Argument Mining in Wikipedia Articles

Ben Rothschild
bnroths@uchicago.edu

Julia Zhou
juliazhou@uchicago.edu

Abstract

1 Introduction

Argumentation mining aims to identify structured argument data from unstructured text. Applications of argumentation mining are vast and include improving information retrieval from long texts, summarizing arguments in legal texts, scientific writing, and news articles or accessing a student's command of subject knowledge in essay assignments. Recently argumentation mining has also been used to develop technologies that can assist humans and dialog systems to debate and reason.

The task of identifying arguments is difficult though because of the various forms arguments can take in relation to a document. Part of argumentation mining is stance identification which is determining if an argument is Pro vs. Con the given topic. This is a difficult NLP task because the subtle nature language can play in taking a stance that often relies on context and word choices or stance development within an argument.

Current approaches to argument mining are engineered to address specific domains, for example, a specific model might be built just to analyze claims in court documents using attributes specific to court documents and legal vocabulary. However, we believe that argumentative sentences are often characterized by common rhetorical structures, independently of the domain and we propose to explore a method that exploits structured parsing information to detect claims without resorting to topic-specific information.

An example of argumentation mining would be taking a topic like the sale of violent video games harms minors and a Wikipedia article about the Video game content rating system and identifying if a specific sentence or entire text of the article has a Pro or Con stance towards the topic. The article includes the sentence Exposure to violent video games causes at least a temporary increase in aggression and this exposure correlates with aggression in the real world which should be labeled as a PRO stance.

In this paper, we break this task into two main tasks in this problem defined as:

1. Identifying claims in a text
2. Identifying the stance of a claim on a topic

By combining these tasks we will be able to tell what sentences in a long text support and oppose a topic.

2 Related Work

(Bar-Haim et al., 2017) LIT REVIEW ANYTHING? (good examples in Roy Bar-Haim paper

3 Data

The dataset we are using is the Claim Stance Dataset from IBM Debater project which can be accessed here http://www.research.ibm.com/haifa/dept/vst/debating_data.shtml. It contains 2,394 labeled claims for 55 topics that are pulled from 1,065 wikipedia articles. For each article we are given the following data points:

- Full text from Wikipedia (tex)
- Topic Target (text)
- Claim Text (text)
- Claim Start Index (integer)
- Claim End Index (integer)

- Stance (Pro or Con)

The dataset was created by first looking at the list of controversial issues https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues Wikipedia identifies to find a subset of 56 topics where there was a clear two-sided debate. From these topics 546 articles were identified that night have claims in them. Annotators then read these articles and identified claims and stances within the articles. An example observation in the dataset would be as follows:
Full Text 44,000 word plain text wikipedia article
Topic Target the sale of violent video games to minors
Claim Text they increase the violent tendencies among youth
Claim Start Index 8119
Claim End Index 8167
Stance PRO
Each of the 55 topics was annotated and claims were labeled independently by five annotators. Some claims were thrown out because of annotator disagreement. In the final dataset, 98.5% of the claims had agreement on the claim boundaries and stance. A full description of the data annotation and collection process can be found here (Toledo-Ronen et al., 2016)

4 Method Overview

We divided this research problem into two parts, claim identification and stance identification.

4.1 Claim Identification

4.2 Claim Stance Classification

Stance classification focuses on the problem of determining if a claim is PRO or CON, a classification stack given by the following function:

$$Stance(c, t) = f(claim, topic) \qquad (1)$$

The first hypothesis we have is the following: Vectors representing the claim and the target were similar then the stance would be PRO and if they very different than the stance would be CON.

To test this hypothesis, we created word embeddings using Word2Vec and the *gensim* Python package trained on Wikipedia corpus given in the dataset. From these embeddings, we created a mean embedding vector which was the mean vector of the list of vectors for the words that represent the sentence. We trained the model using Logistic Regression and Random Forests. The similarity between the topic vector and claim vector was measured using cosine similarity.

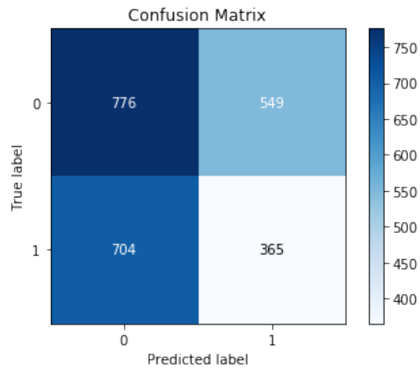

Figure 1: Confusion Matrix for Logistic Model

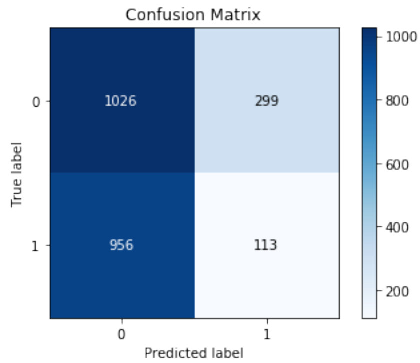

Figure 2: Confusion Matrix for Random Forest

Below are the accuracy and loss of the model trained and reported using stratified cross-validation with three folds. StratifiedKFold

	Log Loss	Accuracy
Logistic	1.15	47.6
Random Forests	.809	47.5

Considering there are only two classes, this accuracy is very bad, and we aren't able to determine better than chance at the stance of a specific claim. The code for this section is 4_stance_detection.ipynb

The next hypothesis we had is that instead of containing similar words or sentence embeddings, claims and targets can be classified by using their sentiment. The hypothesis would be as follows: Claim that have a similar sentiment as a topic will be PRO and divergent sentiment will be CON. This idea is summarized in the following table:

	$Claim_+$	$Claim_-$
$Topic_+$	PRO	CON
$Topic_-$	CON	PRO

To quickly test this hypothesis we will use a pre-trained sentiment analysis model $Vader$. We find the positive, negative and neutral sentiment for each sentence and use these to form a vector which calculates the similarity using cosine similarity. The confusion matrix is below:

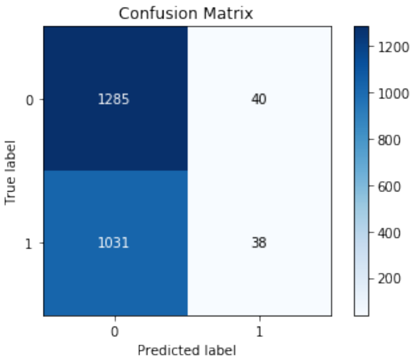

Figure 3: Confusion Matrix for Logistic Model

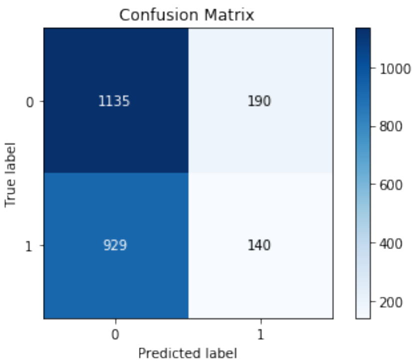

Figure 4: Confusion Matrix for Random Forest

Loss and accuracy are improved from before:

	Log Loss	**Accuracy**
Logistic	.687	55.2
Random Forests	.699	53.2

The code for this section is 4_stance_detection_sentiment.ipynb

From our progress so far we see that this method of comparing sentiment between claim and topic is promising. We think that further analysis of sentiment matching would be helpful. To extend this idea, we think it would be useful to determine the sentiment toward the specific topic instead of the entire sentence. One possible way of doing this would be to create a dependency tree of the claim and to extract the words in that part of the tree. While we didn't have time to prove this theory out we think it would be a valuable area to research further. However, even with a simple model of matching sentiments, we can achieve an accuracy of 55.2% which is better than random but slightly below the results from (Bar-Haim et al., 2017) which had 64.5% accuracy.

5 Conclusion

References

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 251–261.

Orith Toledo-Ronen, Roy Bar-Haim, and Noam Slonim. 2016. Expert stance graphs for computational argumentation. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 119–123.