Argument Mining in Wikipedia Articles

Ben Rothschild
bnroths@uchicago.edu

Jingyuan Zhou
juliazhou@uchicago.edu

Abstract

Claim stance classification has become an active field of research in the past years. We show that claim identification could be further improved with features including mean embedding representation of a sentence, the similarity between sentence and topic target, the location of the sentence, length of sentence and PageRank score of the sentence. We also achieve a better than average accuracy instance classification using sentiment analysis between claim and topic. This paper also shows how the combination of many features types and optimization methods are necessary to achieve accuracy in this NLP task.

1 Introduction

Argumentation mining aims to identify structured argument data from unstructured text. Argumentation is a useful task for applications such as retrieving information from long texts, summarizing arguments in legal documents, scientific writing, and news articles or accessing a student's command of subject knowledge in essay assignments.

Current approaches to argument mining are engineered to address specific domains, for example, a particular model might be built to analyze claims in court documents or legislative records using attributes and vocabularies particular to these domains. However, we believe that argumentative sentences are often characterized by common rhetorical structures, independently of the area, 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 long text support and oppose a topic.

2 Related Work

According to Walton (Walton, 2009), there are four tasks undertaken by argumentation mining: identification, analysis, evaluation, and invention each requiring its specific approach. In this paper, we will focus on identification which we break into two parts: claim identification and stance identification. Recently there have been more attempts to improve argumentation mining benchmarks due to the proliferation of dialog systems and deep learning.

While previous researcher mainly focused on context-dependent claim identification where models are trained using context-specific features (such as speaker identification), new benchmarks are trying to improve context-independent claim identification. The first paper to publish results for context-independent claim identification using the same dataset we did was Lippi and Torroni (Lippi and Torroni, 2015).

Stance identification is also a part of claim identification and state-of-the-art models also use context-dependent features.(Levy et al., 2014) Previous work on stance classification was done using debating forms, congressional floor debates, public comments on proposed regulations among other sources. Often these datasets focused on one subject or topic and utilized classifies with unigram or n-gram features. Additional features are often added depending on datasets such as adding the speaker's identity or political party. Typically these models can get an accuracy rate of 50%-65% depending on their domain.(Bar-Haim et al., 2017)

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 (text)
- 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 might 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

Claim identification is to classify whether a sentence is a claim or not.

4.1.1 Feature Engineering

Features we used for this task include:

1. Mean word embeddings of the sentence

   We created word embeddings using Word2Vec and the *gensim* Python package trained on Wikipedia corpus given in the data set. 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.

2. Similarity between sentence and topic target of the article

   For this feature, we first create mean embedding vector of the topic target and then calculate the cosine similarity between this topic target vector and the sentence vector. The hypothesis behind this idea is that claims have more similar meaning to the topic target than the other sentences.

3. length of the sentence

   The hypothesis behind this feature is that claims may be relatively shorter than the other sentences.

4. location of the sentence in the article

   We calculate the normalized location of a sentence by:

$$loc = \frac{index + 1}{sentencelength} \tag{1}$$

5. Textrank weight of the sentence

   Textrank is the PageRank algorithm applied on the text network. To be more specific, we first calculate the tf-idf values of each word in the sentences of an article. Given the tf-idf matrix, M, we estimate the similarity matrix with $M * M^T$. Thus, this similarity matrix of text represents the same structure of a network graph. We then apply the $pagerank$ function in $networkx$ python package to the matrix to obtain a score for each sentence. Just like the original PageRank algorithm, this score indicates the importance of each sentence. The higher the score is, the more critical the sentence is in the article.

   The hypothesis is that claims tend to be the most important sentences in the articles, so the importance of the sentences could have predictive power for how likely it's going to be a claim.

4.1.2 Upsampling

After we obtained these features, we quickly noticed that our data set is severely imbalanced.

	Count
Label = 1 (claim)	2223
Label = 0 (not claim)	88273

To balance the dataset, we randomly upsampled the minority class, label=1, with replacement to 3500 observations, and also randomly sampled 3500 observations from the majority class, label=0.

4.1.3 Models and Experiments

We experimented with a simple fully connected feed forward neural network model and an LSTM model. For each model, we use grid search to find the optimized combination of the number of hidden layers, layer width and activation function based on model accuracy on the test set.

The optimized simple neural network model is with one hidden layer of width 512 and sigmoid activation function.

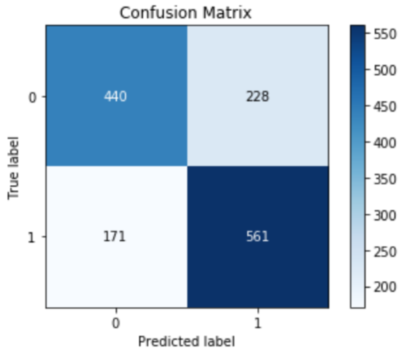

Figure 1: Confusion Matrix for Simple NN

The optimized LSTM model is with one hidden layer of width 512 and tanh activation function.

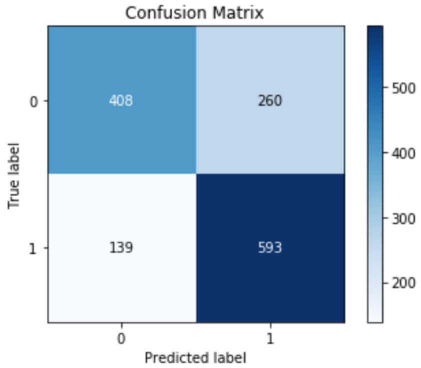

Figure 2: Confusion Matrix for LSTM

Performance of the models is summarized below:

	Log Loss	**Accuracy**
Simple NN	0.571	71.50
LSTM	0.577	71.49

It turns out that both models achieve almost the same highest accuracy for this task.

4.2 Claim Stance Classification

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

$$Stance(claim, topic) = PRO, CON \qquad (2)$$

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 position would be CON.

To test this hypothesis, we used the mean embedding vector of each sentence as described in the previous section. We trained the model using Logistic Regression and Random Forests. The similarity between the topic vector and claim vector was measured using cosine similarity.

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 terrible, 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:

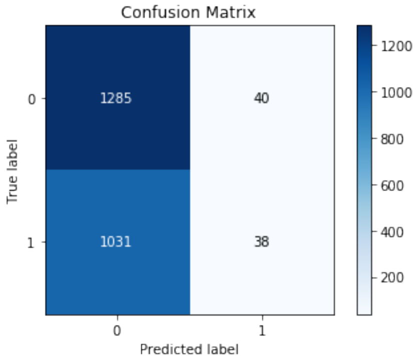

Figure 3: Confusion Matrix for Logistic Model

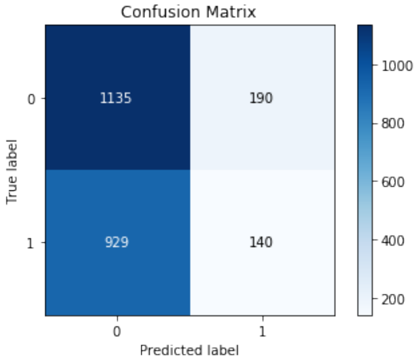

Figure 4: Confusion Matrix for Random Forest

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 5_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

In this paper, we've shown that claim identification, and claim stance classification could benefit from a combination of many features types and optimization methods. A remaining challenge is to build models that are less dependent on training data type and achieve similar results on a variety of data sets including online forum discussions and newspaper articles.

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.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500.

Marco Lippi and Paolo Torroni. 2015. Context-independent claim detection for argument mining. In *IJCAI*, volume 15, pages 185–191.

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.

Douglas Walton. 2009. Argumentation theory: A very short introduction. In *Argumentation in artificial intelligence*, pages 1–22. Springer.