Mining Arguments in US Presidential Campaign Debates

Chen Peng, Mariia Poiaganova and Milena Voskanyan
M.Sc Cognitive Systems
University of Potsdam, Germany
firstname.surname@uni-potsdam.de

Abstract

This project is focused on a currently relevant task in the field of Natural Language Processing – Argument Mining. We aim at reproducing the study published by Shorhen Haddadan, Elena Cabrio and Serena Villata, who used machine learning techniques to classify argumentative sentences found in the presidential debates' speeches. (Haddadan et al., 2019) Thus, the interest of the project is motivated by two directions: the first one is the practical study of argumentation mining task using texts from the political discourse along with the capacity of certain computational methods to solve this task. The second direction is the replication potential of the original paper, given no source code. As for our main results, we have been able to replicate the authors' methodology and for three out of eight proposed experiments, we managed to exceed the reported results. For the rest, we have achieved a consistent performance. Our reproduction[1] is of a special interest as it mostly relies on different Python frameworks than the authors used and introduces some methodological expansions to the original version.

1 Introduction

1.1 Argumentation Mining Overview

Following the definition of Frans H. van Eemeren, argumentation aims at increasing or decreasing the acceptability of a controversial standpoint for the listener or reader. It is done verbally, "by putting forward a constellation of propositions intended to justify or refute the standpoint before a rational judge". (van Eemeren et al., 1996) Argumentative structures are usually complex and are comprised of several units – argument components. The central component of an argument is a claim – a controversial standpoint that needs to be justified. Another argument component is a premise – it serves as a reason for the claim's justification or rejection. Claims and premises form argumentative relations between each other based on their support/oppose characteristic. (Stab & Gurevych, 2017)

Argument mining, or computational argumentation, focuses on automated analysis of texts in natural language with respect to their argumentative structure. The sub-tasks of argument mining might include finding argumentation-related passages in the texts and classifying them from non-argumentative (Sardianos et al., 2015), identifying claims and premises pertaining to the controversial standpoint and predicting the relations between argument components (Stab & Gurevych, 2017), generating new arguments (Schiller et al., 2021). Within the framework of our project, following the research question of (Haddadan et al., 2019), we focus on two tasks: 1) argument identification and 2) classification of claims and premises as argument components.

Decades of computational approaches to argument mining have resulted in various corpora with argument annotation. The discourse of the corpora materials also varies. For example, (Stab& Gurevych, 2014) present a corpus of persuasive essays, comprised of 90 essays split into 1.673 total sentences. The annotations mark claims and premises as well as support and attack relations. Another corpus example is a representative of legal domain. (Poudyal et al., 2019) The researchers built an annotated corpus of 42 decisions of the European Court of Human Rights (ECHR). The corpus is annotated in terms of three types of clauses useful in argument mining: premise, conclusion, and non-argument parts of the text.

Political debates, being a natural collection of

[1] https://github.com/pc852/Argument-Mining-Presidential-Debates

controversial statements and supporting judgements, plays a special role as material for argument mining. Several computational approaches have been tested on political discourse data. For example, (Lippi & Torroni, 2016) have developed an original dataset based on the 2015 UK political elections debates. They combine features from texts and speech to test the potential improvement of claim detection by expanding the feature set to the spoken language dimension. Another example is the work of (Visser et al., 2019). The authors build US2016, the corpus of transcriptions of television debates leading up to the 2016 US presidential elections, and reactions to the debates on Reddit. The paper we aim at reproducing introduces a novel corpus of political debates. It includes speeches from 39 political debates, split into 34.013 sentences, where each sentence is split into argument components (if any) and annotated as claims or premises. To the best of our knowledge, this is the largest corpus of political debates tailored for argument mining.

1.2 Reproducibility Challenge

A substantial interest of our project is motivated by the process of reproducing the mentioned research and the results of the reproduction. Recently, the topic of studies' reproducibility has been actively discussed in computer science and especially in its domains like Machine Learning and Natural Language Processing. The problem is put on the agenda as more and more researchers report failures to reproduce the results of other studies. According to Monya Baker, 70% of scientists faced the impossibility to reproduce someone else's results, and more than a half did not manage to reproduce their own. (Baker, 2016) At the same time, only 15% of AI studies share their source code publicly[2]. Failure to repeat the results and the absence of the source code are likely to be tightly connected. To facilitate reproducibility and openness of science, organizations like ACL (Association for Computational Linguistics)[3] and ACM (Association for Computing Machinery) encourage authors to share the data and the source code and set it as a requirement to get published.

Speaking about reproducibility itself, it is important to set the terminological paradigm used in this project. Paper reviews repeatedly show that there is no certain definition to the terms reproduction/reproducibility and to replication/replicability, having usually a synonymic meaning. (Cohen et al., 2018) In the framework of our project, we will rely on the definitions provided by the Association for Computing Machinery.[4] With respect to it, reproducibility is when "an independent group can obtain the same result using the author's own artifacts", and the results can be considered as reproduced when "the main results of the paper have been obtained by a person or team other than the authors, using, in part, artifacts provided by the author". Replicability "means that an independent group can obtain the same result using artifacts which they develop completely independently" and the results are replicated if "the main results of the paper have been independently obtained in a subsequent study by a person or team other than the authors, without the use of author-supplied artifacts".

Under such terminological framework, our project falls into the first category of reproduction. We create an entirely new software, not based on any other pre-published source code, but we use the exact same data, including train, validation and test splits, as the authors did. We first build the related software based on descriptions provided in original paper and then, do several adjustments to it upon the information provided by the authors after a personal contact, scheduled to clarify the methodological details of the research.

2 Data

2.1 Corpus

The dataset *USElecDeb60To16 v.01*[5] is a set of debate transcripts from past United States presidential debates, taken from the website of the Commission on Presidential Debates[6]. It includes debate transcripts dating from 1960 to 2016. The corpus was annotated by expert annotators. Sentence components were marked as premises, claims, or none of the two. The definition of claims and premises with respect to the corpus annotation are detailed in the following subsections.

According to the original paper, three expert annotators independently conducted the annota-

[2] https://www.stateof.ai/

[3] https://www.aclweb.org/portal/

[4] https://www.acm.org/publications/policies/artifact-review-and-badging-current

[5] https://github.com/ElecDeb60To16/Dataset

[6] https://www.debates.org/

tion of 39 debates, relying on the guidelines handbook created by other expert annotators. The observed agreement percentage and inter-annotator agreement at sentence-level (following (Stab& Gurevych, 2014)) for argumentative - non- argumentative sentences are 0.83% and $\varkappa$ = 0.57 respectively. For the argument component they are 63% and $\varkappa$ = 0.4.

In our experiments, like in the original paper, we conduct two subsequent binary classification tasks: arguments vs non-arguments and premises vs claims. Both tasks were carried out on a sentence level. When a sentence contained both a claim and a premise, the sentence was labeled according to the longest component. Table 1 shows the number of sentences per each class and the total number of sentences in the corpus.

Total	Argument	Non-Argument	Claim	Premise
29.621	22.280	7.252	11.964	10.316

Table 1: Classes distribution in the dataset, displayed in the number of sentences

We use exactly same data splits for train, test and validation sets as the authors did. Table 2 displays the statistics of these sets.

	Train	Test	Validation
Task 1	14.044	8.455	7.033
Task 2	10.464	6.575	5.241

Table 2: Data splits statistics, displayed in number of sentences

2.2 Data classes

This subsection describes data classes as applied to the corpus annotations. Certain examples of claims and premises from the corpus are provided. Claims are marked in bold, premises in Italics and the component boundaries are defined by [square brackets].

Arguments. A statement which aims at persuading the audience can be described as an argument. Its components are claims and premises, where claims are the important part of an argument, because the assertion of a claim is its goal. There are several structures of an argument, and the simplest is a claim assisted by a combination of premises that are justifying it.

Claims. In political debates, a claim is a policy stated by a party or a candidate to be undertaken which needs a reason that supports it. In the example presented below, Kennedy expresses his opinion on the issue with the farm policy, explaining what consequences the country could face if the policy, which he claimed to be failing, continued. These kind of arguments might include discourse connectives, such as “I believe”, “in my opinion”, “I think”, etc., which makes it easy to indicate claims.

(1) Kennedy-Nixon, 1960

KENNEDY: *[So if the farmers’ economy continues to decline as sharply as it has in recent years, then I think you would have a recession in the rest of the country]. [So I think the case for the government intervention is a good one]. [Secondly, my objection to present farm policy is that there are no effective controls to bring supply and demand into better balance]. [The dropping of the support price in order to limit production does not work, and we now have the highest uh - surpluses - nine billion dollars worth]. [We’ve had a uh - higher tax load from the Treasury for the farmer in the last few years with the lowest farm income in many years]. [***I think that this farm policy has failed**].

Other ways of representing claims are by adopting a position towards a policy, as in (2), and answering to a question asked from a certain candidate (3):

(2) Bentsen-Quayle, 1988

QUAYLE: *[The premise of your question, John, is that somehow this administration has been lax in enforcement of the OSHA regulations]. [***And I disagree with that**]. *[And I’ll tell you why. If you want to ask some business people that I talk to periodically, they complain about the tough enforcement of this administration and, furthermore, let me tell you this for the record, when we have found violations in this administration, there has not only been tough enforcement, but there have been the most severe penalties - the largest penalties in the history of the Department of Labor - have been levied when these violations have been found].*

(3) Bentsen-Quayle, 1988

MARGOLIS: Senator, we’ve all just finished - most America has just finished one of the hottest summers it can remember. And apparently this year will be the fifth out of the last nine that are among the hottest on record. No one knows, but most scientists think, that something we’re doing, human beings are doing, are exacerbating this problem, and that this could, in a couple of generations, threaten our descendants comfort and health and perhaps even their existence. As Vice President what would you urge our government to do to deal with this problem? And specifically as a Texan, could you support a substantial reduction in the use of fossil fuels which might be necessary down the road? BENTSEN: Well, [**I think what you can do in that one**], and which would be very helpful, **[is to use a lot more natural gas, which burns a lot cleaner]**.

Premises. In order to support or attack a claim, a reason or justification, also defined as premise, are given. In political debates, politicians usually use examples from the past as in (4), where the candidate aims at proving the efficiency of his office, by giving some events which took place during his presidency. Besides, some discourse connectives (e.g., "because", "for example", "since") might appear in premises.

(4) Obama-Romney, 2012

OBAMA: **[America remains the one indispensable nation]. [And the world needs a strong America], and [it is stronger now than when I came into office].** *[Because we ended the war in Iraq, we were able to refocus our attention on not only the terrorist threat, but also beginning a transition process in Afghanistan]. [It also allowed us to refocus on alliances and relationships that had been neglected for a decade].*

In (5) premise is presented as an example and proves the claim to be right. Another way of justifying a claim is to use statistics (6), thus making the point more convincing.

(5) Trump-Clinton, 2016

CLINTON: **[Race remains a significant challenge in our country]. [Unfortunately, race still determines too much]**, *[often determines where people live], [determines what kind of education in their public schools they can get], and, yes, [it determines how they're treated in the criminal justice system]. [We've just seen those 12 two tragic examples in both Tulsa and Charlotte].*

(6) Nixon-Kennedy, 1960

NIXON: We often hear gross national product discussed and in that respect may I say that when we compare the growth in this administration with that of the previous administration that then *[there was a total growth of eleven percent over seven years]; [in this Administration there has been a total growth of nineteen percent over seven years].* That shows that **[there's been more growth in this Administration than in its predecessor].**

3 Reproduction Experiments

3.1 Problem Setting

We attempt at reproducing both classification tasks with the respective machine learning model archetypes as Shohreh Haddadan and colleagues did. The first task, later referred to as Task 1, is classifying the sentences as either an argumentative ones (containg a claim or a premise) or not. The second task, later Task 2, is classifying an argumentative sentence as either a premise or a claim. The datasets used for training and evaluation are identical to those used by the authors. Not all hyper-parameter and architecture settings of the models were provided as reference, however, we shall still specify our computational details below.

3.2 Methods

The methodology of the project is entirely reflected by the original paper. We, like (Haddadan et al., 2019), split the practical part into four experimental settings. The splits are based on different methodological approaches.

Within the first setting, we create tf-idf matrices of unigrams to represent the sentences and use them to train linear kernel SVM (Support Vector Machines), the penalty parameter C is 10. The second setting is the extension of the former. Tf-idf unigram matrices are enriched by bigrams and trigrams and several engineered features (see the subsection Features). These linguistic features and tf-idf matrices form an input for another SVM model, with rbf kernel and C equal to 10. For both settings with SVM models and both classification tasks, we first replicate the models with original parameters, and then tune the hyperparameters on the validation set.

For the third setting, we test a LSTM (Long short-term memory) (Hochreiter& Schmidhuber, 1997) neural network model with word embeddings. We train a LSTM model with an input embedding layer, a bidirectional LSTM layer, and an output layer, using the python libraries Keras and Tensorflow[7]. The input embedding layer has a dimension of 300, corresponding to the pre-trained Fasttext (Joulin et al., 2016; Mikolov et al., 2018) embeddings. The LSTM layer has 128 neurons for

[7] https://keras.io/about/

the first task and 200 neurons for the second task. The output layer uses the sigmoid activation function. The model is compiled with the binary cross-entropy loss function and the Adam (Kingma & Ba, 2014) optimization algorithm.

For the fourth setting, we test a feed-forward neural network (FFNN) which takes the full set of engineered features as input. Two hidden layers are of 64 and 32 neurons respectively. The first hidden layer uses a rectified linear unit (ReLU) activation function while the second hidden layer and the output layer use the sigmoid activation function. The FFNN is compiled with the same parameters as the LSTM model.

All four settings were used to approach both classification tasks. Our GitHub repository contains a folder with eight notebooks demonstrating four methodological settings × two tasks [8].

3.3 Features

Two out of four experimental settings required engineered features as models' input. For each of the listed features, we implemented a separate function from scratch, either with the use of a side Python library or not. The motivations behind the features and computational procedures are described below.

Tf-idf features. The lexical pillar of the sentences was introduced by tf-idf matrices. We utilize scikit-learn library to model those. Unlike in the first experimental setting, where tf-idf matrix included only unigrams, in the second and the fourth settings, we include bigrams and trigrams. Following the authors' practice, our matrix has only most occurring uni-, bi- and trigrams. We set the threshold to 10.000 n-grams.

Part-of Speech for adverbs and adjectives. As mentioned in the original paper, certain adjectives and adverbs serve the purpose of characterizing the debaters' premises and their correctness. We make the use of SpaCy[9] POS model and make two features based on the number of either adverbs or adjectives occurring in the sentence.

Tenses for verbs, modal verbs. Past tense verbs used by debates carry a meaning of referring to the past and assessing the events, thus, providing the basis for premises. Modal verbs can indicate the certainly of assertions made by debates. We use SpaCy POS model and retrieve a detailed POS tag for each word. By that, we collect tags for each verb and a tag for modal verbs. We create several features for each tense category and for the modal verb category. The features are based on the number of words of a certain category occurring in the sentence.

NER features. Motivation behind Named Entity Recognition labels is, as suggested in the original paper, to facilitate the recognition of premises. Politicians often use the names of other presidents, party members, provide statistics to strengthen their claims. We use SpaCy NER model and create features for each possible NER label, based on the number words with a respective label occurring in the sentence.

Syntactic features. Syntax conveys the complexity of the structure of a sentence. For each sentence, we get a consistency parse tree with nltk functionality and pre-defined formal grammar[10]. Inspired by the approach in (Stab & Gurevych, 2014), based on the parse tree, we create a feature with the number of productions in a sentence and a feature with the number of verbal phrases (VP) in a sentence. Also, we create a feature to capture the depth of a tree of a sentence.

Sentiment polarity. We create one feature for each sentence, indicating the sentiment polarity of a sentence – the debaters' attitude towards the topic. To model this, we utilize VADER [11] package and process each sentence through it to assign a sentiment score to a sentence. The scores are distributed from -1 to 1, where -1 is negative and 1 is positive.

Discourse connectives. Argumentative sentences are usually supported by linking devices, especially those indicating the effect, the cause, the result of something. We analyzed the list of discourse connectives for the English language (Das et al., 2018) and found out that many connectives could bias the model because of their ambiguity (e.g., 'so', 'because', 'given'). Thus, we manually extracted several connectives based on, first, their meaning (indicating the cause or the result), and second, on their unambiguous sense. We created a Boolean feature standing for the presence of at least one of these connectives in a sentence.

First-person pronouns. In addition to the authors' feature list, we define one Boolean feature

[8] https://github.com/pc852/Argument-Mining-Presidential-Debates/tree/main/code/experiments

[9] https://spacy.io/usage/linguistic-features

[10] https://www.nltk.org/book/ch08.html

[11] https://github.com/cjhutto/vaderSentiment

Experimental Setting	Class	Results from (Haddadan)	Our results, replicated	Our results, tuned
Majority Baseline	Arg	0.810		
	None	0.000		
	Average	0.551		
Tf-idf + SVM	Arg	0.855	0.848	**0.894**
	None	0.486	0.465	0.457
	Average	0.737	0.763	**0.797**
All features + SVM	Arg	0.916	0.896	0.895
	None	0.433	0.399	**0.498**
	Average	0.823	0.785	0.807
Word embeddings + LSTM network	Arg	0.913	0.902	
	None	0.547	0.524	
	Average	0.843	0.818	
All features + FFNN	Arg	0.872	**0.887**	
	None	0.498	**0.538**	
	Average	0.800	**0.809**	

Table 3: f1-scores for task 1, classification of argumentative and non-argumentative sentences

for the presence of a singular first-person pronoun (forms like ‘i’, ‘me’, ‘mine’, ‘my’, ‘myself’). Similarly, we define one Boolean feature for first-person plural forms (‘we’, ‘our’, ‘ours’, ‘ourselves’). These features were motivated by the work of (Stab & Gurevych, 2014). We introduce them as a potential attempt to extend the authors’ feature approach and examine the outcome in the models’ scores.

4 Results and Discussion

To evaluate the models, we followed the authors’ protocol for both tasks. For each class (argumentative/non-argumentative), (claim/premise), we used precision, recall and f1-score measures and calculated them on the test set after training the models. We also calculated the weighted f1-score for both classes. As a baseline model, we used the majority baseline scores, presented in the original paper. Being the harmonic mean between precision and recall, f1-score served as main measure for models’ evaluation and performance comparison. These f1-scores over all experiments for task 1 and task 2 are displayed in Table 3 and Table 4 respectively. The numbers in bold represent the cases when our reproduction exceeds the original score. Detailed results for all experiments can be found in the notebooks in the project repository.

4.1 Task 1: Arguments and Non-arguments

For the task one, classification of argumentative and non-argumentative sentences, all models significantly outperform the baseline – the same result is seen in the original paper. As for the reproduction part, the first experiment with tf-idf unigrams and identical to authors’ SVM parameters shows higher average score than the original one. After hyperparameter tuning, we report performance improvement of 6% comparing to the original research, and the best parameters being C equal to 1 and gamma equal to 1. The results of the second experimental setting show that both in our reproduction and in the original paper, the enlarged set of features leads to performance improvement. However, our reproduction, even after hyperparameter tuning, is not reaching the original score, being 1.6 % lower. Even though it is slightly lower, we would still consider such results as consistent to the original. The potential explanation to that might lie in the way the features were gathered. As found out upon the personal meeting with authors, they mainly used CoreNLP package for feature engineering, while we utilized SpaCy, nltk and VADER. With the third model, word embeddings and LSTM, we get average f1-score 2.5% lower than the authors do. It is worth mentioning that this setting yields the best results on the task 1 of our replication, the same pattern is seen for the authors’ LSTM. With the fourth model setting, all features and FFNN, we are, on the con-

Experimental Setting	**Class**	**Results from (Haddadan)**	**Our results, replicated**	**Our results, tuned**
Majority Baseline	Claim	0.68		
	Premise	0.00		
	Average	0.35		
Tf-idf + SVM	Claim	0.685	0.644	0.671
	Premise	0.599	0.574	**0.610**
	Average	0.643	0.610	0.641
All features + SVM	Claim	0.717	0.694	
	Premise	0.581	0.552	
	Average	0.651	0.624	
Word embeddings + LSTM network	Claim	0.829	0.658	
	Premise	0.710	0.654	
	Average	0.673	0.656	
All features + FFNN	Claim	0.667	0.666	
	Premise	0.611	**0.634**	
	Average	0.640	**0.651**	

Table 4: f1-scores for task 2, classification of claims and premises

trary, getting higher average f1-score and higher f1-scores for both classes. Since neither of the neural models was in detail covered in the original paper, these score variations might stem from the model architectures and learning parameters.

4.2 Task 2: Claims and Premises

All four experiments for the task 2, classification of premises and claims, have also predictably shown significant baseline outperformance. Reproduction results for the first experimental setting were slightly lower (3%) with the identical to authors SVM parameters, but hyperparameter tuning led to the same average f1-score and a higher f1-score for the minority class of premises (0.610 against the authors' 0.599). The best parameters were C equal to 1 and gamma equal to 0.1. The second setting with features and SVM model as well as LSTM setting showed, like in the previous task, slightly lower performance than the authors, up to 2.7 % on the average f1-score. The fact of it happening for these two settings again, supports the reasons stated in the task 1 results analysis. Also, adding linguistic features to the input again result in improved results. This, on the one hand, might confirm that engineered features create relevant weights for the model, as suggested in the original paper and described in Features section. On the other hand, experiments with features also use rbf kernel against the experiments with only unigrams and linear kernal SVM. Since rbf kernel is generally considered to be more "powerful", such an improvement might stem from it.

4.3 Reproduction Bottom Line

All in all, we consider the reproduction goal as successfully achieved. Even though for some models our scores are slightly lower, they are still in the acceptable interval, being never lower than 3% on average f1-scores. The substantial part of reproduction success is seen in models which outperform those of authors (up to 6% for the first experiment). Such result variations can easily be due to randomization events, differences in the used kinds or versions of software frameworks, hyperparameter inconsistencies. To compare the model settings, it is important to mention that LSTM model shows higher performance than the models with engineered features. As mentioned in (Haddadan et al., 2019), it is an important observation since the input dimension for LSTM model is significantly smaller and less computationally expensive while more robust.

5 Linguistic Error Analysis

Upon the predictions of classification models, we perform a linguistic error analysis. We only consider results of the task 1 for the error analysis since the misclassification rate is relatively low and misclassified sentences could contain the erroneous patterns. The performance of claims and premises classification, even though outperform-ing the baseline, remains rather low for practical applications and unreasonable for error analysis.

We analyze the errors of two models: 1) LSTM model as it is the best performing one and 2) SVM model based on a full feature-set. We motivate the second choice by the interest to compare patterns of errors produced by a model with solely semantic input (LSTM) to the model with lexical, morphological and syntactic features. For both models, we randomly extract 30 false positive sentences (sentences that were classified as argumentative, but are not in reality), and 30 false negatives sentences (sentences that were classified as non-argumentative but do contain an argument component) from the predicted lists. In total, our analysis is based on 120 misclassified sentences.

We observe that both models share similar error patterns for false positives, and it is worth mentioning that we noticed the models misclassify the same sentences. As in the original paper, we also observed that short sentences, containing argument indicators (such as 'because', 'but', 'so', etc.), were labeled by the models as argumentative. For example, the sentence: 'But let's sort of keep this out of politics because it's pretty dicey right now', was classified as an argumentative sentence by both LSTM and SVM models. However, the context shows that it is non-argumentative. Another pattern, which also was mentioned by the authors, is found in sentences formulated like an argument (e.g., 'I think we need to go back and take a look.') and verbalized in non-argumentative context. Both models misclassified sentences containing negations: 'I can't fix the statistics.', adjectives: 'It's an awesome responsibility'. The reason for the confusion is that candidates often use negative clauses or words with negative polarity in order to contract their opponents, and adjectives are used to stress the correctness of their justifications, give them an assessing value. Some sentences included proper nouns which are usually used by politicians as statistics to make their justification sound more persuasive. As for the false negatives, the performance of the models is slightly different. Incomplete sentences[12] (for example, "No discrimination."), as well as conditional sentences (e.g., "If you don't like what I did, you should have changed the laws'.') are labeled as non-argumentative, whereas the context implies them to contain an argument component. However, not all patterns observed appear in both of the models. Compared to the LSTM model results, several SVM model outputs do not recognise questions (for instance, 'How do we pay for it?') to be sentences with an argument component.

6 Conclusion

In this project, we addressed the problem of argument mining and looked at it through two classification tasks. The first task implied argument detection, while the second task was dedicated to classification of claims and premises. Both tasks were carried out in the paradigm of a full reproduction of a study by (Haddadan et al., 2019). Starting the reproduction process, we were at disposal of the original data. At the same time, we lacked the source code for the research, but we managed to develop all the necessary software described by the authors from scratch. As for the reproduction results, we claim to have reached comparable results to the authors' and, thus, to have reproduced the research.

Our contribution, or adjustment, to the original research, lies in the following. First, the tools used to engineer the features were different from those the authors used. Secondly, we slightly enriched the feature set by adding first-person singular and plural pronouns as two Boolean features. Thirdly, on the experiments with SVM models, we tuned the hyperparameters, which resulted in discovering more robust parameter settings than the authors reported. Finally, we repeated linguistic error analysis and got confirming and additional observations to those of authors.

Future expansion of the project could be directed at trying out other models, especially transfer learning (e.g., BERT and follow-up) for the same classification tasks. Observing that neural model using only word embeddings is still beating the models with engineered features, hints that transfer learning could also perform well on both tasks. In addition to that, other sub-tasks could be tested on the existing corpus. For example, predicting relations between claims and premises (supportive or opposing) or classifying the types of premise components (e.g., statistics, anecdotes, testimonies, etc.). The latter task should be probably carried out using shorter text entries than a sentence in order to avoid "noisy" side information unrelated to the premise component.

[12] sentences without subject or predicate or both

References

Monya Baker. 2016. 1,500 scientists lift the lid on reproducibility. Nature 533, 452–454. https://doi.org/10.1038/533452a

K. Bretonnel Cohen, Jingbo Xia, Pierre Zweigenbaum, Tiffany J. Callahan, Orin Hargraves, Foster Goss, Nancy Ide, Aurelie Névéol, Cyril Grouin, and Lawrence E. Hunter. 2018. Three Dimensions of Reproducibility in Natural Language Processing. LREC: International Conference on Language Resources Evaluation : [proceedings]. International Conference on Language Resources Evaluation, 156–165.

van Eemeren, Frans H., Rob Grootendorst, and Francisca Snoeck Henkemans. 1996. Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments. Routledge, Taylor Francis Group.

Debopam Das, Tatjana Scheffler, Peter Bourgonje, and Manfred Stede. 2018. Constructing a Lexicon of English Discourse Connectives. In Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, pages 360–365, Melbourne, Australia. Association for Computational Linguistics.

Shohreh Haddadan, Elena Cabrio, and Serena Villata. 2019. Yes, we can! Mining Arguments in 50 Years of US Presidential Campaign Debates. Proceedings Of The 57Th Annual Meeting Of The Association For Computational Linguistics. https://doi.org/10.18653/v1/p19-1463

Sepp Hochreiter, and Jürgen Schmidhuber. 1997. Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hervé Jégou and Tomás Mikolov. 2016. FastText.zip: Compressing text classification models. arXiv preprint arXiv:1612.03651.

Marco Lippi and Paolo Torroni. 2016. Argument mining from speech: Detecting claims in political debates. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, pages 2979–2985.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Diederik P. Kingma, Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. 2015. Argument Extraction from News. In Proceedings of the 2nd Workshop on Argumentation Mining, pages 56–66, Denver, CO. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Aspect-Controlled Neural Argument Generation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 380–396, Online. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014. Annotating argument components and relations in persuasive essays. In COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland, pages 1501–1510. ACL.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. Computational Linguistics, 43(3):619–659.

Jacky Visser, Barbara Konat, Rory Duthie, Marcin Koszowy, Katarzyna Budzynska, and Chris Reed. 2019. Argumentation in the 2016 us presidential elections: annotated corpora of television debates and social media reaction. Language Resources and Evaluation.

Prakash Poudyal, Jaromir Savelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. ECHR: Legal Corpus for Argument Mining. In Proceedings of the 7th Workshop on Argument Mining, pages 67–75, Online. Association for Computational Linguistics.