Automated Identification of Argument Structure in Natural Language

May 2016

Indian Institute of Technology Kharagpur

for award of the degree of

Bachelor of Technology

by

Arkanath Pathak

under the guidance of

Dr. Pawan Goyal

and

Dr. Plaban Bhowmick

Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

Declaration

This is to certify that

1. The work contained in this thesis is original and has been done by myself under the general supervision of my supervisors.
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have followed the guidelines provided by the Institute in writing the thesis.
4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
6. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them in the text of the thesis and giving their details in the references.

Arkanath Pathak, Department of Computer Science and Engineering
IIT Kharagpur
Date :

Authorization by the Advisor

I have looked at the report and I find it satisfactory.

Pawan Goyal, Assistant Professorm Department of Computer Science and Engineering
IIT Kharagpur
Date :

Contents

1 Introduction

The research field of argumentation mining deals with the identification of argument structures in a text. The argument structure is typically represented as a graph consisting of textual propositions as nodes, and both support and attack relations as edges between the propositions.

In their influential work, Mochales and Moens (2011) define an argument as “discussions in which reasons (premises) are advanced for and against some proposition or proposal (conclusion)”. Needless to say, arguments might not be present in some texts (which we call non-argumentative text). Mochales and Moens have discussed this problem in details with precise definitions, frameworks, and terminologies. They define the argumentation structure as consisting of various “arguments”, forming a tree structure, where each argument consists of a single conclusion and (possibly many) premises. Fig. 1 shows an example structure shown in their work. Our framework is inspired by their work.

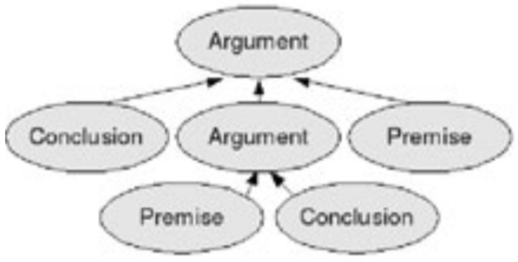

Figure 1: Example argument structure considered by Mochales and Moens (2011).

There are several variations of the frameworks to define an argumentation structure. One of the most widely used of them is the Freeman theory of argumentation structures Freeman (1991; 2011), which treats an argument as propositions being proponent or opponent nodes for a central claim. Fig. 2 shows an example structure shown in their work. However, we refrain from using this framework and modify on Mochales and Moens (2011)’s framework which suits a hierarchical structure better.

There are various possible applications of argumentation mining, mostly in the fields of information processing. Mochales and Moens (2011) have described how we can benefit in information retrieval when argumentation mining is performed on legal documents. Teufel and Moens (2002) explain how document summarization can also benefit from a solution to this problem. Mochales and Moens (2011) discuss the history of the problem, its complications and its applications in details.

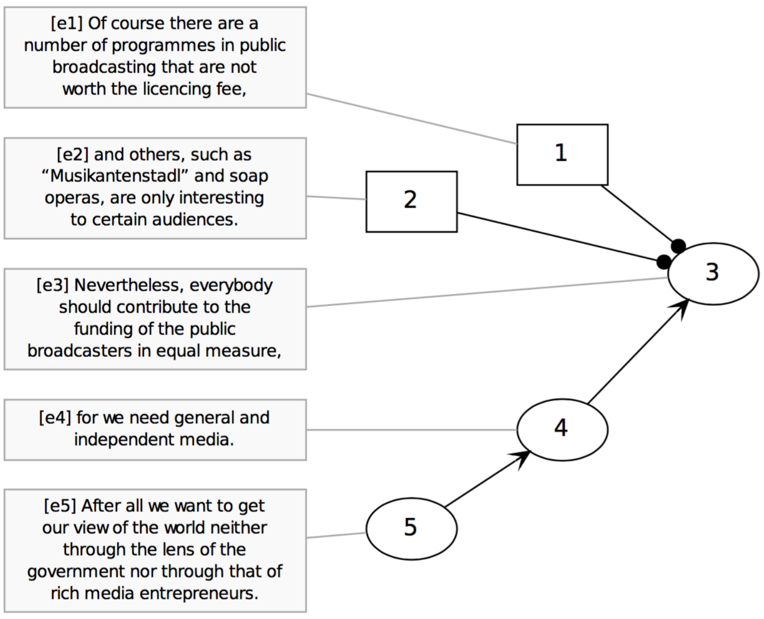
[e1] Of course there are a number of programmes in public broadcasting that are not worth the licencing fee,
[e2] and others, such as "Musikantenstadl" and soap operas, are only interesting to certain audiences.
[e3] Nevertheless, everybody should contribute to the funding of the public broadcasters in equal measure,
[e4] for we need general and independent media.
[e5] After all we want to get our view of the world neither through the lens of the government nor through that of rich media entrepreneurs.
1
2
3
4
5

Figure 2: An example text and its reduced argumentation structure in the Freeman's framework: Text segments, proponent (round) and opponent (box) nodes, supporting (arrowhead) and attacking (circle-head) relations. This image is taken from (Peldszus and Stede, 2015).

The full problem of argumentation mining can be divided into four subtasks:

1. **Segmentation**: Splitting the text into propositions.
2. **Detection**: Identifying the argument propositions.
3. **Classification**: The argumentative propositions are then classified into pre-defined classes (e.g. premise or conclusion in the Mochales and Moens' framework and proponent or opponent in the case of Freeman's framework).
4. **Structure Identification**: Building the structure by identifying the relations (the edges in the argumentative graph structure) between the propositions. We confine our work to support relations only [†].

Our work jointly tackles the Classification and Structure Identification subtasks using a unified approach. Our focus is on the identification of the linguistic properties involved

[†]For the corpus that we are using, there are very few attack relations. Furthermore, support relations are largely due to complex linguistic inferences.

in the formation of an argumentative structure. More formally, we tackle the problem of identification of argument structure given the set of propositions using a machine learning approach.

To our knowledge, there has been very limited work done for structure identification subtask. (Mochales and Moens, 2011) used a hard-coded context-free grammar to predict argumentation trees. They report an accuracy for structure prediction of 60% when run on a corpus for legal documents. However, they have not reported the structure prediction accuracy for the database that we are using. We are familiar with only two approaches which can be compared to our work.

(Lawrence et al., 2014), in their work on 19^{th} Century Philosophical Texts proposed to form bidirectional edges between propositions. They used the euclidean distance metric between topic measures derived from a generated topic model for the text to be studied. Each proposition identified in the test data is then compared to the model, giving a similarity score. They achieved a raw accuracy of 33% for linking the edges. However, their approach is not robust since the choice of threshold parameter is not trivial, and more importantly, they do not form directed edges, which is essential in case of arguments.

(Peldszus and Stede, 2015) jointly predict different aspects of the argument structure. Although their approach is also data-driven, we can not comment on their results because of two reasons. First, they use a database which uses the Freeman theory for argumentation framework. Second, they have not reported the final structure prediction scores, despite stating that one can use MST decoding algorithm to achieve the final tree structure.

We propose a data-driven approach focusing on the identification of the linguistic properties involved in the formation of an argument structure.

2 Dataset Description

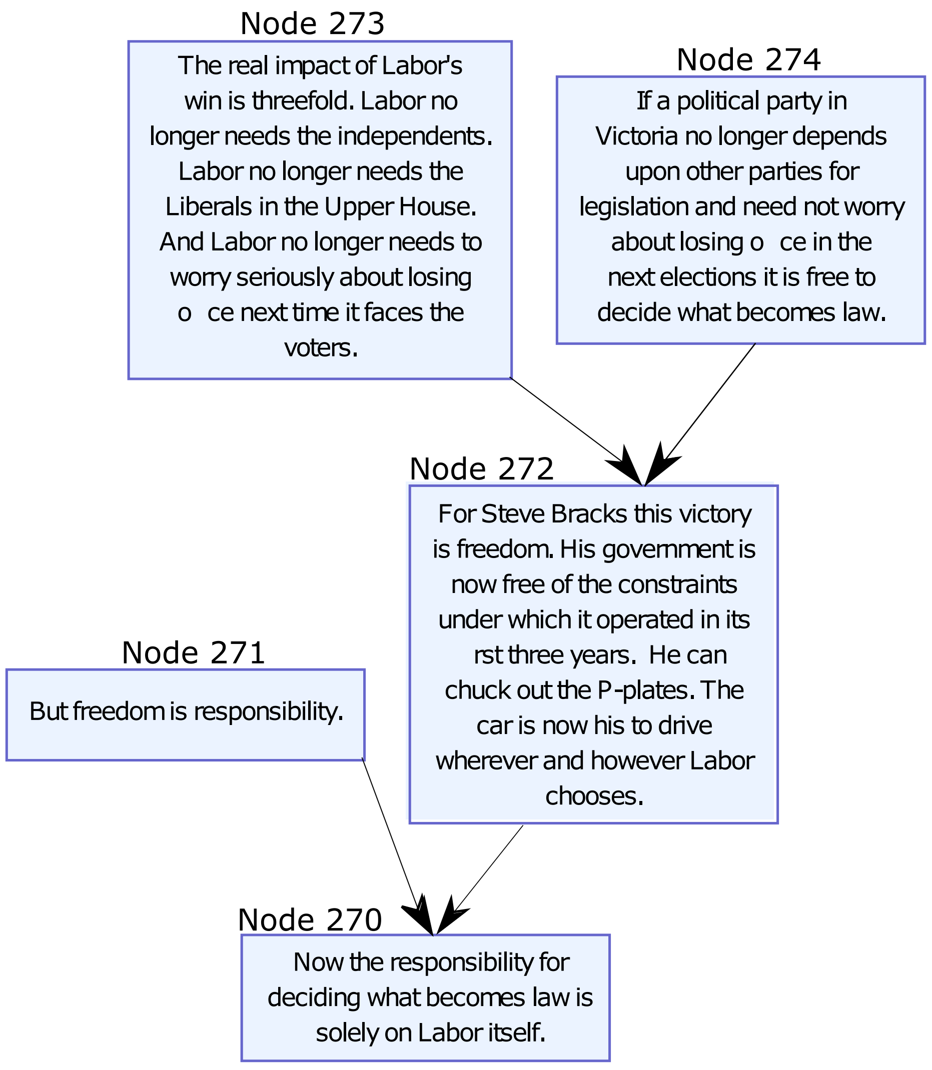
Node 273
The real impact of Labor's win is threefold. Labor no longer needs the independents. Labor no longer needs the Liberals in the Upper House. And Labor no longer needs to worry seriously about losing o ce next time it faces the voters.
Node 274
If a political party in Victoria no longer depends upon other parties for legislation and need not worry about losing o ce in the next elections it is free to decide what becomes law.
Node 272
For Steve Bracks this victory is freedom. His government is now free of the constraints under which it operated in its rst three years. He can chuck out the P-plates. The car is now his to drive wherever and however Labor chooses.
Node 271
But freedom is responsibility.
Node 270
Now the responsibility for deciding what becomes law is solely on Labor itself.

Figure 3: Sample argument (Argument No. 9) from the AraucariaDB.

We use the AraucariaDB (Reed and Rowe, 2004) dataset to discuss the problem at hand. This corpus consists of a structured set in English, collected and analysed as a part of a project at the University of Dundee (UK). The data was collected over a 6 week period in 2003, during which time a weekly regime of data collection scheduled regular harvests of one argument from 19 newspapers (from the UK, US, India, Australia, South Africa, Germany, China, Russia and Israel, in their English editions where appropriate), 4 parliamentary records (in the UK, US and India), 5 court reports (from the UK, US and Canada), 6 magazines (UK, US and India), and 14 further online discussion boards and 'cause' sources such as HUman Rights Watch (HURW) and GlobalWarming.org.

AraucariaDB, in our opinion, is the most suited freely available corpus for argumentation structure prediction, when the focus is on natural language arguments. Most of the other datasets are specific to some objective, e.g. legal documents, debates, etc. . On the other hand, the arguments in AraucariaDB are not related to one another, and they are well-formed tree structured natural language arguments. The database consists of 661 (numbered from 7 to 667) argument maps (or simply, arguments), some of which will be ignored for this paper. AraucariaDB can be downloaded and visualized with the AIFdb Lawrence et al. (2012) framework at (http://www.arg.dundee.ac.uk/aif-corpora/araucaria).

A sample argument is shown in Fig. 3. The RA nodes stand for the relation of inference, a terminology defined in AIFdb which is being used in this diagram. In our work, however, we will be referring to such relations as support relations. A support relation implies a relation of deduction (or inference). AraucariaDB also contains some arguments with attack relations, but we have ignored those arguments. There are 13 such arguments which have been ignored.

As can be observed in Fig. 3, the support relations are very complex in nature. If our goal is to identify this structure, given the set of noes, the edges must have some distinctive features. Let's consider the edge 271→270. We have the common word "resposnsibility" and there are discourse markers like "But". Despite these, the inference is difficult to deduce using an automated approach, since it involves complex linguistic inference as well as some information about the other nodes which take part in this argument. In a similar fashion, one can investigate the other edges for this argument. We will see later in Section 4 that we find scores for each possible edge with the given set of nodes. This approach relies on the assumption that the true edges are superior in some terms than the false edges that can be constructed. Other than natural language inference, there are many other complicated relations in AraucariaDB like providing expert opinion or practical reasoning.

3 Problem Formulation

Fig. 3 shows a argument structure from AraucaraDB. The edges in the tree represent a support relation. For instance, Node 271 and Node 272 are the children of Node 270. Therefore, $271 \rightarrow 270$ as well as $272 \rightarrow 270$ are support relations. For a given support relation $n_1 \rightarrow n_2$, we call the node n_1 as the *Text* node and n_2 as the *Hypothesis* node. We are following this notation since this is widely used in problems like Recognizing Textual Entailment or Recognizing Textual Inference. Our goal can formally be defined as the following:

For a given set of propositions, with an (unknown) underlying argument structure connecting these propositions, identify the argument structure using some metric for scoring the structures.

Our approach, discussed in details in the next section, tries to arrive at the structure by rating the degree of inference for each possible edge between a pair of nodes. This task is similar to Recognizing Textual Interference (RTE) or Recognizing Textual Inference (RTI). However, the state-of-the-art methods for those problems were not giving satisfactory results in the context of argumentation mining. In Section. 5.2, we will formulate a baseline based on the EDITS RTE tool, and we can observe that our approaches perform much better. (Walton, 2007) have discussed the complexities in an argument. Textual inference, for example, relies solely on linguistics and semantics. However, the formulation of argument might include: witness testimony, position to know, expert opinion, popular opinion, example, analogy, practical reasoning (from goal to action), verbal classification, vagueness of a verbal classification (a rebuttal to the previous scheme), sign, popular practice, sunk costs, appearance, ignorance (lack of evidence), cause to effect, correlation to cause, abductive argumentation scheme, consequences, alternatives, threat, fear appeal, pity, commitment, ad hominem argument (various subtypes), inconsistent commitment, bias (a rebuttal), gradualism, slippery slope argument (various types), established rule, exceptional case, precedent (Walton, 2007).

4 Proposed Approach

Assuming we have the set of nodes N for an argument, our approach can be divided into two subtasks:

- **Score Assignment** : Assign scores $s_{n_1,n_2} \in [0,1]$ for every pair of nodes $n_1, n_2 \in N$, $n_1 \neq n_2$. These scores represent the degree of support relation present in the hypothetical edge connecting n_1 and n_2.

- **Tree Prediction** : Choose the tree T^* with the maximum sum of edge scores, i.e. $T^* = \underset{T}{\operatorname{argmax}} \sum_{(n_1,n_2)\in E(T)} s_{n_1,n_2}$

, where T can be any tree with the set of nodes N and $E(T)$ denotes the set of edges in T.

We use the confidence measures provided by machine learning classifiers as the edge scores in task 1. The task is of binary classification with the classes being *Support* and *Neutral*. The classifier takes as input an ordered pair (n_1,n_2), where n_1 and n_2 are the text nodes. We now describe the features which we found most suitable for this problem.[†]:

4.1 Discourse Markers

Discourse markers may be treated as the most direct implication of argumentative direction between two propositions. For example, the presence of the word "*therefore*" clearly implies that the proposition is probably a Hypothesis rather than a Text. Discourse markers have persistently been used for both RTE and NLI tasks before. **?** have also discussed the role of discourse markers in the context of argumentation mining. *However, the presence of such words are rare in the AraucariaDB dataset.* This will be more evident later when report the results of the Ablation study.

We have used the following features as discourse markers:

1. Counts of the following words in the Text
   - *as*, *or*, *and*, *roughly*, *then*, *since*
2. Counts of the following words in the Hypothesis

[†]We also experimented with other kinds of features which are often used in similar problems: POS n-grams, the length of propositions, POS of the main verb, etc. However, these features have been observed to have insignificant effect on overall performance. We suspect this is due to the fact that many attributes of these features are already captured in the features we have chosen.

- *therefore*, *however*, *though*, *but*, *quite*

This feature set gives rise to 11 (6+5) features.

4.2 Modal Features

Modal features are similar to discourse markers but they do not inherently belong to one of Text or Hypothesis. Therefore, we take the counts of these as features for both Text as well as Hypothesis inputs.

We have used the following as the modal words:

- *can*, *could*, *may*, *might*, *must*, *will*, *would*, *should*

Mddal feature set gives rise to 16 (8×2) features.

4.3 Common Wikipedia Entities

In many cases, one can observe that a specific argument usually revolves around some entities. For example, Argument 9 (Fig.3) involves the entities *Steve Bracks*, *Labor*, etc. To account for these, we have used an entity annotation tool called TAGME (Ferragina and Scaiella, 2010). After we have the annotations as a vector for both Text and Hypothesis we take the resulting vector inner product as a single feature.

4.4 Word N-grams

We have used unigrams and bigrams found in the training data as features. For feature selection, we have ordered the n-grams by the relative likelihood of occurrence in Text nodes and of occurrence in Hypothesis nodes. We first the filter the set of n-grams by the count of occurrence(we have used the threshold as 3 throughout our experiments). This set is created for both Text and Hypothesis nodes. We then order the n-grams with decreasing relative likelihood values of $\frac{p(ngram|Text)}{p(ngram|Hypothesis)}$ for Text nodes. This set is again filtered with some threshold value for ratio (we have used the ratio cut-off of 3 throughout our experiments). A similar process is repeated to find the top n-grams for Hypothesis nodes with the likelihood as $\frac{p(ngram|Hypothesis)}{p(ngram|Text)}$. Once we have the set of n-grams (we have used features for unigrams and bigrams), the count of each such n-gram is used as a feature for both Text and Hypothesis nodes. It can be observed that the count of features will depend on the training data. Since we have performed cross validation, these counts will vary. The mean count for unigrams was 115.4 and that for bigrams was 251.8. Hence, the average number of features was 734.4 (115.4×2 + 251.8×2).

4.5 Word Vector Embeddings

Word vectors capture a variety of helpful information in the context of natural language. We have directly used the pre-trained vectors trained on part of Google News dataset (Mikolov et al., 2013) [†]. These vectors have a dimension of 300. We have used the sum of word vectors over words in Text node to form a feature vector. A similar process is repeated for the Hypothesis node so we have another feature vector. These vectors are concatenated to give rise to 600 features for an input pair. Assuming the dimensions are sparse, although the process of summing is expected to induce noise, however it should retain some information. On the other hand, it's difficult as well as inefficient to consider the word vectors for each word as a separate feature. Using word vectors as features can help with various attributes inherent in support edges. First, a simple similarity measure can be the difference of the two sum vectors which can be well captured by using classifiers like linear SVM. Second, word vectors trained over an external dataset like Google News can provide the knowledge base for language not present in training set, which is very likely in case of argumentation mining since the arguments are expected not to be related. And finally, since word vectors are based on contextual information it can infer support relation from similar contexts present in training data.

Word vectors therefore give rise to 600 (300×2) features.

4.6 Longest Common Phrase

We have used a single feature which captures the maximum number of contiguous words which are common to both Text and Hypothesis nodes.

4.7 Algorithm For Finding Best Structure

Once we have the scores for all pairs of nodes, the process of finding the best tree is straightforward. To iterate on the structures we follow the a simple recursive algorithm described in Algorithm 1. Note that we need to call the algorithm for each possible root node and choose the one with the best score. The algorithm can be modified as per purpose to also return various other attributes of the tree like height or set of edges.

[†]The pre-trained word vectors for Google News dataset is freely available at https://code.google.com/archive/p/word2vec/

```
Data: Set $LastLevel$ of nodes, Set of remaining nodes $N$, Dictionary $E$ mapping
      edges to scores
Result: The sum of edge scores of the best tree structure
If $N$ is empty, $return$ 0;
Initialize $MaxScore$=0;
for every possible non-empty subset of $N$ as $NewLevel$ do
    Recursively call the algorithm with $NewLevel$, $N - NewLevel$, $E$ as inputs,
    store the returned value as $Score$;
    for every node $n$ in $NewLevel$ do
        Find the node $n_{parent}$ in $LastLevel$ with the maximum score $E[(n, n_{parent})]$;
        $Score+ = E[(n, n_{parent})]$;
    end
    if $Score > MaxScore$ then
        $MaxScore = Score$;
    end
end
$return\ MaxScore$;
```

Algorithm 1: Algorithm for finding the best structure. The algorithm needs to be called for each possible root node.

5 Experiments

All of the experiments in this section are performed on the AraucariaDB arguments. The experiments are performed in a 5-fold cross-validation framework and the mean scores are reported. The folds are over the set of arguments rather than the pairs of text nodes, to maintain contextual independence between the folds. It should be noted that arguments containing relations other than support are ignored.

Measure	**type-1 SVM**	**type-2 SVM**	**type-2 MLP**
$confidence_S$	0.691	0.643	0.678
$confidence_N$	0.425	0.356	0.306
$recall_S$	0.759	0.677	0.677
$recall_N$	0.532	0.681	0.692
$precision_S$	0.193	0.68	0.688
$precision_N$	0.937	0.678	0.681
$accuracy$	0.561	0.679	0.684

Table 1: Classifier Performance: the mean values are reported for both support (S) and neutral (N) relations.

5.1 Classifier Performance

The support edges present in the input argument structures are directly taken as support pair examples for the classifier. However, the neutral pairs generation is not so straightforward. To do so, we have experimented with two kinds of frameworks.

The first framework, which we call the *type-1* framework, considers all the pairs (n_1,n_2) as a neutral such that n_1 and n_2 are text nodes belonging to the same argument and $n_1 \rightarrow n_2$ is not a support relation. This, however, gives rise to a huge imbalance between the support and neutral examples. Many classifiers fail to perform well in such imbalance. This issue can be resolved in classifiers like linear SVM, however, by assigning class weights inversely proportional to class frequencies (King and Zeng, 2001) in the input data. We have followed this framework for type-1 SVM. Another way to resolve the problem of imbalance is to down-sample the neutral relations randomly. However, random sampling did not perform well in our experiments, and this makes us believe that just a random subset might not be a good training sample.

To counter this, we devised the *type-2* framework which considers only those pairs (n_1,n_2) as neutral such that n_1 and n_2 are text nodes belonging to the same argument, and $n_2 \rightarrow n_1$

is a support relation. This gives a perfectly balanced input dataset with one neutral example corresponding to each support example, making it suitable for most machine learning classifiers.

A Multi-layer Perceptron (MLP) classifier using the type-2 framework performed better than type-1 and type-2 SVM implementations for arguments with 3 nodes in our experiments. The network has 3 hidden layers with 200 neurons each.

Table 1 summarizes the results for the three classifiers. We have shown 7 performance measures for the each classifier. Specifically, we present the mean of the confidence values provided by the classifier for each class label, which is used directly in the next stage. We have scaled the confidence measure[†] linearly to between 0 and 1 before using it as scores for structure prediction. A mean confidence of 1, therefore, will be the perfect score for the Support pairs. Similarly, a mean confidence of 0 will be the perfect score for the Neutral pairs. We can observe that each classifier outperforms the others for atleast some metric.

One can observe that type-2 classifiers perform better in predicting neutral pairs. The confidence measure is lower than type-1 and the recall is higher as well. However, the precision for neutral is better for type-1 SVM because of the data imbalance. Type-2 MLP gave the best accuracy in our experiments. Deciding on the best classification framework is difficult since each performs better in some aspect. However, considering all aspects, type-1 SVM can be chosen to be the ideal candidate for general experiments.

5.2 Structure Prediction Performance

Since the arguments are complex in nature, our approach (Section 4) fails to predict the entire structure very often. To counter this, we formulate a measure to evaluate the similarity between the predicted tree and input tree. The measure, $SimScore$, is defined as:

$$SimScore(T_1, T_2) = \frac{|E(T_1) \cap E(T_2)|}{|E(T_1)|}$$

where T_1 and T_2 have the same set of nodes and $E(T)$ is the set of edges for a tree T.

Since our problem formulation is new, it is difficult to compare the results with existing experiments for structure prediction in argumentation mining. However, we compare our

[†]We have used the SVM and Multi-layer Perceptron classifier implementations provided by the open source library scikit-learn (Pedregosa et al., 2011). For the confidence measure, we have used the decision function in the case of SVM and the predicted probability in the case of MLP. Class imbalance was handled using ‘balanced’ weighting of classes.

Nodes	**Arguments**	$SimScore$				
		type-1 SVM	**type-2 SVM**	**type-2 MLP**	**EDITS**	**Random**
2	10	**0.9**	0.8	0.8	0.7	0.5
3	187	0.564	0.566	**0.625**	0.363	0.333
4	85	0.529	**0.552**	0.482	0.250	0.250
5	62	**0.446**	0.399	0.435	0.231	0.2
6	72	**0.363**	0.341	0.322	0.263	0.166
7	58	**0.369**	0.323	0.309	0.231	0.142
8	41	**0.230**	0.19	0.199	0.205	0.125
9	19	**0.351**	0.28	0.222	0.265	0.111
10	23	**0.217**	0.188	0.115	0.212	0.1
Any	557	**0.459**	0.442	0.447	0.289	0.234

Table 2: Structure Prediction Performance: Mean of $SimScore$ for the arguments grouped by the number of nodes in the argument.EDITS and Random are baselines whereas type-1 SVM, type-2 SVM and type-2 MLP are proposed approaches.

performance with two baselines.

The first baseline, **Random**, is a random baseline which gives equal weighting to every tree structure with the given set of nodes. It can be shown that the expected value of the $SimScore(T_i, T)$ for a given tree T, is equal to $1/n$ where n is the number of nodes in T.

For the second baseline, **EDITS**, we use the state-of-the-art software package EDITS (Kouylekov and Negri, 2010) for recognizing Textual Entailment, for scoring the edges instead of the classifiers we proposed. However, the metric for scoring structures remains the same as the sum of edge scores. In this case, the entailment relation corresponds to the support relation. EDITS has also been used previously by (Cabrio and Villata, 2012) in the context of argumentation mining.

In Table 2, we compare the mean value of $SimScore$ for each classifier. We have further categorized the results for arguments with the number of nodes. It is ideal that the performance will degrade as the number of nodes increase. We can observe that all the three classifiers outperform the EDITS and Random baselines with a large margin. For the arguments with 3 nodes, type-2 MLP outperforms type-1 SVM and type-2 SVM. However, for arguments with higher number of nodes, type-1 SVM performs the best.

Feature Set	**$\%$ decrease in** $SimScore$
Discourse Markers	0.09%
Modal Features	0.27%
Wikipedia Similarity	0.59%
Word N-grams	1.56%
Word Vectors	**11.4%**
Longest Common Phrase	1.02%

Table 3: Ablation study: The decrease in structure prediction performance due to the removal of each kind of feature.

5.3 Ablation Study

To test the efficacy of each individual feature/feature group, we have performed a leave out ablation test. In Table 3, we report the $\%$ decrease in the mean structure prediction similarity score (for any number of nodes) when the type-1 SVM classifier is used.

Interestingly, Discourse Markers were the least helpful features in our experiments. We think this is due to two reasons. First, discourse markers are fairly rare in AraucariaDB. Second, we suspect Word Vectors and Word N-grams capture the discourse markers to a certain extent. We can observe that the Word Vectors feature group is clearly the major contributor to structure prediction performance. It hence makes sense to deduct that word vectors are capturing much more information that n-grams and other linguistic features in the context of arguments.

5.4 Error Analysis

To provide a sense of where our approach can fail, we provide some examples where our classifiers fail to predict the correct relation. For Argument 9 (Fig.3), let's consider the edge $271 \rightarrow 270$. The type-1 SVM classifier gave a score of 0.687 for $(271, 270)$, however it gave a score of 0.859 for $(270, 271)$. A similar situation is observed for Argument 41 (Fig.4). In this case the classifier was able to predict $501 \rightarrow 499$, but it chose $(499, 500)$ (score of 0.858) over $(500, 499)$ (score of 0.801). Here again, one can see that knowing about node 501 one might infer that it is more likely that 500 supports 499. Therefore, determination of a support relation, in many cases, is influenced by the other support relations around the same claim. Apart from that, resolution of support relations require inferences drawn on world knowledge. It can be observed that to eliminate these errors, one must require the knowledge of the domain. That is, the other nodes of arguments might help identifying the relationship.

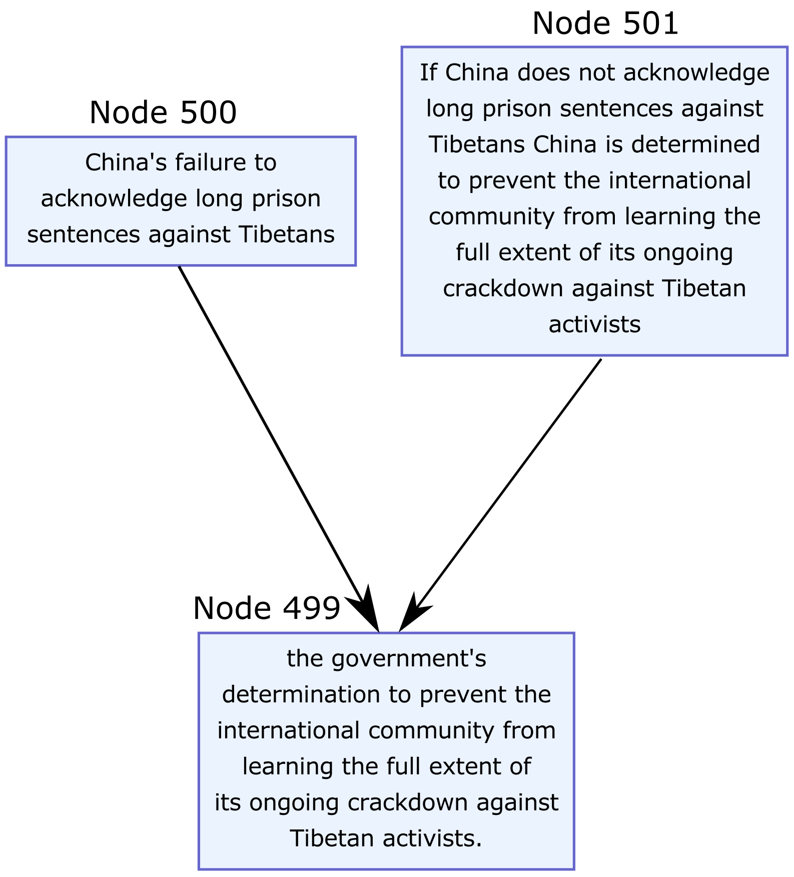
Node 501
If China does not acknowledge long prison sentences against Tibetans China is determined to prevent the international community from learning the full extent of its ongoing crackdown against Tibetan activists
Node 500
China's failure to acknowledge long prison sentences against Tibetans
Node 499
the government's determination to prevent the international community from learning the full extent of its ongoing crackdown against Tibetan activists.

Figure 4: Argument 41 from the AraucariaDB.

6 Extension Of The Approach To Arguments With Attack Relations

Till now, we have only considered arguments without attack relations. A natural extension of this approach is for the arguments which include both support and attack relations. We were unable to find datasets like Araucaria which include attack relations in a proportionate quantity. Before discussing the possible approaches, we first describe two datasets that we found the most suitable for this context. Note that these datasets don't inherently follow the framework we have been using. These datasets are freely available at *http://www-sop.inria.fr/NoDE/NoDE-xml.html*.

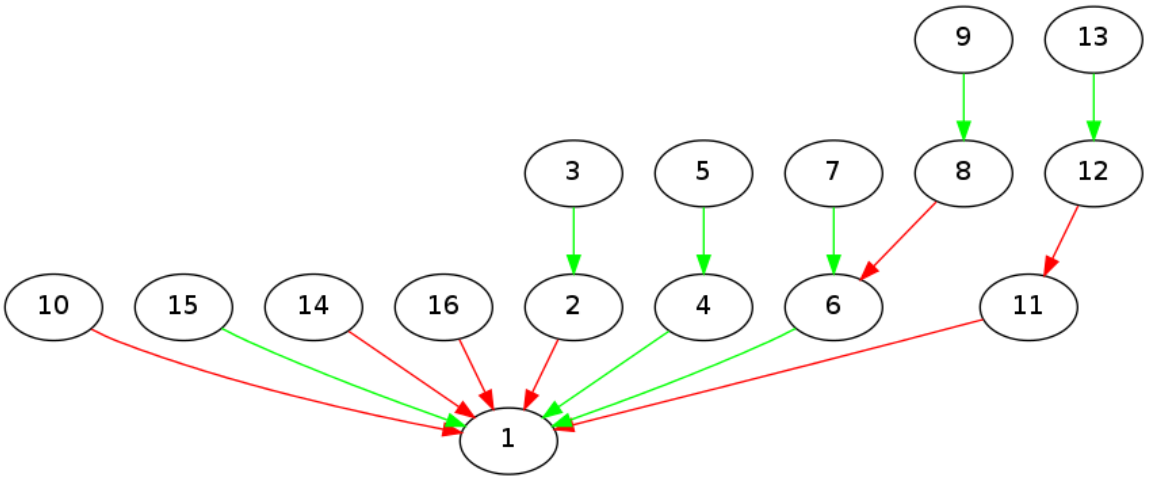

Figure 5: An example debate structure from Debatepedia for the topic of Violent Games. Green edges indicate support relation whereas red edges indicate attack relation.

6.1 Debatepedia+ProCon datasets

These two datasets, Debatepedia and ProCon, are two websites for debates on trending and crucial issues. These two datasets are commonly used for RTE tasks. To fill in the first layer of the dataset, there is a set of topics of Debatepedia/ProCon debates, and for each topic the following procedure is applied (Cabrio and Villata, 2013):

1. The main issue (i.e., the title of the debate in its affirmative form) is considered as the starting argument.

2. Each user opinion is extracted and considered as an argument.

3. Since attack and support are binary relations, the arguments are coupled with the starting argument, or other arguments in the same discussion to which the most recent argument refers.

4. The resulting pairs of arguments are then tagged with the appropriate relation, i.e., attack or support. Fig. 5 shows an example debate from the dataset for the topic of "Violent Games". Note that the structures need not necessarily be a tree.

The datasets, combined, include 260 pairs (140 supports, 120 attacks).

6.2 Wikipedia dataset

This dataset of natural language arguments is built on two dumps of the English Wikipedia. It focuses on the five most revised pages at that time (i.e. George W. Bush, United States,

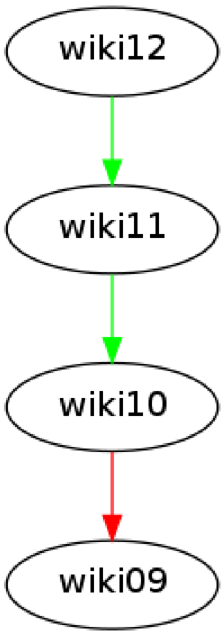

Figure 6: An example argument from Wikipedia dataset.

Michael Jackson, Britney Spears, and World War II). And then follow their yearly evolution up to 2012, considering how they have been revised in the next Wikipedia versions. After extracting plain text from the above mentioned pages, each document has been sentence-splitted, and the sentences of the two versions have been automatically aligned to create pairs. Then, to measure the similarity between the sentences in each pair, the Position Independent Word Error Rate (PER), i.e. a metric based on the calculation of the number of words which differ between a pair of sentences is adopted. For each pair of extracted sentences, TE pairs are created setting the revised sentence as Text and the original sentence as Hypothesis.

Wikipedia dataset consists of 452 pairs (215 supports, 237 attacks).

6.3 Possible Approaches

To extend the approach described in Section. 4, there are a few complications that we need to resolve. First, the structures are not tree-like. For DebatePedia, fig. 7 shows an example debate on the topic of "Ground zero mosque". For wikipedia, all the arguments are basically linear graphs, as shown in Fig. 6. Second, to extend our approach for decoding the best possible tree, we need to modify the algorithm. The classifiers are no longer binary.

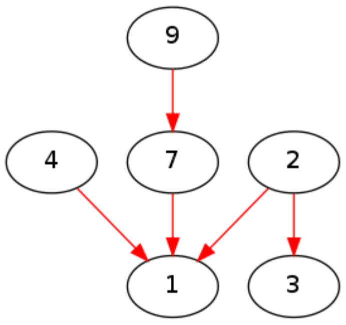

Figure 7: An example DebatePedia debate which doesn't follow a tree-like structure.

There are now three possible relations between an unordered pair of nodes (a,b):

- Support
- Attack
- Neutral

This complicates the matter in a lot of ways. In the earlier case, distinction between Support and Neutral also included the effect of features which are not related to inference. For example, similarity of concepts. However, those features are now common to both Support and Attack relations.

To account for these complications, there are two possible approaches:

1. **Two-Step Approach**: In the first step a classifier marks the pairs which have either Support or Attack edge. In the second step, an independent classifier resolves those edges into Support or Attack.
2. **Single-Step Approach**: A multi-class classifier resolves all the three relations: Support, Attack or Neutral.

6.4 Building Classifiers for Attack as well as Support relations

We performed experiments on building classifiers for distinguishing between support and attack relations (Second step of the Two-Step Approach described earlier). Two types of features were included in addition to the features described in Section 4:

1. **Negation Discourse Markers**: A new set (26 words) of discourse markers were included. These markers try to capture contrast or negation sentiments in a sentence. Examples of markers include: *can't*, *never*, etc.

2. **Negation/Contrast Relation Indicators (Harabagiu et al., 2006)**: Features in this category intend to capture contrast or negation relations present between an ordered pair of sentences. We have followed the approaches provided in the work by Harabagiu et al. An example feature takes the count of words which appear in contrasting contexts in the two sentences. For example, if the first sentence includes the phrase "Peter's eating the chocolate" whereas the second sentence includes the phrase "Peter is not eating the chocolate". Here, eating appears in a negated context in the second phrase whereas non-negated context in the first phrase.

We achieved the following accuracies for the datasets using an SVM classifier:

- **Wikipedia**: 0.645 ± 0.11
- **DebatePedia+ProCon**: 0.665 ± 0.08

The experiments for structure prediction in the presence of attack relations are left for future work because of the complications mentioned in Section 6.3.

7 Conclusion

In this report, we have proposed a two-stage approach towards identification of argument structure in natural language text. To do so, we used machine learning classifiers to predict which edges might be involved in the argument structure by retrieving a confidence score for each possible edge. We also provided two frameworks for training on data that will be encountered for such formulation. With the help of Ablation study, we showed that word vectors trained on an external corpus can be a crucial feature for such tasks. For the overall goal of structure prediction, our proposed approach predicted almost twice as many correct edges than with the random baseline. Through error analysis, the knowledge of other nodes in the argument has been conjectured to be important in link prediction. This is something which has been left for future exploration. We also explored the complications which arise when we extend this approach to arguments with both support and attack relations.

8 Bibliography

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

E. Cabrio and S. Villata. A natural language bipolar argumentation approach to support users in online debate interactions†. *Argument & Computation*, 4(3):209–230, 2013.

P. Ferragina and U. Scaiella. Fast and accurate annotation of short texts with wikipedia pages. *arXiv preprint arXiv:1006.3498*, 2010.

J. B. Freeman. *Dialectics and the macrostructure of arguments: A theory of argument structure*, volume 10. Walter de Gruyter, 1991.

J. B. Freeman. *Argument Structure:: Representation and Theory*, volume 18. Springer Science & Business Media, 2011.

S. Harabagiu, A. Hickl, and F. Lacatusu. Negation, contrast and contradiction in text processing. In *AAAI*, volume 6, pages 755–762, 2006.

G. King and L. Zeng. Logistic regression in rare events data. *Political analysis*, 9(2): 137–163, 2001.

M. Kouylekov and M. Negri. An open-source package for recognizing textual entailment. In *Proceedings of the ACL 2010 System Demonstrations*, pages 42–47. Association for Computational Linguistics, 2010.

J. Lawrence, F. Bex, C. Reed, and M. Snaith. Aifdb: Infrastructure for the argument web. In *COMMA*, pages 515–516, 2012.

J. Lawrence, C. Reed, C. Allen, S. McAlister, A. Ravenscroft, and D. Bourget. Mining arguments from 19th century philosophical texts using topic based modelling. In *Proceedings of the First Workshop on Argumentation Mining*, pages 79–87. Citeseer, 2014.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

R. Mochales and M.-F. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19 (1):1–22, 2011.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

A. Peldszus and M. Stede. Joint prediction in mst-style discourse parsing for argumentation mining. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pages 938–948, 2015.

C. Reed and G. Rowe. Araucaria: Software for argument analysis, diagramming and representation. *International Journal on Artificial Intelligence Tools*, 13(04):961–979, 2004.

S. Teufel and M. Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445, 2002.

D. Walton. Visualization tools, argumentation schemes and expert opinion evidence in law. *Law, Probability and Risk*, 6(1-4):119–140, 2007.