Classifying Arguments by Scheme

Vanessa Wei Feng
Department of Computer Science
University of Toronto
Toronto, ON, M5S 3G4, Canada
weifeng@cs.toronto.edu

Graeme Hirst
Department of Computer Science
University of Toronto
Toronto, ON, M5S 3G4, Canada
gh@cs.toronto.edu

Abstract

Argumentation schemes are structures or templates for various kinds of arguments. Given the text of an argument with premises and conclusion identified, we classify it as an instance of one of five common schemes, using features specific to each scheme. We achieve accuracies of 63–91% in one-against-others classification and 80–94% in pairwise classification (baseline = 50% in both cases).

1 Introduction

We investigate a new task in the computational analysis of arguments: the classification of arguments by the *argumentation schemes* that they use. An argumentation scheme, informally, is a framework or structure for a (possibly defeasible) argument; we will give a more-formal definition and examples in Section 3. Our work is motivated by the need to determine the unstated (or implicitly stated) premises that arguments written in natural language normally draw on. Such premises are called *enthymemes*.

For instance, the argument in Example 1 consists of one explicit premise (the first sentence) and a conclusion (the second sentence):

Example 1 ***[Premise:]*** *The survival of the entire world is at stake.*
[Conclusion:] *The treaties and covenants aiming for a world free of nuclear arsenals and other conventional and biological weapons of mass destruction should be adhered to scrupulously by all nations.*

Another premise is left implicit — "*Adhering to those treaties and covenants is a means of realizing survival of the entire world*". This proposition is an enthymeme of this argument.

Our ultimate goal is to reconstruct the enthymemes in an argument, because determining these unstated assumptions is an integral part of understanding, supporting, or attacking an entire argument. Hence reconstructing enthymemes is an important problem in argument understanding. We believe that first identifying the particular argumentation scheme that an argument is using will help to bridge the gap between stated and unstated propositions in the argument, because each argumentation scheme is a relatively fixed "template" for arguing. That is, given an argument, we first classify its argumentation scheme; then we fit the stated propositions into the corresponding template; and from this we infer the enthymemes.

In this paper, we present an argument scheme classification system as a stage following argument detection and proposition classification. First in Section 2 and Section 3, we introduce the background to our work, including related work in this field, the two core concepts of argumentation schemes and scheme-sets, and the Araucaria dataset. In Section 4 and Section 5 we present our classification system, including the overall framework, data preprocessing, feature selection, and the experimental setups. In the remaining section, we present the essential approaches to solve the leftover problems of this paper which we will study in our future work, and discuss the experimental results, and potential directions for future work.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 987–996,
Portland, Oregon, June 19-24, 2011.

2 Related work

Argumentation has not received a great deal of attention in computational linguistics, although it has been a topic of interest for many years. Cohen (1987) presented a computational model of argumentative discourse. Dick (1987; 1991a; 1991b) developed a representation for retrieval of judicial decisions by the structure of their legal argument — a necessity for finding legal precedents independent of their domain. However, at that time no corpus of arguments was available, so Dick's system was purely theoretical. Recently, the Araucaria project at University of Dundee has developed a software tool for manual argument analysis, with a point-and-click interface for users to reconstruct and diagram an argument (Reed and Rowe, 2004; Rowe and Reed, 2008). The project also maintains an online repository, called AraucariaDB, of marked-up naturally occurring arguments collected by annotators worldwide, which can be used as an experimental corpus for automatic argumentation analysis (for details see Section 3.2).

Recent work on argument interpretation includes that of George, Zukerman, and Nieman (2007), who interpret constructed-example arguments (not naturally occurring text) as Bayesian networks. Other contemporary research has looked at the automatic detection of arguments in text and the classification of premises and conclusions. The work closest to ours is perhaps that of Mochales and Moens (2007; 2008; 2009a; 2009b). In their early work, they focused on automatic detection of arguments in legal texts. With each sentence represented as a vector of shallow features, they trained a multinomial naïve Bayes classifier and a maximum entropy model on the Araucaria corpus, and obtained a best average accuracy of 73.75%. In their follow-up work, they trained a support vector machine to further classify each argumentative clause into a premise or a conclusion, with an F_1 measure of 68.12% and 74.07% respectively. In addition, their context-free grammar for argumentation structure parsing obtained around 60% accuracy.

Our work is "downstream" from that of Mochales and Moens. Assuming the eventual success of their, or others', research program on detecting and classifying the components of an argument, we seek to determine how the pieces fit together as an instance of an argumentation scheme.

3 Argumentation schemes, scheme-sets, and annotation

3.1 Definition and examples

Argumentation schemes are structures or templates for forms of arguments. The arguments need not be deductive or inductive; on the contrary, most argumentation schemes are for *presumptive* or *defeasible* arguments (Walton and Reed, 2002). For example, *argument from cause to effect* is a commonly used scheme in everyday arguments. A list of such argumentation schemes is called a *scheme-set*.

It has been shown that argumentation schemes are useful in evaluating common arguments as fallacious or not (van Eemeren and Grootendorst, 1992). In order to judge the weakness of an argument, a set of critical questions are asked according to the particular scheme that the argument is using, and the argument is regarded as valid if it matches all the requirements imposed by the scheme.

Walton's set of 65 argumentation schemes (Walton et al., 2008) is one of the best-developed scheme-sets in argumentation theory. The five schemes defined in Table 1 are the most commonly used ones, and they are the focus of the scheme classification system that we will describe in this paper.

3.2 Araucaria dataset

One of the challenges for automatic argumentation analysis is that suitable annotated corpora are still very rare, in spite of work by many researchers. In the work described here, we use the Araucaria database[1], an online repository of arguments, as our experimental dataset. Araucaria includes approximately 660 manually annotated arguments from various sources, such as newspapers and court cases, and keeps growing. Although Araucaria has several limitations, such as rather small size and low agreement among annotators[2], it is nonetheless one of the best argumentative corpora available to date.

[1] http://araucaria.computing.dundee.ac.uk/doku.php#araucaria_argumentation_corpus

[2] The developers of Araucaria did not report on inter-annotator agreement, probably because some arguments are annotated by only one commentator.

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*.
Argument from cause to effect *Major premise*: Generally, if *A* occurs, then *B* will (might) occur. *Minor premise*: In this case, *A* occurs (might occur). *Conclusion*: Therefore, in this case, *B* will (might) occur.
Practical reasoning *Major premise*: I have a goal *G*. *Minor premise*: Carrying out action *A* is a means to realize *G*. *Conclusion*: Therefore, I ought (practically speaking) to carry out this action *A*.
Argument from consequences *Premise*: If *A* is (is not) brought about, good (bad) consequences will (will not) plausibly occur. *Conclusion*: Therefore, *A* should (should not) be brought about.
Argument from verbal classification *Individual premise*: *a* has a particular property *F*. *Classification premise*: For all *x*, if *x* has property *F*, then *x* can be classified as having property *G*. *Conclusion*: Therefore, *a* has property *G*.

Table 1: The five most frequent schemes and their definitions in Walton's scheme-set.

Arguments in Araucaria are annotated in a XML-based format called "AML" (Argument Markup Language). A typical argument (see Example 2) consists of several AU nodes. Each AU node is a complete argument unit, composed of a conclusion proposition followed by optional premise proposition(s) in a linked or convergent structure. Each of these propositions can be further defined as a hierarchical collection of smaller AUs. INSCHEME is the particular scheme (e.g., *"Argument from Consequences"*) of which the current proposition is a member; enthymemes that have been made explicit are annotated as "missing = yes".

Example 2 ***Example of argument markup from Araucaria***

```
<TEXT>If we stop the free creation of art, we will stop
 the free viewing of art.</TEXT>
<AU>
  <PROP identifier="C" missing="yes">
    <PROPTEXT offset="-1">
      The prohibition of the free creation of art should
      not be brought about.</PROPTEXT>
    <INSCHEME scheme="Argument from Consequences"
     schid="0" />
  </PROP>
  <LA>
    <AU>
      <PROP identifier="A" missing="no">
        <PROPTEXT offset="0">
          If we stop the free creation of art, we will
          stop the free viewing of art.</PROPTEXT>
        <INSCHEME scheme="Argument from Consequences"
         schid="0" />
      </PROP>
    </AU>
    <AU>
      <PROP identifier="B" missing="yes">
        <PROPTEXT offset="-1">
          The prohibition of free viewing of art is not
          acceptable.</PROPTEXT>
        <INSCHEME scheme="Argument from Consequences"
        schid="0" />
      </PROP>
    </AU>
  </LA>
</AU>
```

There are three scheme-sets used in the annotations in Araucaria: Walton's scheme-set, Katzav and Reed's (2004) scheme-set, and Pollock's (1995) scheme-set. Each of these has a different set of schemes; and most arguments in Araucaria are marked up according to only one of them. Our experimental dataset is composed of only those arguments annotated in accordance with Walton's scheme-set, within which the five schemes shown in Table 1 constitute 61% of the total occurrences.

4 Methods

4.1 Overall framework

As we noted above, our ultimate goal is to reconstruct enthymemes, the unstated premises, in an argument by taking advantage of the stated propositions; and in order to achieve this goal we need to first determine the particular argumentation scheme that the argument is using. This problem is depicted in Figure 1. Our scheme classifier is the dashed round-cornered rectangle portion of this

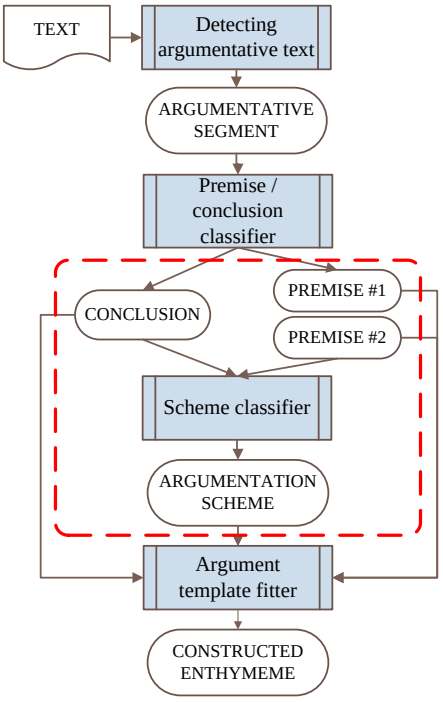

Figure 1: Overall framework of this research.

overall framework: its input is the extracted conclusion and premise(s) determined by an argument detector, followed by a premise / conclusion classifier, given an unknown text as the input to the entire system. And the portion below the dashed round-rectangle represents our long-term goal — to reconstruct the implicit premise(s) in an argument, given its argumentation scheme and its explicit conclusion and premise(s) as input. Since argument detection and classification are not the topic of this paper, we assume here that the input conclusion and premise(s) have already been retrieved, segmented, and classified, as for example by the methods of Mochales and Moens (see Section 2 above). And the scheme template fitter is the topic of our on-going work.

4.2 Data preprocessing

From all arguments in Araucaria, we first extract those annotated in accordance with Walton's scheme-set. Then we break each complex AU node into several simple AUs where no conclusion or premise proposition nodes have embedded AU nodes. From these generated simple arguments, we extract those whose scheme falls into one of the five most frequent schemes as described in Table 1. Furthermore, we remove all enthymemes that have been inserted by the annotator and ignore any argument with a missing conclusion, since the input to our proposed classifier, as depicted in Figure 1, cannot have any access to unstated argumentative propositions.

The resulting preprocessed dataset is composed of 393 arguments, of which 149, 106, 53, 44, and 41 respectively belong to the five schemes in the order shown in Table 1.

4.3 Feature selection

The features used in our work fall into two categories: general features and scheme-specific features.

4.3.1 General features

General features are applicable to arguments belonging to any of the five schemes (shown in Table 2).

For the features **conLoc**, **premLoc**, **gap**, and **lenRat**, we have two versions, differing in terms of their basic measurement unit: *sentence*-based and *token*-based. The final feature, **type**, indicates whether the premises contribute to the conclusion in a linked or convergent order. A *linked argument* (LA) is one that has two or more inter-dependent premise propositions, all of which are necessary to make the conclusion valid, whereas in a *convergent argument* (CA) exactly one premise proposition is sufficient to do so. Since it is observed that there exists a strong correlation between **type** and the particular scheme employed while arguing, we believe **type** can be a good indicator of argumentation scheme. However, although this feature is available to us because it is included in the Araucaria annotations, its value cannot be obtained from raw text as easily as other features mentioned above; but it is possible that we will in the future be able to determine it automatically by taking advantage of some scheme-independent cues such as the discourse relation between the conclusion and the premises.

4.3.2 Scheme-specific features

Scheme-specific features are different for each scheme, since each scheme has its own cue phrases or patterns. The features for each scheme are shown in Table 3 (for complete lists of features see Feng (2010)). In our experiments in Section 5 below, all these features are computed for all arguments; but

conLoc: the location (in token or sentence) of the conclusion in the text.

premLoc: the location (in token or sentence) of the first premise proposition.

conFirst: whether the conclusion appears before the first premise proposition.

gap: the interval (in token or sentence) between the conclusion and the first premise proposition.

lenRat: the ratio of the length (in token or sentence) of the premise(s) to that of the conclusion.

numPrem: the number of explicit premise propositions (PROP nodes) in the argument.

type: type of argumentation structure, i.e., linked or convergent.

Table 2: List of general features.

the features for any particular scheme are used only when it is the subject of a particular task. For example, when we classify *argument from example* in a one-against-others setup, we use the scheme-specific features of that scheme for all arguments; when we classify *argument from example* against *argument from cause to effect*, we use the scheme-specific features of those two schemes.

For the first three schemes (*argument from example*, *argument from cause to effect*, and *practical reasoning*), the scheme-specific features are selected cue phrases or patterns that are believed to be indicative of each scheme. Since these cue phrases and patterns have differing qualities in terms of their precision and recall, we do not treat them all equally. For each cue phrase or pattern, we compute "confidence", the degree of belief that the argument of interest belongs to a particular scheme, using the distribution characteristics of the cue phrase or pattern in the corpus, as described below.

For each argument $\mathcal{A}$, a vector $\mathbf{CV} = \{c_1, c_2, c_3\}$ is added to its feature set, where each c_i indicates the "confidence" of the existence of the specific features associated with each of the first three schemes, $scheme_i$. This is defined in Equation 1:

$$c_i = \frac{1}{N} \sum_{k=1}^{m_i} (P(scheme_i|cp_k) \cdot d_{ik}) \qquad (1)$$

Argument from example
8 keywords and phrases including *for example*, *such as*, *for instance*, etc.; 3 punctuation cues: ":", ";", and "—".

Argument from cause to effect
22 keywords and simple cue phrases including *result*, *related to*, *lead to*, etc.; 10 causal and non-causal relation patterns extracted from WordNet (Girju, 2003).

Practical reasoning
28 keywords and phrases including *want*, *aim*, *objective*, etc.; 4 modal verbs: *should*, *could*, *must*, and *need*; 4 patterns including imperatives and infinitives indicating the goal of the speaker.

Argument from consequences
The counts of positive and negative propositions in the conclusion and premises, calculated from the *General Inquirer*[2].

Argument from verbal classification
The maximal similarity between the *central word* pairs extracted from the conclusion and the premise; the counts of *copula*, *expletive*, and *negative modifier* dependency relations returned by the *Stanford parser*[3] in the conclusion and the premise.

[2] http://www.wjh.harvard.edu/~inquirer/
[3] http://nlp.stanford.edu/software/lex-parser.shtml

Table 3: List of scheme-specific features.

Here m_i is the number of scheme-specific cue phrases designed for $scheme_i$; $P(scheme_i|cp_k)$ is the prior probability that the argument $\mathcal{A}$ actually belongs to $scheme_i$, given that some particular cue phrase cp_k is found in $\mathcal{A}$; d_{ik} is a value indicating whether cp_k is found in $\mathcal{A}$; and the normalization factor N is the number of scheme-specific cue phrase patterns designed for $scheme_i$ with at least one support (at least one of the arguments belonging to $scheme_i$ contains that cue phrase). There are two ways to calculate d_{ik}, *Boolean* and *count*: in *Boolean* mode, d_{ik} is treated as 1 if $\mathcal{A}$ matches cp_k; in *count* mode, d_{ik} equals to the number of times $\mathcal{A}$ matches cp_k; and in both modes, d_{ik} is treated as 0 if cp_k is not found in $\mathcal{A}$.

For *argument from consequences*, since the arguer has an obvious preference for some particular consequence, sentiment orientation can be a good indicator for this scheme, which is quantified by the counts of positive and negative propositions in the conclusion and premise.

For *argument from verbal classification*, there exists a hypernymy-like relation between some pair of propositions (entities, concepts, or actions) located in the conclusion and the premise respectively. The existence of such a relation is quantified by the maximal Jiang-Conrath Similarity (Jiang and Conrath, 1997) between the "central word" pairs extracted from the conclusion and the premise. We parse each sentence of the argument with the Stanford dependency parser, and a word or phrase is considered to be a central word if it is the dependent or governor of several particular dependency relations, which basically represents the attribute or the action of an entity in a sentence, or the entity itself. For example, if a word or phrase is the dependent of the dependency relation *agent*, it is therefore considered as a "central word". In addition, an arguer tends to use several particular syntactic structures (*copula*, *expletive*, and *negative modifier*) when using this scheme, which can be quantified by the counts of those special relations in the conclusion and the premise(s).

5 Experiments

5.1 Training

We experiment with two kinds of classification: *one-against-others* and *pairwise*. We build a pruned C4.5 decision tree (Quinlan, 1993) for each different classification setup, implemented by Weka Toolkit 3.6[5] (Hall et al., 2009).

One-against-others classification A one-against-others classifier is constructed for each of the five most frequent schemes, using the general features and the scheme-specific features for the scheme of interest. For each classifier, there are two possible outcomes: *target_scheme* and other; 50% of the training dataset is arguments associated with *target_scheme*, while the rest is arguments of all the other schemes, which are treated as other. One-against-other classification thus tests the effectiveness of each scheme's specific features.

Pairwise classification A pairwise classifier is constructed for each of the ten possible pairings of the five schemes, using the general features and the scheme-specific features of the two schemes in the pair. For each of the ten classifiers, the training dataset is divided equally into arguments belonging to $scheme_1$ and arguments belonging to $scheme_2$, where $scheme_1$ and $scheme_2$ are two different schemes among the five. Only features associated with $scheme_1$ and $scheme_2$ are used.

5.2 Evaluation

We experiment with different combinations of general features and scheme-specific features (discussed in Section 4.3). To evaluate each experiment, we use the average accuracy over 10 pools of randomly sampled data (each with baseline at 50%[6]) with 10-fold cross-validation.

6 Results

We first present the best average accuracy (BAA) of each classification setup. Then we demonstrate the impact of the feature **type** (convergent or linked argument) on BAAs for different classification setups, since we believe **type** is strongly correlated with the particular argumentation scheme and its value is the only one directly retrieved from the annotations of the training corpus. For more details, see Feng (2010).

6.1 BAAs of each classification setup

target_scheme	*BAA*	d_{ik}	*base*	*type*
example	90.6	count	token	yes
cause	70.4	Boolean / count	token	no
reasoning	90.8	count	sentence	yes
consequences	62.9	–	sentence	yes
classification	63.2	–	token	yes

Table 4: Best average accuracies (BAAs) (%) of one-against-others classification.

[5] http://cs.waikato.ac.nz/ml/weka

[6] We also experiment with using general features only, but the results are consistently below or around the sampling baseline of 50%; therefore, we do not use them as a baseline here.

	example	*cause*	*reasoning*	*consequences*
cause	80.6			
reasoning	93.1	94.2		
consequences	86.9	86.7	97.9	
classification	86.0	85.6	98.3	64.2

Table 5: Best average accuracies (BAAs) (%) of pairwise classification.

Table 4 presents the best average accuracies of one-against-others classification for each of the five schemes. The subsequent three columns list the particular strategies of features incorporation under which those BAAs are achieved (the complete set of possible choices is given in Section 4.3.):

- $\mathbf{d_{ik}}$: *Boolean* or *count* — the strategy of combining scheme-specific cue phrases or patterns using either *Boolean* or *count* for d_{ik}.
- **base**: *sentence* or *token* — the basic unit of applying location- or length-related general features.
- **type**: *yes* or *no* — whether **type** (convergent or linked argument) is incorporated into the feature set.

As Table 4 shows, one-against-others classification achieves high accuracy for *argument from example* and *practical reasoning*: 90.6% and 90.8%. The BAA of *argument from cause to effect* is only just over 70%. However, with the last two schemes (*argument from consequences* and *argument from verbal classification*), accuracy is only in the low 60s; there is little improvement of our system over the majority baseline of 50%. This is probably due at least partly to the fact that these schemes do not have such obvious cue phrases or patterns as the other three schemes which therefore may require more world knowledge encoded, and also because the available training data for each is relatively small (44 and 41 instances, respectively). The BAA for each scheme is achieved with inconsistent choices of base and d_{ik}, but the accuracies that resulted from different choices vary only by very little.

Table 5 shows that our system is able to correctly differentiate between most of the different scheme pairs, with accuracies as high as 98%. It has poor performance (64.0%) only for the pair *argument from consequences* and *argument from verbal classification*; perhaps not coincidentally, these are the two schemes for which performance was poorest in the one-against-others task.

6.2 Impact of type on classification accuracy

As we can see from Table 6, for one-against-others classifications, incorporating **type** into the feature vectors improves classification accuracy in most cases: the only exception is that the best average accuracy of one-against-others classification between *argument from cause to effect* and *others* is obtained without involving **type** into the feature vector — but the difference is negligible, i.e., 0.5 percentage points with respect to the average difference. **Type** also has a relatively small impact on *argument from verbal classification* (2.6 points), compared to its impact on *argument from example* (22.3 points), *practical reasoning* (8.1 points), and *argument from consequences* (7.5 points), in terms of the maximal differences.

Similarly, for pairwise classifications, as shown in Table 7, **type** has significant impact on BAAs, especially on the pairs of *practical reasoning* versus *argument from cause to effect* (17.4 points), *practical reasoning* versus *argument from example* (22.6 points), and *argument from verbal classification* versus *argument from example* (20.2 points), in terms of the maximal differences; but it has a relatively small impact on *argument from consequences* versus *argument from cause to effect* (0.8 point), and *argument from verbal classification* versus *argument from consequences* (1.1 points), in terms of average differences.

7 Future Work

In future work, we will look at automatically classifying **type** (i.e., whether an argument is linked or convergent), as **type** is the only feature directly retrieved from annotations in the training corpus that has a strong impact on improving classification accuracies.

Automatically classifying **type** will not be easy, because sometimes it is subjective to say whether a premise is sufficient by itself to support the conclusion or not, especially when the argument is about

target_scheme	***BAA-t***	***BAA-no t***	***max diff***	***min diff***	***avg diff***
example	90.6	71.6	22.3	10.6	14.7
cause	70.4	70.9	−0.5	−0.6	−0.5
reasoning	90.8	83.2	8.1	7.5	7.7
consequences	62.9	61.9	7.5	−0.6	4.2
classification	63.2	60.7	2.6	0.4	2.0

Table 6: Accuracy (%) with and without **type** in one-against-others classification. *BAA-t* is best average accuracy with **type**, and *BAA-no t* is best average accuracy without **type**. *max diff*, *min diff*, and *avg diff* are maximal, minimal, and average differences between each experimental setup with **type** and without **type** while the remaining conditions are the same.

$scheme_1$	$scheme_2$	***BAA-t***	***BAA-no t***	***max diff***	***min diff***	***avg diff***
cause	example	80.6	69.7	10.9	7.1	8.7
reasoning	example	93.1	73.1	22.8	19.1	20.1
reasoning	cause	94.2	80.5	17.4	8.7	13.9
consequences	example	86.9	76.0	13.8	6.9	10.1
consequences	cause	87.7	86.7	3.8	−1.5	−0.1
consequences	reasoning	97.9	97.9	10.6	0.0	0.8
classification	example	86.0	74.6	20.2	3.7	7.1
classification	cause	85.6	76.8	9.0	3.7	7.1
classification	reasoning	98.3	89.3	8.9	4.2	8.3
classification	consequences	64.0	60.0	6.5	−1.3	1.1

Table 7: Accuracy (%) with and without **type** in pairwise classification. Column headings have the same meanings as in Table 6.

personal opinions or judgments. So for this task, we will initially focus on arguments that are (or at least seem to be) empirical or objective rather than value-based. It will also be non-trivial to determine whether an argument is convergent or linked — whether the premises are independent of one another or not. Cue words and discourse relations between the premises and the conclusion will be one helpful factor; for example, *besides* generally flags an independent premise. And one premise may be regarded as linked to another if either would become an enthymeme if deleted; but determining this in the general case, without circularity, will be difficult.

We will also work on the argument template fitter, which is the final component in our overall framework. The task of the argument template fitter is to map each explicitly stated conclusion and premise into the corresponding position in its scheme template and to extract the information necessary for enthymeme reconstruction. Here we propose a syntax-based approach for this stage, which is similar to tasks in information retrieval. This can be best explained by the argument in Example 1, which uses the particular argumentation scheme *practical reasoning*.

We want to fit the *Premise* and the *Conclusion* of this argument into the *Major premise* and the *Conclusion* slots of the definition of *practical reasoning* (see Table 1), and construct the following conceptual mapping relations:

1. *Survival of the entire world* ⟶ a goal *G*
2. *Adhering to the treaties and covenants aiming for a world free of nuclear arsenals and other conventional and biological weapons of mass destruction* ⟶ action *A*

Thereby we will be able to reconstruct the missing *Minor premise* — the enthymeme in this argument:

> Carrying out *adhering to the treaties and covenants aiming for a world free of nuclear arsenals and other conventional and biological*

weapons of mass destruction is a means of realizing *survival of the entire world*.

8 Conclusion

The argumentation scheme classification system that we have presented in this paper introduces a new task in research on argumentation. To the best of our knowledge, this is the first attempt to classify argumentation schemes.

In our experiments, we have focused on the five most frequently used schemes in Walton's scheme-set, and conducted two kinds of classification: in one-against-others classification, we achieved over 90% best average accuracies for two schemes, with other three schemes in the 60s to 70s; and in pairwise classification, we obtained 80% to 90% best average accuracies for most scheme pairs. The poor performance of our classification system on other experimental setups is partly due to the lack of training examples or to insufficient world knowledge.

Completion of our scheme classification system will be a step towards our ultimate goal of reconstructing the enthymemes in an argument by the procedure depicted in Figure 1. Because of the significance of enthymemes in reasoning and arguing, this is crucial to the goal of understanding arguments. But given the still-premature state of research of argumentation in computational linguistics, there are many practical issues to deal with first, such as the construction of richer training corpora and improvement of the performance of each step in the procedure.

Acknowledgments

This work was financially supported by the Natural Sciences and Engineering Research Council of Canada and by the University of Toronto. We are grateful to Suzanne Stevenson for helpful comments and suggestions.

References

Robin Cohen. 1987. Analyzing the structure of argumentative discourse. *Computational Linguistics*, 13(1–2):11–24.

Judith Dick. 1987. Conceptual retrieval and case law. In *Proceedings, First International Conference on Artificial Intelligence and Law*, pages 106–115, Boston, May.

Judith Dick. 1991a. *A Conceptual, Case-relation Representation of Text for Intelligent Retrieval*. Ph.D. thesis, Faculty of Library and Information Science, University of Toronto, April.

Judith Dick. 1991b. Representation of legal text for conceptual retrieval. In *Proceedings, Third International Conference on Artificial Intelligence and Law*, pages 244–252, Oxford, June.

Vanessa Wei Feng. 2010. Classifying arguments by scheme. Technical report, Department of Computer Science, University of Toronto, November. `http://ftp.cs.toronto.edu/pub/gh/Feng-MSc-2010.pdf`.

Sarah George, Ingrid Zukerman, and Michael Niemann. 2007. Inferences, suppositions and explanatory extensions in argument interpretation. *User Modeling and User-Adapted Interaction*, 17(5):439–474.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 76–83, Morristown, NJ, USA. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explorations Newsletter*, 11(1):10–18.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X)*, pages 19–33.

Joel Katzav and Chris Reed. 2004. On argumentation schemes and the natural classification of arguments. *Argumentation*, 18(2):239–259.

Raquel Mochales and Marie-Francine Moens. 2008. Study on the structure of argumentation in case law. In *Proceedings of the 2008 Conference on Legal Knowledge and Information Systems*, pages 11–20, Amsterdam, The Netherlands. IOS Press.

Raquel Mochales and Marie-Francine Moens. 2009a. Argumentation mining: the detection, classification and structure of arguments in text. In *ICAIL '09: Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 98–107, New York, NY, USA. ACM.

Raquel Mochales and Marie-Francine Moens. 2009b. Automatic argumentation detection and its role in law and the semantic web. In *Proceedings of the 2009 Conference on Law, Ontologies and the Semantic Web*, pages 115–129, Amsterdam, The Netherlands. IOS Press.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection

of arguments in legal texts. In *ICAIL '07: Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pages 225–230, New York, NY, USA. ACM.

John L. Pollock. 1995. *Cognitive Carpentry: A Blueprint for How to Build a Person*. Bradford Books. The MIT Press, May.

J. Ross Quinlan. 1993. C4.5: Programs for machine learning. *Machine Learning*, 16(3):235–240.

Chris Reed and Glenn Rowe. 2004. Araucaria: Software for argument analysis, diagramming and representation. *International Journal of Artificial Intelligence Tools*, 14:961–980.

Glenn Rowe and Chris Reed. 2008. Argument diagramming: The Araucaria project. In *Knowledge Cartography*, pages 163–181. Springer London.

Frans H. van Eemeren and Rob Grootendorst. 1992. *Argumentation, Communication, and Fallacies: A Pragma-Dialectical Perspective*. Routledge.

Douglas Walton and Chris Reed. 2002. Argumentation schemes and defeasible inferences. In *Workshop on Computational Models of Natural Argument, 15th European Conference on Artificial Intelligence*, pages 11–20, Amsterdam, The Netherlands. IOS Press.

Douglas Walton, Chris Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.