp. 88–97.

[21] S. Teufel and M. Moens, “Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status,” Comput. Linguist., vol. 28, no. 4, pp. 409–445, 2002.

[22] R. M. Palau and M.-F. Moens, “Argumentation mining,” Artif Intell Law, vol. 19, no. 1, pp. 1–22, 2011.

[23] S. Wells, “Argument Mining: Was Ist Das?,” presented at the the 14th Int. Workshop on Computational Models of Natural Argument (CMNA14), Krakow, Poland, 2014.

[24] J. Lawrence, C. Reed, C. Allen, S. McAlister, and A. Ravenscroft, “Mining Arguments From 19th Century Philosophical Texts Using Topic Based Modelling,” in Proceedings of the First Workshop on Argumentation Mining, Baltimore, Maryland, 2014, pp. 79–87.

[25] M.-F. Moens, E. Boiy, R. M. Palau, and C. Reed, “Automatic detection of arguments in legal texts,” in The Eleventh International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 4-8, 2007, Stanford Law School, Stanford, California, USA, 2007, pp. 225–230.

[26] M. P. G. Villalba and P. Saint-Dizier, “Some Facets of Argument Mining for Opinion Analysis,” Front. Artif. Intell. Appl., pp. 23–34, 2012.

[27] M. Lippi and P. Torroni, “Argumentation Mining: State of the Art and Emerging Trends,” ACM Trans Internet Technol, vol. 16, no. 2, p. 10:1–10:25, Mar. 2016.

[28] F. H. Van Eemeren and R. Grootendorst, A systematic theory of argumentation: The pragma-dialectical approach, vol. 14. Cambridge University Press, 2004.

[29] C. Stab and I. Gurevych, “Identifying Argumentative Discourse Structures in Persuasive Essays,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 46–56.

[30] L. Carstens and F. Toni, “Towards relation based Argumentation Mining,” in Proceedings of the 2nd Workshop on Argumentation Mining, Denver, CO, 2015, pp. 29–34.

[31] I. Habernal, J. Eckle-Kohler, and I. Gurevych, “Argumentation Mining on the Web from Information Seeking Perspective.,” in ArgNLP, 2014.

[32] Z. Lin, M.-Y. Kan, and H. T. Ng, “Recognizing Implicit Discourse Relations in the Penn Discourse Treebank,” in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1, Stroudsburg, PA, USA, 2009, pp. 343–351.

[33] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, 2nd Ed. Upper Saddle River, NJ, USA: Pearson Education, Inc., 2008.

[34] D. Marcu and A. Echihabi, “An Unsupervised Approach to Recognizing Discourse Relations,” in Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 368–375.

[35] M. Lippi and P. Torroni, “Context-independent Claim Detection for Argument Mining,” in Proceedings of the 24th International Conference on Artificial Intelligence, Buenos Aires, Argentina, 2015, pp. 185–191.

[36] E. Aharoni et al., “A Benchmark Dataset for Automatic Detection of Claims and Evidence in the Context of Controversial Topics,” in Proceedings of the First Workshop on Argumentation Mining, Baltimore, Maryland, 2014, pp. 64–68.

[37] R. Srikant and R. Agrawal, “Mining Sequential Patterns: Generalizations and Performance Improvements,” in Advances in Database Technology - EDBT’96, 5th International Conference on Extending Database Technology, Avignon, France, March 25-29, 1996, Proceedings, 1996, pp. 3–17.

[38] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford, “Okapi at TREC-3,” 1996, pp. 109–126.

[39] M. Lippi and P. Torroni, “Argument Mining: A Machine Learning Perspective,” in Theory and Applications of Formal Argumentation: Third International Workshop, TAFA 2015, Buenos Aires, Argentina, July 25-26, 2015, Revised Selected Papers, E. Black, S. Modgil, and N. Oren, Eds. Cham: Springer International Publishing, 2015, pp. 163–176.

[40] R. M. Palau and M.-F. Moens, “Argumentation mining: the detection, classification and structure of arguments in text,” in The 12th International Conference on Artificial Intelligence and Law, Proceedings of the Conference, June 8-12, 2009, Barcelona, Spain, 2009, pp. 98–107.

[41] A. Moschitti, “Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees,” in Machine Learning: ECML 2006, 17th European Conference on Machine Learning, Berlin, Germany, September 18-22, 2006, Proceedings, 2006, pp. 318–329.

[42] M. Collins and N. Duffy, “New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron,” in Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2002, pp. 263–270.

[43] S. Filice, G. D. S. Martino, and A. Moschitti, “Structural Representations for Learning Relations between Pairs of Texts,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, 2015, pp. 1003–1013.

[44] A. Severyn and A. Moschitti, “Large-Scale Support Vector Learning with Structural Kernels,” in Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III, 2010, pp. 229–244.

[45] A. Moschitti, “Making Tree Kernels Practical for Natural Language Learning,” in EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy, 2006, pp. 113–120.

[46] T. Joachims, “Advances in Kernel Methods,” B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 169–184.

[47] The PDTB Research Group, “The PDTB 2.0. Annotation Manual,” Institute for Research in Cognitive Science, University of Pennsylvania, Technical Report IRCS-08-01, 2008.

[48] M.-C. de Marneffe and C. D. Manning, “Stanford typed dependencies manual.” 03-Sep-2008.

[49] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” in Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland, 2014, pp. 55–60.

[50] Q. Do, D. Roth, M. Sammons, Y. Tu, and V. Vydiswaran, “Robust, light-weight approaches to compute lexical similarity,” Comput. Sci. Res. Tech. Rep. Univ. Ill., p. 94, 2009.

[51] G. A. Miller, “WordNet: A Lexical Database for English,” Commun ACM, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[52] C. Fellbaum, Ed., WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press, 1998.

[53] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 2010, pp. 45–50.

[54] L. Padró and E. Stanilovsky, “FreeLing 3.0: Towards Wider Multilinguality,” in Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012, 2012, pp. 2473–2479.

[55] R. Levy, L. Ein-Dor, S. Hummel, R. Rinott, and N. Slonim, “TR9856: A Multi-word Term Relatedness Benchmark,” in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, 2015, pp. 419–424.

[56] S. Bhagwani, S. Satapathy, and H. Karnick, “sranjans : Semantic Textual Similarity using Maximal Weighted Bipartite Graph Matching,” in *SEM 2012: 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 (SemEval 2012), Montréal, Canada, 2012, pp. 579–585.

[57] D. Pighin and A. Moschitti, “On Reverse Feature Engineering of Syntactic Tree Kernels,” in Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL 2010, Uppsala, Sweden, July 15-16, 2010, 2010, pp. 223–233.

BIOGRAPHICAL SKETCH

Waleed Mebane is a Master of Science student in the Department of Computer Science at the Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas. Waleed's fields of interest are natural language argumentation, argument mining, natural language processing, artificial intelligence, and machine learning.

CURRICULUM VITAE

Waleed Mebane

Student

Master of Science in Computer Science Program
Department of Computer Science
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
The University of Texas at Dallas
800 W. Campbell Road
Richardson, TX 75080 U.S.A.

RESEARCH INTERESTS

Natural language processing, argumentation, argument mining, artificial intelligence, advanced programming languages

TECHNICAL SKILLS

Database programming, functional programming, logic programming