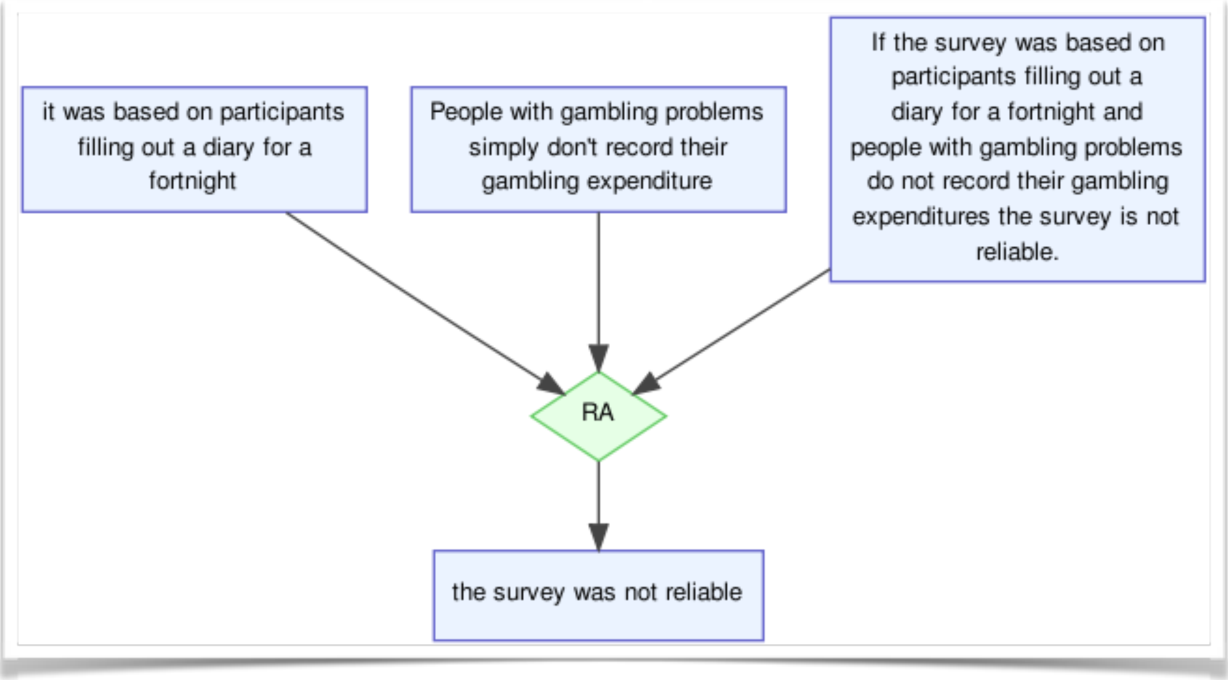
Automated Identification of Argument Structure in Natural Text

B.Tech. Project Report, Autumn 2015


it was based on participants filling out a diary for a fortnight
People with gambling problems simply don't record their gambling expenditure
If the survey was based on participants filling out a diary for a fortnight and people with gambling problems do not record their gambling expenditures the survey is not reliable.
RA
the survey was not reliable

Arkanath Pathak

Roll No. 12CS10007

4th Year B.Tech. Student

Dept. of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Project Advisors:

Dr. Pawan Goyal

Assistant Professor
Dept. of Computer Science and Engineering, IIT Kharagpur

Dr. Plaban Bhowmick

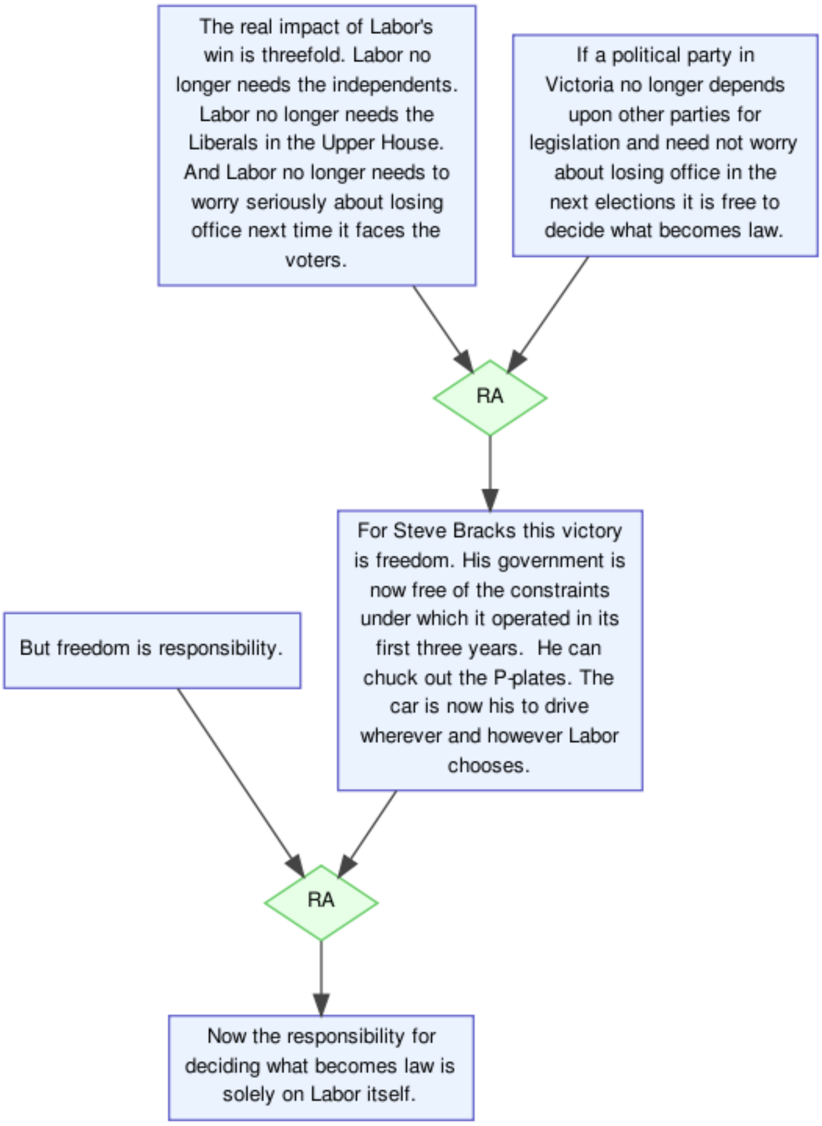
Assistant Professor
Center for Educational Technology, IIT Kharagpur

Table of Contents

Introduction

Argumentation mining is a relatively new challenge in corpus-based discourse analysis with the ultimate goal of identifying argumentative structures within a document. The various subtasks involved in the field are identification of the premises, conclusion, and argumentation scheme of each argument, as well as argument - sub-argument (the hierarchy involved, thus identifying what we are calling the argumentative structure) and argument - counter-argument relationships between pairs of arguments in the document.

For an example, let’s consider the argument structure taken from the Araucaria DB [1] in the following figure:



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 office next time it faces the voters.

If a political party in Victoria no longer depends upon other parties for legislation and need not worry about losing office in the next elections it is free to decide what becomes law.

For Steve Bracks this victory is freedom. His government is now free of the constraints under which it operated in its first three years. He can chuck out the P-plates. The car is now his to drive wherever and however Labor chooses.

But freedom is responsibility.

Now the responsibility for deciding what becomes law is solely on Labor itself.

Fig.1: Structure of Argument 9, Araucaria DB

The RA nodes stand for the relation of inference, a terminology introduced by the Argument Interchange Format (AIF) 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), which in our case is very complex since it deals with natural language. There have been more than one proposed representations of arguments that researchers have used in the past. Two notable ones which are most commonly used are Freeman's theory of argumentation structures [2] and the more generic premise-conclusion structures, a sample of which is described by Mochales,Moens in [3]. We have assumed the more generic premise-conclusion structure without any constraints on the number of conclusions, as opposed to that in [3] which imposes a limit of a single constraint.

A premise-conclusion structure assumes an argument scheme to be composed of a conclusion being supported by a number of premises. Thus all these premises form a support relation with the conclusion. Some other representations also include the attack relations (conflict relations), but we have ignored attack relations at present because of the lack of training examples in the dataset we have used. The overall structure of the argument can be assumed to be a hierarchical structure composed of sub-arguments supporting a conclusion to form a bigger argument, as shown in Fig. 1. To provide the required hierarchical structure, we have assumed the argument structure to be a directed tree. This assumption usually holds for all the data sets and is generally a valid assumption capable of supporting complex structures. In fact, this assumption is supported by around 95% of the argument analyses contained in AIFdb [4].

Returning to the main task of Argumentation Mining, we have to construct this structure given the initial block of text. Hence we will have to segment the sentences, identify the premises and conclusions and derive the edges. Some surface level complexities that arise for these tasks are:

1. How to identify the nodes: Involves finding text boundaries, and problems like whether to keep a single sentence or to keep multiple sentences which together form a concrete sense. In Fig. 1 for example, we have nodes that comprise of multiple sentences. This will be the case for the dataset we will be working on throughout this report.
2. How to identify the support relations. The problem of finding natural language inference has been discussed in the past but the results are very poor in case the logic involved is dependent on meaning rather than some semantic words. In the case of natural language arguments, the deductions are even more complex involving relations like providing expert opinion or practical reasoning. Most existing approaches use the RTE (Recognizing Textual Entailment) task

classifiers built for this purpose. In our approach, most of the existing approaches have failed to be satisfactory, in fact, our present investigation is based on devising effective features which can help increase the accuracy. We will discuss about this point in more detail in the section where we discuss our ongoing investigation.

In general, there are 3 major steps to be performed towards this goal.

1. Detection of argumentative propositions.
2. Classification of propositions into premises and conclusions.
3. Identifying the structure of argument by adding edges between the propositions.

Our goal merges the work to be done in steps 2 and 3. In the next section we describe our goal in more detail.

One can refer to the work done by Mochales and Moens [3] to get a more detailed introduction to these 3 steps and Argumentation Mining in general.

Our Goal

We describe our problem formulation as follows:

Given a set of argumentative propositions (unstructured, english), find the structure of the argument by joining all the propositions to form a directed tree.

Hence, we are given the nodes of the graph and we have to construct the edges. This problem statement already encompasses both the steps 2 and 3 described in the previous section. However, we are not doing the step 2 separately, rather we are forming edges directly. The main conclusion can be treated as the root of the tree. There will be a set of premises for the main conclusion which can themselves be conclusions for a deeper level of premises. Hence this forms a hierarchical structure.

Review of Literature

There have been some successful approaches towards the steps 1 and 2, i.e. detection and classification. Moens et al., 2011 [3] reported an accuracy of 73-80% for the task of detection of argumentative phrases and an accuracy of 68-74% for the task of classification. However, most of the approaches for step 3 are in some sense supervised, e.g. [3] discusses a method using a hand-written context-free grammar (CFG) for detection of argumentation scheme. There have been very few approaches towards automated (unsupervised) identification of the structure, most of them being in the last few years.

Cabrio et al. [5], in 2012, in their work on online debates discussed one of the earliest approaches of automated identification by using textual entailment as the first stage of joining propositions and then using argumentation theory to reject invalid arguments.

Lawrence et al. [4], in 2014, in their work on 19th Century Philosophical Texts proposed to form bidirectional edges between propositions based on euclidean distance between topic measures by a generating a topic model for the text to be studied and then each proposition identified in the test data is compared to the model, giving a similarity score for each topic. The achieved a raw accuracy of **33%** for linking the edges. However, their approach is also not very robust since they have to adjust a threshold value to make edges and then use some workarounds to get to the tree structure. Also, they don't form directed edges, which is required in case of arguments.

More recently, this year, Peldszus et al. [6] published a paper quite close to our approach, but still having some crucial inherent differences. They have used a dataset based on the Freeman's theory. They first perform the task of attachment classification, finding if there is an argumentative attachment or not. We shall be able to observe later, that the way the solution for this would work is much different than the way the solution to our inherent problem (i.e. detection of the premise to conclusion edges) would work. The latter, in our opinion, being a much harder task. Then they assume there is a central claim to which each proposition would either support or attack. We are not comparing our approach with their approach more in this report since the work done by us till now, and the problem we are stuck at, is not relevant for their work. We explain our approach now in the next section.

Approach

Our overall approach, as planned yet, can be described as a 3 step process:

1. Find the edge weights for each possible ordered pair of nodes.
2. Construct the tree structure using the edge weights found in Step 1.
3. Find out the accuracy for our approach using some scoring model.

We briefly describe each of the steps in the following subsections.

Finding the edge weights

The edge weights account for the degree of support between an ordered pair of propositions. For example, the edge weight between a pair of nodes might just be a number between 0 to 1 representing the degree of support. This is certainly the most difficult step out of the three steps, and can be treated as a bottleneck for our approach. Our whole approach assumes that using the edge weights should be enough to infer the structure. In a way this assumption is valid since we are not taking into account the previous context encountered. However, looking at the results in the present state of investigation we might have to use the context for a feature in the future as well.

Natively, researchers have used RTE tools (e.g. [5]) to get the edge weights in the past. In the last semester, we began with the same approach but the results were not satisfactory. We realized that most of the RTE tools use only syntactic and few basic semantic measures to build a classifier. But in our case, the database is filled with natural logic. So we read about the works done in Natural Language Inference [7], from where we learnt that most successful approaches train a ML based classifier. So we decided to train our own classifier using the features relevant in our scenario, hoping to get better results.

To sum up, our goal in this step is to try out different settings of classifiers and features, and come up with the optimal.

Constructing the structure

Once we have the edge weights, inferring the tree structure is not that easy either. But that is a step for which we have not yet got the opportunity to explore. A possible approach that came to our mind is to use some simple MST decoding algorithm. In the baseline model that we have implemented we simply choose the structure with the highest linear sum of the edges. But there must exist possible improvements to this model.

Deciding on a scoring model

Once we have the structure, we will have to decide on some measure to find the similarity with the actual structure in the training data. A possible approach that came to our mind is to use graph edit distance.

Note that at the present stage we might even have to defer from our goals. Hence it is not wise to dive deep into the steps 2 and 3 at this stage. A successful approach to finding the edge weights itself would be a noticeable work in our opinion.

Experimental Settings & Results

Classifier Used

We have used an SVM with an rbf kernel (radial basis function, a.k.a. Gaussian kernel) for the task of binary classification. The classes would correspond to support and neutral relations respectively.

The classifier would take as input an ordered pair of texts **t,h** (text and hypothesis, we use this notation for the purpose of classifier since it is more widely used for the problem of inference). The edge weights are to be derived from the confidence scores for the classes (which comes from the distance of the point to the separating hyperplane) given by the SVM.

About Araucaria DB

We use the Araucaria DB provided by AIFdb (http://www.arg.dundee.ac.uk/aif-corpora/). This corpus consists of a structured set in English collected and analysed according to a specific methodology 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.

The database consists of 661 argument maps, few of which have the conflict relations that we are ignoring at present. A sample argument was shown in Fig. 1.

For the input to the classifier, we take all the support relation pairs as the training data corresponding to the support labels. Whereas we take all possible ordered pairs which don't have a support edge as a valid neutral pair. This is not a very fair assumption but this is the only possible approach to derive neutral pairs from the Araucaria DB. SVM is highly sensitive to unbalanced data, favoring the major label in that case. Hence, in our case, we had to down-sample the neutral pairs in a random manner so that the number of neutral pairs is in the same range as of the number of support pairs.

Note that we will have edges in both directions for an unordered pair of nodes, we will have to choose which way gives the better score.

Features for Classifier

We chose the following features in the initial experiment:

1. Discourse Markers (e.g. “if”, “therefore”)
2. Modal Features (e.g. “would”, “could”)
3. Common Wikipedia Entities between the text and hypothesis, provided by the TagMe API (http://tagme.di.unipi.it/) [10]
4. Count of all possible word bi-grams from the train set
5. Count of all possible POS bi-grams from the train set
6. Avg. vector over the words in text and hypothesis, found by using the Google News trained word vectors (https://code.google.com/p/word2vec/) [8]

Most of these are usually the features chosen by the existing textual entailment or inference solvers, except the Wikipedia and Word2vec that we decided to introduce ourselves.

For feature selection, we used the ANOVA F-value metric for the provided sample to choose the top k features. This is similar to the chi-square metric which is commonly used.

Results

Train Data Statistics

Argument Maps: 400

Support Pairs: 2128

Neutral Pairs: 14154 used a 0.15 random sampler for each train run to have a balanced train data.

Test Data Statistics

Argument Maps: 193

Support Pairs: 989

Neutral Pairs: 6428

Classifier Results

Support Pairs Labeling: 523 labeled as Support, 466 labeled as Neutral.

Neutral Pairs Labeling: 3957 labeled as Neutral, 2511 labeled as Support.

Thus leading to the following scores:

Accuracy: 0.604

Macro-averaged precision: 0.53

Macro-averaged recall: 0.565

There is no existing results to compare with because all the previous approaches have been different in some way. Either they find the undirected edges or they find out the accuracy after they have the premise-conclusion classification done.

However, we decided to choose the EDITS RTE tool, which decides whether a pair is Entailment or Non-Entailment. The entailment in our case shall correspond to the support relation. This tool was used by Cabrio et al. [5]. We use this to arrive at some baseline model to compare with. Here are the results using the EDITS tool for detecting Entailment or Non-Entailment:

Support Pairs Labeling: 192 labeled as Support, 768 labeled as Neutral.
Neutral Pairs Labeling: 6623 labeled as Neutral, 626 labeled as Support.

What is to be observed is that the detection of the support relations fail miserably. The recall for support pairs in this case would be 0.16 as compare to 0.52 achieved by us. This is because EDITS is based on syntactic features which are not so useful in the case of Araucaria DB. For these reasons we were bound to build our own classifier. Our classifier also maintains to get a recall of 0.61 for the neutral pairs hence it is blindly favoring a class.

Present Investigation

After realizing the results, we felt the need to improve it the results by introducing more features. We tried to debug the false classifications made by the SVM manually. What we observed was that it was in fact because of the poorly chosen features. The complexity of the logic inherent in the arguments can not be captured well by the features that we had chosen.

So we decided to dive deeper into the given dataset to identify the features causing the argumentation. In the present section we describe some of our findings yet. Though many arguments involved what we identified as “Complex Logic” which can not be captured by numerical features so easily, we still identified some features that can help us improve the accuracy, namely:

1. Tense of the verbs involved: active or passive.
2. Relative length of the text between argument nodes and conclusions.
3. POS of the word which followed the verb, e.g. “We should” should usually occur in an affirmative sentence, giving rise to a possible conclusion.
4. Identifying some effective bigrams and unigrams may be more effective than using all possible bigrams/

5. Containment criteria of conclusion in premise (might involve context): Many arguments commonly contained an extra premise which joins the existing premises at the level to the conclusion. We call this a containment relation. This effect is illustrated by the example in the Fig. 2 (Argument 17, Araucaria DB).
6. Finding paraphrase similarities.
7. More similarity measures between text and hypothesis (other than the common Wikipedia Entities measure).
8. Any inference possible by using LIWC categories.
9. Any inference possible by using discourse parsers.

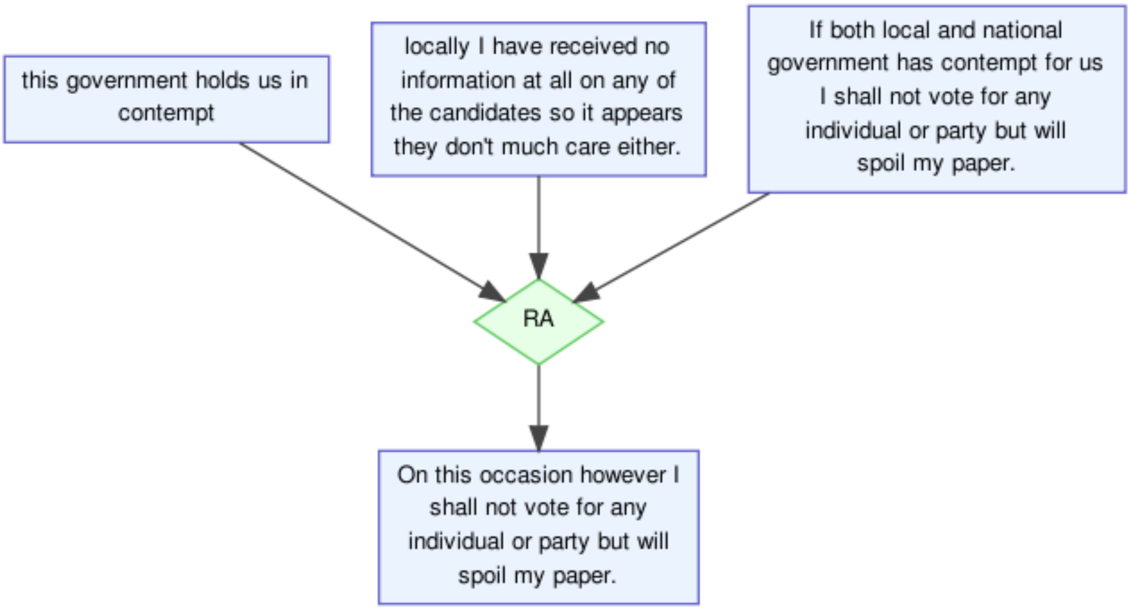

this government holds us in contempt
locally I have received no information at all on any of the candidates so it appears they don't much care either.
If both local and national government has contempt for us I shall not vote for any individual or party but will spoil my paper.
RA
On this occasion however I shall not vote for any individual or party but will spoil my paper.

Fig.2: Structure of Argument 17, Araucaria DB

While we have not yet explored all of these factors, we report the findings of some of the factors in the following paragraphs.

For the length measure, we found out that while could not observe a very natural consistent heuristic, we can infer that the max length nodes in an argument is more probable to be text rather than hypothesis, the distribution is shown in Fig. 3.

For the tense of the main verb, we have tried out an approach of using the POS tag of the root node in the dependence parse structure. However, it did not give an interesting difference between the distribution for the text nodes and the hypothesis nodes.

For the text nodes, these were the top 10 POS tags:
["NN", 293], ["VBD", 181], ["NNS", 91], ["JJ", 83], ["VBZ", 63], ["VBG", 60], ["VBP", 30], ["RB", 23], ["NNP", 23], ["IN", 22]

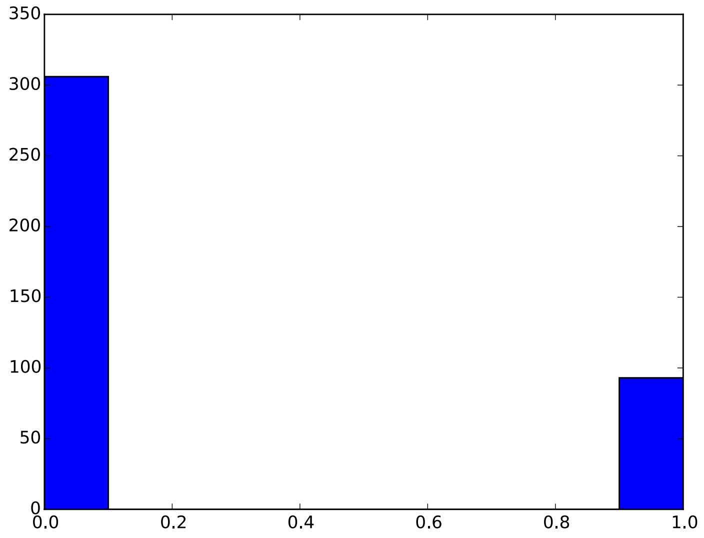


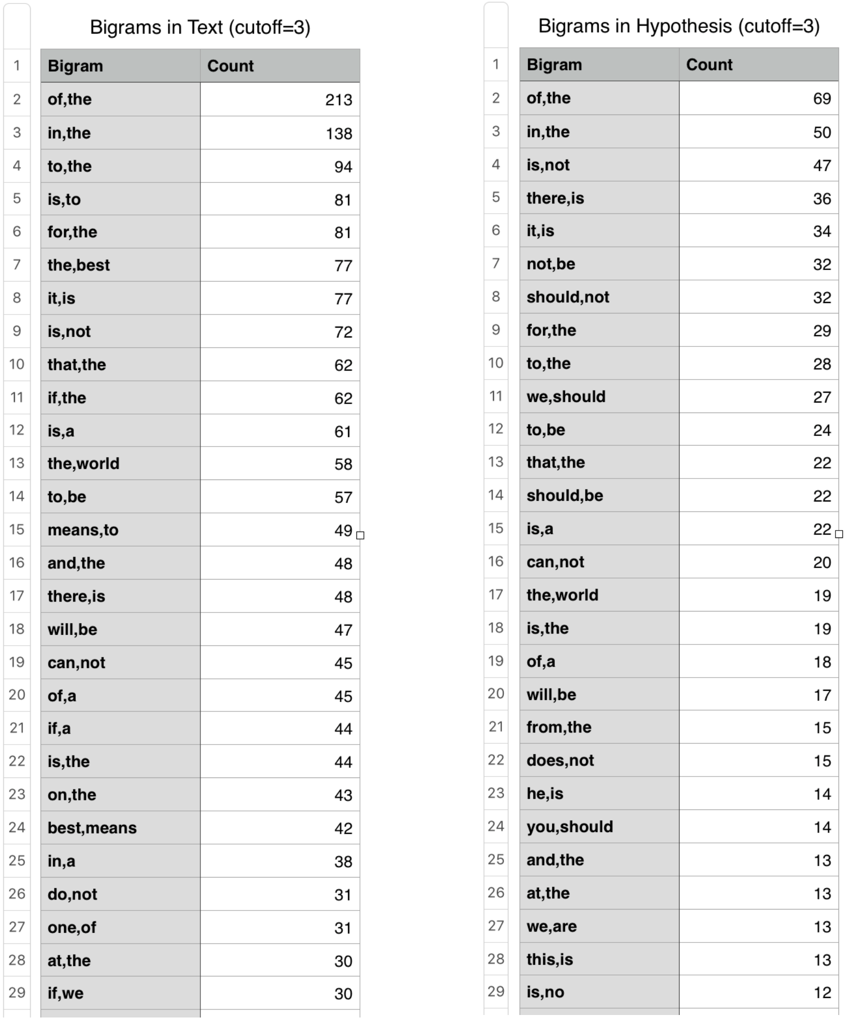
Fig.3: Distribution of roles for max length nodes in arguments

For the hypothesis nodes, these were the top 10 POS tags:
["NN", 175], ["VBD", 93], ["JJ", 60], ["VBG", 26], ["VBZ", 26], ["NNS", 20], ["VBP", 15], ["NNP", 10], ["VB", 9], ["RB", 6]

For the unigram and bigram statistics, we did find some interesting results and we are hoping it would help us improve the accuracy. We show the bigram results comparison for text and hypothesis in Fig. 4.

One can observe from the figure that the distribution is quite different for the top bigrams (the bigrams shown in Fig. 4) in the two cases. As an example, for the case of the bigram we discussed before, **"we should".** The probability of the bigram belonging in the hypothesis set is **7.56** times more than the probability of the bigram belonging in the text set. On the other hand, the bigram "**if the**" is **2.45** times more probable to be in the text set than the hypothesis set. These results are considering the bigrams which have occurred more than 3 times.

Using these information we can find out the top bigrams and unigrams to be used for features.



Bigrams in Text (cutoff=3)

Bigram	Count
of,the	213
in,the	138
to,the	94
is,to	81
for,the	81
the,best	77
it,is	77
is,not	72
that,the	62
if,the	62
is,a	61
the,world	58
to,be	57
means,to	49
and,the	48
there,is	48
will,be	47
can,not	45
of,a	45
if,a	44
is,the	44
on,the	43
best,means	42
in,a	38
do,not	31
one,of	31
at,the	30
if,we	30

Bigrams in Hypothesis (cutoff=3)

Bigram	Count
of,the	69
in,the	50
is,not	47
there,is	36
it,is	34
not,be	32
should,not	32
for,the	29
to,the	28
we,should	27
to,be	24
that,the	22
should,be	22
is,a	22
can,not	20
the,world	19
is,the	19
of,a	18
will,be	17
from,the	15
does,not	15
he,is	14
you,should	14
and,the	13
at,the	13
we,are	13
this,is	13
is,no	12

Fig.4: Counts for top 29 bigrams for text and hypothesis nodes

Conclusion & Future Work

Our approach is novel and we hope to arrive at some interesting results. We hope to implement these complex features into the classifier and also play with different classifiers available. The work done for Natural Language Inference by Bowman et al. [9] uses a neural network model centered around a Long Short-Term Memory network to achieve the state of the art efficiency. We might as well explore that approach. Once we have the classifier ready, we will move on to steps 2 and 3 as described in the approach.

References

1. Reed, Chris, and Glenn Rowe. "Araucaria: Software for argument analysis, diagramming and representation." International Journal on Artificial Intelligence Tools 13.04 (2004): 961-979.
2. Freeman, James B. “Argument Structure:: Representation and Theory”. Vol. 18. Springer Science & Business Media, 2011.
3. Mochales, Raquel, and Marie-Francine Moens. "Argumentation mining." Artificial Intelligence and Law 19.1 (2011): 1-22.
4. Lawrence, John, et al. "Mining arguments from 19th century philosophical texts using topic based modelling." ACL 2014 (2014): 79.
5. Cabrio, Elena, and Serena Villata. "Combining textual entailment and argumentation theory for supporting online debates interactions." Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012.
6. Peldszus, Andreas, and Manfred Stede. "Joint prediction in MST-style discourse parsing for argumentation mining." Proc. of the Conference on Empirical Methods in Natural Language Processing. 2015.
7. MacCartney, Bill. “Natural language inference”. Diss. Stanford University, 2009.
8. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.
9. Bowman, Samuel R., et al. "A large annotated corpus for learning natural language inference." arXiv preprint arXiv:1508.05326 (2015).
10. Ferragina, Paolo, and Ugo Scaiella. "Fast and accurate annotation of short texts with Wikipedia pages." arXiv preprint arXiv:1006.3498 (2010).
11. Wyner, Adam, et al. “Approaches to text mining arguments from legal cases.” Springer Berlin Heidelberg, 2010.
12. Somasundaran, Swapna, and Janyce Wiebe. "Recognizing stances in ideological on-line debates." Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Association for Computational Linguistics, 2010.
13. Feng, Vanessa Wei, and Graeme Hirst. "Classifying arguments by scheme." Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011.
14. Bach, Ngo Xuan, et al. "A two-phase framework for learning logical structures of paragraphs in legal articles." ACM Transactions on Asian Language Information Processing (TALIP) 12.1 (2013): 3.

15. Allen, Kelsey, Giuseppe Carenini, and Raymond T. Ng. "Detecting Disagreement in Conversations using Pseudo-Monologic Rhetorical Structure.”
16. Cayrol, Claudette, and Marie-Christine Lagasquie-Schiex. "On the acceptability of arguments in bipolar argumentation frameworks." Symbolic and quantitative approaches to reasoning with uncertainty. Springer Berlin Heidelberg, 2005. 378-389.
17. Cabrio, Elena, and Serena Villata. "Towards a Benchmark of Natural Language Arguments." arXiv preprint arXiv:1405.0941 (2014).
18. van Benthem, Johan. “Argumentation in artificial intelligence”. Eds. Iyad Rahwan, and Guillermo R. Simari. Vol. 47. Dordrecht/New York: Springer, 2009.
19. Moens, Marie-Francine, et al. "Automatic detection of arguments in legal texts." Proceedings of the 11th international conference on Artificial intelligence and law. ACM, 2007.
20. Peldszus, Andreas. "Towards segment-based recognition of argumentation structure in short texts." ACL 2014 (2014): 88.
21. Bench-Capon, Trevor JM, and Paul E. Dunne. "Argumentation in artificial intelligence." Artificial intelligence 171.10-15 (2007): 619-641.
22. Aharoni, Ehud, et al. "A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics." ACL 2014 (2014): 64.
23. Boltuzic, Filip, and Jan Šnajder. "Back up your stance: Recognizing arguments in online discussions." Proceedings of the First Workshop on Argumentation Mining. 2014.
24. Schneider, Jodi. "Automated argumentation mining to the rescue? Envisioning argumentation and decision-making support for debates in open online collaboration communities." Proceedings of the First Workshop on Argumentation Mining. 2014.
25. Ghosh, Debanjan, et al. "Analyzing argumentative discourse units in online interactions." Proceedings of the First Workshop on Argumentation Mining. 2014.
26. Ong, Nathan, Diane Litman, and Alexandra Brusilovsky. "Ontology-based argument mining and automatic essay scoring." ACL 2014 (2014): 24.
27. Rahwan, Iyad. "Mass argumentation and the semantic web." Web Semantics: Science, Services and Agents on the World Wide Web 6.1 (2008): 29-37.
28. Budzynska, Katarzyna, et al. "Towards Argument Mining from Dialogue." Computational Models of Argument: Proceedings of COMMA 2014 266 (2014): 185.
29. Eckle-Kohler, Judith, Roland Kluge, and Iryna Gurevych. "On the Role of Discourse Markers for Discriminating Claims and Premises in Argumentative Discourse.”
30. REED, Mathilde JANIER John LAWRENCE Chris. "OVA+: an Argument Analysis Interface." Computational Models of Argument: Proceedings of COMMA 2014 266 (2014): 463.
31. Bex, Floris, et al. "On logical specifications of the Argument Interchange Format." Journal of Logic and Computation (2012): exs033.