Argument mining of online discussions using word embeddings and Neural Networks

Paul Ozkohen S2575973
Thijmen Kupers S2251418
Demos Ioannou S3839656

October 2018

1 Introduction

1.1 Problem

Debating and argumentation have been a big part of human society. With the rise of the internet, numerous amounts of internet discussion forums have appeared, where people can freely discuss various subjects. One such discussion forum is Reddit.

The forum Reddit is a big influence on the internet with 1.6 billion user accounts as of july 2018. Within this community there is a subreddit called r/changemyview which currently has 597k followers. This subreddit has the purpose for the community to change a persons view on a particular topic through arguments. A user can create a discussion by opening a thread with a particular view about a particular subject. The Original Poster (OP) then starts off the discussion with his arguments that support his view. It is up to other participating posters to post counter-arguments against this view, in the hopes of convincing the OP that his view is wrong or that it should be changed. If the OP finds the arguments from a poster convincing enough such that his view on the topic is changed in some way, he will award a Delta towards that user. A Delta is a virtual point that gets added to a user's account. A user's total delta count for the entire subreddit is viewable and users will want to gather as much Delta points as possible so they can be seen as good debaters.

The subreddit is an interesting place where formal language and natural language seem to collide. Rules and regulations are posed, but the text is mostly written on small mobile screens with two thumbs. This reddit subforum therefore gives some insight to how people currently use arguments on topics on a social media platform. This is different from formal language in essays. It can greatly expand the knowledge of current argumentation that is used in everyday life. Automatically extracting arguments from these discussions will lead to insights about the nature of convincing arguments and how people on the internet can be influenced in their opinions on all kinds of topics.

1.2 Argument Mining

Argument Mining is the term used in this paper, but it has many different names, such as, argumentation mining, computational argumentation or debating technologies, and might have a slightly different meaning in different contexts. However, the definition we use is, it is the automatic identification and extraction of argument data from large resources of natural language texts. Argument mining is a natural continuation and evolution of sentiment analysis and opinion mining: two areas of text mining which became very successful and important both academically and commercially. In sentiment analysis, the work focuses on extracting peoples attitudes (positive, neutral or negative) towards persons, events or products. In opinion mining, the work aims to mine peoples opinions about persons, events or products. Argument mining, on the other hand, allows for recognizing not only what attitudes and opinions people hold, but also **why** they hold them.[1, 2]

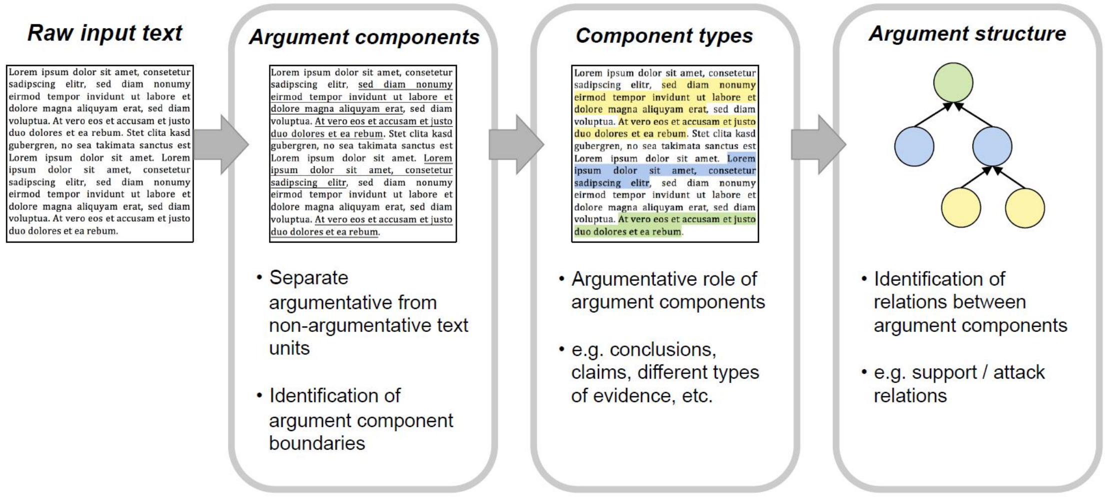

Figure 1: The processing pipeline for the typical argument mining task.

The first step of an argumentation mining pipeline typically focuses on the identification of argumentative text units before analyzing the components or the structure of arguments. This task is usually considered as a binary classification task that labels a given text unit as argumentative or non-argumentative. Although the separation of argumentative from non-argumentative text units is an important step in argumentation mining, it merely enables the detection of text units relevant for argumentation and does not reveal the argumentative role of argument components. The second step of an argumentation mining pipeline focuses on the classification of argument components, aimed at identifying the argumentative role (e.g. claims and premises) of argument components. The third step of an argumentation mining pipeline focuses on understanding the argumentative relations and building the argumentative structure. [1] A graphic representation of this pipeline is shown in Figure 1. In this research, we will focus mostly on the first step in the argumentation pipeline: detection of argumentative text units.

To automatically identify argument components from raw text, we apply machine learning techniques. Specifically, we will use a recurrent Neural Network architecture called a Long-Short Term Memory (LSTM) network. Such Neural Networks can be applied for sequence classification, as they can remember the context when classifying a new input, based on what it has seen before.[5] This makes it a useful model for language tasks. The problem with such supervised learning methods is that large training data has to be provided. Luckily, a few online corpora are available. Such a corpus consists of sentences and is manually constructed by experts who annotate these sentences for specific features. Such features can include whether a sentence is a good argumentative sentence with respect to a specific topic, or how good the quality of the argumentative sentence is, As the manual annotation is a highly time-consuming task, sharing and re-using analyzed data becomes very valuable.T he process of annotation starts with segmenting (splitting) the text into elementary discourse units (EDUs) or in fact into argumentative discourse units (ADUs). Annotators use software tools which help them to assign labels from the annotation scheme set to ADUs directly in a code. Next, the annotated data have to be stored as a corpus.[1, 2]

1.3 State of the art

This research is based on the work described in [5]. Here, LSTM networks are applied for argument sentence detection. The Networks are trained on discriminating argumentative sentences from non-argumentative sentences on specific topics. Then, they are tested in a cross-topic setting where discrimination has to be done without having been trained on these topics. The results show that their approach leads to good performance and shows that cross-topic argument mining is possible using just Neural Networks and barely any manual feature engineering. We adopt one of their four models in our research. More-over, the authors from [5] provide a corpus for argument detection, which we will use as well.

1.4 New idea

We apply a model and corpus from [5]to a different domain than it is designed for, namely Reddit, to see how accurately the model can detect and score arguments. We also want to see if our system scores the delta posts higher than the other comments, since these posts apparently were more convincing.

The corpus contains data from internet pages such as news reports, blogs and debate forums. Since Reddit is a similar domain, we expected that a model trained on the corpus data would still be able to perform reasonably well on Reddit. It is a challenging domain, since language use and text structure can be quite different due to the large difference in topics. As an example of that, below are three topics on the forum that belong in three different domains (biology, culture and philosophy):

- CMV: Your birth wasn't an accident and it wasn't random
- CMV: Inconsistent treatment of TV personalities who make rasist remarks, actually makes racism worse.
- CMV: Objective quality does not exist in art.

2 Methods

There are various components to our system that need to be described. First up is the data gathering: both corpora have to be selected to train our Neural Networks on, and Reddit data has to be gathered to evaluate our cross-topic argument mining. We will briefly describe how the word2vec and LSTM models work. Lastly, we will describe our implemented argument mining pipeline that we apply on Reddit posts.

2.1 Gathering corpus data for training

In this work we use two corpora to train our models on. First, we have the corpus dataset from [5]. As described above, this corpus contains sentences, the topics that they relate to and their label. Technically, the sentences are labelled with the following labels: "argument_for", "argument_against" and "no_argument". Since we aren't interested in argument stance detection, we simply combine the first two labels into the positive class (argument) and the third one into the negative class (no argument). The corpus contains 8 different topics: abortion, cloning, the death penalty, gun control, marijuana legalization, minimum wage, nuclear energy and school uniforms.

The second corpus we used was created by the authors of [6]. Unlike the first corpus, which is focused on argument sentence detection, this corpus is about scoring the quality of an argumentative sentence. 4 topics are used here, which are: Gun Control, Gay Marriage, Death Penalty and Evolution. Each sentence in the corpus has an annotated score between 0 and 1, indicating the argumentative quality of the sentence with respect to its topic.

2.2 Gathering Reddit data for evaluation

The first step is to gather data on topics. The way we do this is to use a simple web crawler/web scraper to gather posts from Reddits CMV subreddit. The scraper follows the html structure of $https : //old.reddit.com/r/changemyview/$ as it is less protected against multiple page requests in a time frame and therefore better suited for the web scraper. The scraper is build with $Python$ and implements $Beautifulsoup4$ library to read the html content of the webpage. In Figure 2 the flowchart of the webscraper is explained visually.
At the start is the main webpage, which has a list of 25 topics. Therefore in order to get a large body of topics the url of the next page is found that gets the next 25 topics. When a topic is selected the OP name and post is collected and checked whether the OP has changed his or her view by rewarding delta's to other commenters. If this is not the case, the topic gets dropped and the next topic of the list will be used. However, if delta's are rewarded by the OP, the text and name of these deltaposts is collected. In the return function the results contain the topic, text of the OP, the delta posts and all other comments.

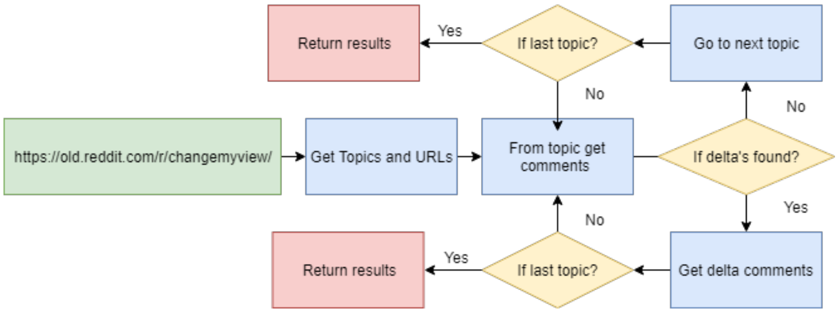

Figure 2: Flowchart of web scraper

2.3 Word2vec and LSTM

Word2vec, as described by [4], is a Neural Network that is trained to produce dense word embeddings which can be very useful for Natural Language Processing (NLP). Such a network can be trained in different ways on textual data. Given a word, other words have to be predicted that occur around the given word in a specified window. In this way, the network learns which words occur in similar contexts. Words with similar contexts will be mapped to similar vectors, such that words in the vector space created by the Word2vec model will be close to vectors of words that are related or are similar in meaning. Once trained, a word2vec model maps an input word to a N-dimensional feature vector representation, where N is specfied during the training phase. In this research, a pre-trained Word2vec model trained on millions of words from Google News is used that will map a word to 300 features compact features.

The Long-Short Term Memory (LSTM) is a recurrent Neural Network first proposed in [3]. Unlike non-recurrent networks such as the Multi-Layer Perceptron (MLP) that treat each new input it gets independently of the other inputs, the LSTM has recurrent connections which allows it to "remember" inputs that it has seen before. This makes it a fitting model for sequence prediction tasks for NLP. In our work and in [5], we treat a sentence as a sequence of words. The LSTM will then process each word from the sequence one-by-one, at each word updating its prediction about whether or not the entire sentence is argumentative or not by using the context or its memory of the words that it has seen before in the sentence. The model learns which words and which order of words together make up good argumentative sentences, by training it on the corpus data. Since the LSTM (and most classifierS) cannot handle strings, words have to be encoded somehow. As we described, this is done with the Word2Vec model. In addition, as described in [5], we add one more feature to the feature vector of each word. First, all the words from the topic are embedded using Word2vec. We then compute the average word vector for the topic. Finally, we compute the cosine similarity between the current word vector and the average word vector. This similarity is the new feature. This means that each word in a sentence is converted to a 301-dimensional feature vector.

Since we are dealing with two different corpora, we also use two LSTM models. The architecture for these models is the same except for the output layer. The model trained on the argument detection corpus, which we name the detection model, performs binary classification. The LSTM trained on the argument quality corpus, which we name the quality model, performs regression on the argumentative sentence scores. Both models will be used to compute the final post score of a Reddit post.

2.4 Argument Mining Pipeline

2.4.1 Performance measure

In this research, we wanted to study whether it was possible to automatically detect and score arguments from online discussions. On Reddit's Change My View subreddit, the delta system provides an interesting way of separating good posts from bad posts. The assumption that we are making is that if a post receives a delta then it should be more convincing and of higher argumentative quality compared to non-delta posts. We will make our system score delta posts and see if these delta posts receive higher scores than non-delta posts. Our accuracy per topic will then be computed as follows:

$$accuracy = \frac{1}{N}\sum_{j}^{N}\frac{1}{k}\sum_{i}^{k}\frac{|C_{ji}|}{|C_j|}, \tag{1}$$

where N is the amount of topics, 135 in our case, k the amount of delta posts and C_j all the posts from topic j minus the delta posts. C_{ji} is the set of comments from the topic that have a lower score than delta post i from topic j: D_{ji}. C_{ji} is determined as follows:

$$\{C_j \mid S(C_j) < S(D_{ji})\}, \tag{2}$$

where $S(x)$ is our scoring function which is a function of our detection and scoring model and D_{ji} is delta post i from topic j. In short, the accuracy of our system is the average topic accuracy, where the topic accuracy is determined by the average delta post accuracy, where the delta post accuracy is determined by the amount of normal posts that have a lower score than our delta post score. Since a topic can have multiple delta posts, we compute the delta accuracy for each and then take the average.

2.4.2 Gathering Reddit topics

First, Reddit data is gathered. The idea was to test our system on around 100 topics to get a good overview of how our system performs. Unfortunately, not all topic posters are easily convinced and thus topics exist in which no deltas have been given out to any posters. Since we want to compare delta posts to normal posts, there is no point in analyzing these topics. We will therefore scrape a set number of topics, but skip the ones without any delta posts. We have seen that on average, approximately 1/3 of the topics scraped have delta posts. We therefore scrape 300 topics. In the end, this resulted in 135 topics, all with delta comments. The results of the scraping stage gives us a set of topics, their corresponding titles, normal comments and delta comments.

2.4.3 Classifying sentences from Reddit posts

After collecting the data, we process the comments for each topic. For each comment, we first have to divide it into sentences. We do this by simply splitting the posts based on the full stop '.'. This gives us sentences, which we send to the Word2Vec model. Each sentence is tokenized and non-alphabetic tokens are removed. The remaining word tokens are encoded and the topic similarity is computed between each word and the title of the topic, leading to the 301-dimensional feature vector representation for each word.

Words in a sentence that do not exist in the Word2vec are a problem, however, as no learned vector representation for these words exist in this case. We use a similar approach as described in [5], where each out-of-vocabulary word is instead mapped to a random vector. Each feature from this vector is a random number in the range [-0.01, 0.01]. This method does not disturb sequence/sentence lengths, unlike removing out-of-vocabulary words. Since the random values from these words are low, they should not affect the results too much and instead act as noise.

After each word from each sentence is encoded, we can apply our detection model. For each post, the detection model will look at each sentence. All of the sentences will be parsed by the detection LSTM to perform binary classification. At the end of the sequence, the final LSTM output will be taken as the probability that the sentence is an argumentative unit. To determine whether the sentence is 'argumentative enough', the LSTM output has to be higher by our pre-defined threshold parameter. This parameter is set to 0.5 by default, but we will try different threshold values to observe the difference in performance. If the LSTM output is larger than the threshold it will be put into the argumentative set. If it is lower than or equal to the threshold, the sentence will be put into the non-argumentative set. The entire non-argumentative set is removed and only the argumentative set will go to the next stage in the pipeline for scoring.

2.4.4 Scoring argumentative sentences for quality

At this point, for a specific post, only the argumentative sentences are left and (hopefully) all non-argumentative units have been filtered away. All that is left to do is to score the argumentative units for argument quality. This is done by feeding these argumentative sentences to the quality model, which performs regression. The quality model will also return a value between 0 and 1, indicating quality, where 0 is a very bad argument, while 1 is a really good argument. For this step, no threshold is used: the output of the quality LSTM is the score belonging to the input sentence.

2.4.5 Scoring Reddit posts

In the end, all sentences have been processed, such that non-argumentative sentences (according to our model) have been filtered away, and every argumentative sentence has both a probability of being an argumentative unit and a quality score. We then define the total score of a Reddit post as the dot product between the sentence probabilities and qualities:

$$S(P) = \sum_i F(P_i) * Q(P_i), \tag{3}$$

where P is the input post that contains sentences, P_i is sentence i from post P, $F(x)$ is the filter function that filters the argumentative from the non-argumentative units using the detection model given an input sentence x and $Q(x)$ is the quality scoring function using the quality model that scores its input sentence x for quality. By using two models trained on different argument corpora, we tried to improve our accuracy and make the system more robust by combining two classifiers instead of relying on one model only. According to our scoring model, a sentence that is clearly very argumentative (high probability) but a low quality score will have a lowered score. An argumentative unit which is not very likely to be argumentative (low probability) can still have good argumentative quality, which will boost its score.

After having computed the scores for all delta and non-delta posts for a specific topic, we compare the non-delta post scores to the scores of the delta posts. We then compute the accuracy for the specific topic as described above.

2.4.6 Gathering the results

For the results, we are interested in the accuracy across all 135 topics. We will apply our system to compute the average topic accuracy and do this for different values of the threshold parameter, which we will vary from 0.1 to 0.9. A low threshold should lead to more noise, as sentences that are quite bad and barely argumentative will still be accepted as arguments. Hopefully, the quality assigned to these sentences by the quality model will not be very high. However, since the probability of these bad argumentative sentences, 0.1 at worst, will be multiplied with the quality score which will severely reduce the impact of this sentence on the total post accuracy. Conversely, setting a high threshold will make the detection model very picky and may skip a lot of sentences that actually provide a lot of argumentative strength to a post.

Once we have gathered the accuracy data for every threshold value, we will look at examples of how our model classifies posts, based on classifications done with the best threshold value. These textual results will be added to the appendix of the report for inspection.

3 Results

Threshold	Accuracy	Variance
0.1	0.668	0.031
0.2	0.659	0.038
0.3	0.665	0.042
0.4	0.662	0.058
0.5	0.615	0.087
0.6	0.546	0.111
0.7	0.493	0.127
0.8	0.408	0.144
0.9	0.260	0.139

Table 1: Mean accuracy and variance for different values of the threshold parameter.

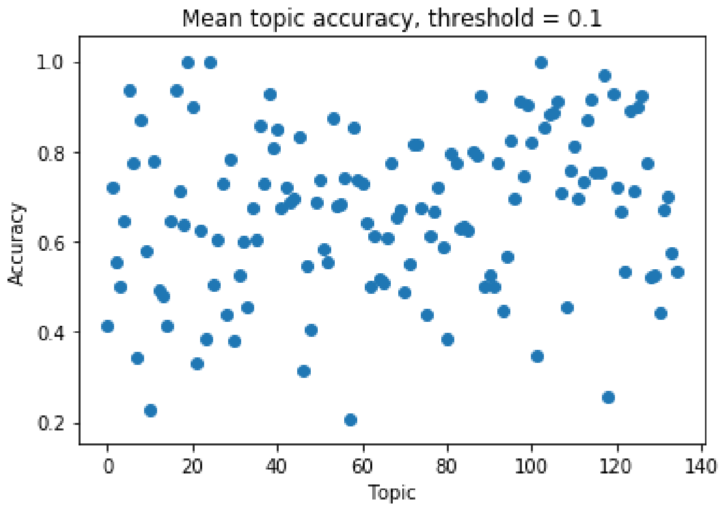

Figure 3: Accuracy of our system on 135 different Reddit topics with threshold parameter of 0.1. The highest score achieved for a topic was 100%, the lowest was 20%.

The accuracy scores for the different threshold values are shown in Table 1. The accuracies for all 135 topics for the best threshold value, 0.1, are shown in Figure 3. Three topics have been selected: one from a topic where 100% accuracy was obtained, one from a topic where around 50% accuracy was obtained and a third one where the accuracy was around 20%. The output of our system on these three topics are shown in the appendix.

4 Conclusion

4.1 Discussion

The values from Table 1 show us that our system with a threshold of 0.1 was able to achieve an accuracy of 66.8%. This can be interpreted in the following way: on average, over 135 topics, delta posts were classified to be better than 66.8% of the other, non-delta comments. This seems to be a decent result and seems to confirm our belief that delta posts have inherent higher argumentative quality compared to non-delta posts.

When the threshold is increased, more and more sentences will be put into the non-argumentative set. This leads to sentences being removed that can be argumentative. When the threshold is set too high, most sentences will end up into the non-argumentative set and a lot of the posts will get a score of 0. This means that a lot of delta posts but also a lot of normal comments will get a score of 0. Since we define our accuracy based on the delta posts being higher than and not equal to the score of normal comments, the accuracy of the system will go down quickly. This can be seen when we look at the accuracy for a high threshold value, such as 0.9,

When we have a low threshold value, such as 0.1, a lot of sentences are put into the argumentative set while mostly noise and unrelated sentences will be filtered away. This leads to more arguments being kept and scored. If a sentence turns out to not be very good in terms of argumentative quality, it's low probability score and/or low quality will drag the post quality down.

If we look at Figure 3, we can see a big variability in the accuracy of the different topics. The models were only trained on a very small number of topics and it seems that, depending on the topic and the writing style, the accuracy can fluctuate a lot between topics.

One of the assumptions our model makes is that the OP has read all the comments and based his delta assignment on this. However, it is likely that this is not the case and some posts are not read due to time limitations, because of the settings (best/top/controversional/old/suggested) used or randomly missing it while scrolling. It is however likely that the OP is invested in the topic, otherwise they would not post it online. It is reasonable to assume that most of the posts are read and influence the decision of the OP to reward delta's.

A different issue is that the model predicts classification of delta's by individuals purely on what it has learned about argumentation from the corpus it has trained on. This means that the model might predict better arguments than the OP has awarded delta's to, but the OP that classifies a post might not find them particularly persuasive or might not have read it as is earlier described. This means that our model might be better at predicting good arguments based on the general rules of argumentation, but might not be able to cope as well with individual bias. In this case, accuracy will be lower because the 'best' arguments, according to our model, were not given any delta's by the OP.

Another problem is if a delta comment is "poorly" written, for example when many typo's or specific abbreviations are used, while a human could understand it due to the human cognitive ability. For example, in one of the topic title examples given earlier, the word 'racist' was misspelled. These misspelled words will most likely not occur in the Word2vec dictionary and thus the word will be embedded as a small random vector and will not have a big influence on the classification, while the word might have had a big influence if it was spelled correctly. This can decrease the score of an argument, even though the argument itself can be quite powerful. This is a problem that could be solved by implementing auto-correct on words that have typing errors and/or a database of common abbreviations. Additional improvements such as proper lemmatization of words could help here as well. One more possible improvement would be to make a new Word2vec model. The model used in this research was pre-trained on news articles. Instead, a new Word2vec model could be trained on millions of online discussions, such that popular online slang words, misspellings and the like all get their own vector embedding, which could greatly improve the performance.

We now move on to inspect the three examples of delta post classifications mentioned before, which are listed below in the appendix. As we can see, the total post quality is the highest for the first topic and the lowest for the third topic. This makes sense, but is not always the case. Good delta posts, for example, can have much lower total post quality sometimes, but since it's all relative to the other, non-delta comments, the accuracy can still be good.

We first inspect the medium delta post example, which is about the following topic: " *I dont think its an unfair consequence to detain and desperate families illegally crossing into the US*". What's interesting about the output here is that more objective sentences get a higher probability and quality score than sentences which are more subjective. For example, the sentence "*When someone commits an illegal action, it is NOT okay to punish their children"* gets a much higher score compared to a sentence like "*Almost everyone agrees on this, and I'm happy to ignore the people who don't*". This seems like a good classification, as more objective argumentative sentences contribute more to the power of an argument. If we look at the bad arguments, it's not always obvious why these have been filtered away. Some of them are filler sentences, but there are still decent arguments in there.

Second, we look at the bad delta post example, which is about the following topic: "*Transgender rights is a low priority issue and the left should divert its political capital elsewhere for now*". We see that the total post quality is low. It is not very obvious why the score is so low, but it seems that there are not a lot of good arguments present in this delta post. However, the best argumentative sentence (in our opinion) is the following: "*They want to live their lives without a bunch of reactionaries throwing themselves in front of the bus to stop them because of "the children*". This sentence also gets the highest score assigned to it, so even in this bad example there still seems to be some decent classifications. However, we can also see that short, unimportant sentences such as "*That's it*" and "*That's all*" still get a relatively high probability assigned to them. Luckily, the assigned quality of these sentences are low, which compensates somewhat.

Lastly, we look at the good delta post example, which is about the following topic: "*Political debates hosted by news stations need a standardized format on how much time candidates can answer and rebuttal to questions and answers*". The sentences that are classified as arguments don't seem to get a very high score. However, the main summarizing point of the delta post's argument, which is the sentence " *all vary and it makes much more sense to be flexible than to stick to a rigid process*" still gets a very high probability and quality score, which seems like a good classification. However, it's not entirely clear why some of the "bad arguments" were not classified as arguments, as some of these sentences seem like better arguments than the ones that are actually in the argumentative set.

4.2 Relevance

As the technological age continues more information and communication is online in a text-based format. This is a huge opportunity for argument miners to understand and even influence people in a relevant way. Up until recently most argument mining was purely academic endeavor, but using this knowledge on real text-based conversations is a huge step forward to the point that everyday practical applications of argument mining is possible. The current research has scratched the surface of online debate in finding the relevant arguments that changes a persons mind about a topic which the OP holds dear. If our current method can be improved it is a highly useful and powerful tool to influence peoples thoughts about certain topics.

When a practically good online argumentation miner is constructed the online conversation can be held to a higher standard, because bad arguments will also be classified. As fake news is a big topic in the media right now, a good counter method is to analyze whether the claims that are made actually have good supporting arguments. A argument analyzer could help plow through the large quantity of texts online to help people make up their mind in a well informed way. Implementing a cross-platform argument miner on a online debating forum can help provide that clarity.

The applications described above are based on cross-platform implementation as was the current research, but argument mining can also be used in a research setting. A usage is the **Knowledge Extraction and Consolidation for Scientific Publications in the Educational Domain** and some implementations of that have been done in previous publications. Today, the number of publications increases heavily which leads to a large effort for researchers to find all relevant information on a topic. Therefore, methods and tools which are able to automatically extract and consolidate knowledge from scientific publications in the exemplar domain of educational research can be of great value to scientific researchers.

References

[1] Katarzyna Budzynska and Serena Villata. Argument Mining. *IEEE Intelligent Informatics Bulletin*, 2016.

[2] Ivan Habernal and Iryna Gurevych. Argumentation mining in user-generated web discourse. *CoRR*, abs/1601.02403, 2016.

[3] Sepp Hochreiter and Jrgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[5] Christian Stab, Tristan Miller, and Iryna Gurevych. Cross-topic argument mining from heterogeneous sources using attention-based neural networks. *CoRR*, abs/1802.05758, 2018.

[6] Reid Swanson, Brian Ecker, and Marilyn A. Walker. Argument mining: Extracting arguments from online dialogue. In *SIGDIAL Conference*, 2015.

Appendix: Great delta post example

```
topic: CMV: Political debates hosted by news stations need a
    standardized format on how much time candidates can answer and
    rebuttal to questions and answers

————————Good arguments————————:
Probability: 0.18165874
Quality: 0.04470614
```

```
 If they want 90, they get 90

Probability: 0.13567969
Quality: 0.0278745
 In others, it makes sense to have long ones

Probability: 0.10369605
Quality: 0.02849899
 In others, it makes sense to be aggressive and talk over one
    another

Probability: 0.93759763
Quality: 0.6936549
 all vary and it makes much more sense to be flexible than to stick
    to a rigid process

Total post quality: 0.19868363
———————Bad arguments————————:
Probability: 0.03660166
Political candidates aren't required to participate in any debate

Probability: 0.02165497
 If I call up Trump and Elizabeth Warren and ask them to participate
     in a debate for my Youtube channel, they can say no

Probability: 0.00627023
 The same applies to CNN, Fox News, or any other media outlet

Probability: 0.00463487

As such, the TV channels don't set the debate rules, candidates do

Probability: 0.00086634
 All debates rules are agreed upon by the candidates in advance

Probability: 0.03160067
 If they agree to 30 seconds, they get 30 seconds

Probability: 0.05194087

That's why there won't be any standardized rules

Probability: 0.03662102
 Candidates are maximizing their chances of winning

Probability: 0.01540928
 In some cases, it makes sense to have shorter debates with shorter
    times

Probability: 0.01443781
 In some debates, it makes sense to look civil

Probability: 0.00977282
 Candidates, the race, the issues of the day, people's attention
    spans, etc
```

Appendix: medium delta post example

Topic: CMV: I don t think it s an unfair consequence to detain and desperate families illegally crossing into the US

——————Good arguments——————:
Probability: 0.56130093
Quality: 0.37403068
So, when someone commits an illegal action, it's okay to punish them

Probability: 0.18275829
Quality: 0.06535857
Almost everyone agrees on this, and I'm happy to ignore the people who don't

Probability: 0.61422455
Quality: 0.45610163

When someone commits an illegal action, it is NOT okay to punish their children

Probability: 0.18094648
Quality: 0.064077
Almost everyone agrees on this, and I'm happy to ignore the people who don't

Probability: 0.82589775
Quality: 0.5568506
When a child participates in a crime with their parent, the assumption is that the child did not have the ability to refuse to participate

Probability: 0.7362224
Quality: 0.5130773

From these three things, while we can conclude that it's okay to punish people who illegally cross the border, it is NOT okay to punish their children

Probability: 0.86059
Quality: 0.5272373
So, is holding a child in a detention facility while separated from their parents a form of punishment? I think so

Probability: 0.54259104
Quality: 0.2912231
Honestly, I can't think of any way to think of it other than as a punishment

Probability: 0.26908678
Quality: 0.15552112

Also, Trump's actions are actually even worse, since these people (who entered illegally) are applying for asylum

Probability: 0.18567926
Quality: 0.11209881

```
The rights of asylum seekers, whether or not their application is
    granted, are explicitly delineated by the IJCR

Probability: 0.8228604
Quality: 0.4980579
Between punishing children for the crimes of their parents and
    violating international law, I'd say I have every reason to be
    angry at Trump for this

Total post quality: 0.32851216
-----------Bad arguments-----------:
Probability: 0.083924
Having established that, let's move on

Probability: 0.05457297
Moving on

Probability: 0.07871078

As a general rule, children are not held responsible for their
    actions

Probability: 0.03100271
This gets contentious for children post-puberty, but pre-puberty
    nearly everyone agrees on this

Probability: 0.08641455
So, that's why people are upset

Probability: 0.07114999
Here's a surprise: we're not allowed to separate families of asylum
    seekers

Probability: 0.05022085

So, yeah
```

Appendix: bad delta post example

```
CMV: Transgender rights is a low priority issue and the left should
    divert its political capital elsewhere for now

-----------Good arguments-----------:
Probability: 0.25564098
Quality: 0.05383475
But, by and large, they are

Probability: 0.16594715
Quality: 0.08461817
As you've said yourself, a tiny portion of the population is
    identifies as transgender

Probability: 0.2682305
Quality: 0.16134323
In general, they want to be safe and respected

Probability: 0.2715475
```

Quality: 0.09053905
That's all

Probability: 0.27173212
Quality: 0.10447008
That's it

Probability: 0.544402
Quality: 0.30837712
They want to live their lives without a bunch of reactionaries throwing themselves in front of the bus to stop them because of "the children"

Probability: 0.24649298
Quality: 0.03717636
I'll help them in any way I can

Probability: 0.3406434
Quality: [.11554949

If people don't "like the plate" because people being treated with dignity is too much for them, or reading a book is a bridge too far, well there's no two ways about it

Total post quality: 0.11948853
——————Bad arguments———————:
Probability: 0.0624784
They want to go pee without involving the supreme court somehow

Probability: 0.09820075
I can't and I won't blame them

Probability: 0.0822042
In fact, I'll do the opposite of that

Probability: 0.05181319
Fuck them