Abstract

Despite the catastrophic effects that climate change is forecasted to have over the next century, the topic still remains a partisan issue in America. The ability to automatically and accurately determine the opinion of a piece of text about climate change could be useful for numerous applications, such as gauging public opinion over time or identifying media outlets that consistently downplay the validity of scientific evidence. Stance detection is a commonly applied classification task in natural language processing that is well suited for developing models to determine the opinion of a piece of text. In this study, I aimed to improve classification accuracy of stance detection performed on a recently curated data set (DeSMOG), containing 2,050 global-warming-related sentences from various news sources. Specifically, I focused on engineering new features and tested the hypothesis that adding sentiment scores as a feature would improve classification accuracy. Using two different methods for sentiment scoring, I show that neither were able to improve performance above a baseline measure, which uses only the pooled output embedding from BERT-based-uncased (0.71 accuracy and 0.70 macro-F1). I discuss possible reasons for this null result as well as broader connections between stance detection and sentiment analysis.

Introduction

Support for climate change policy is becoming an increasingly partisan issue in the United States over recent years (Pew Research Center, 2020). Accordingly, multiple studies have investigated the effects of using different methods to communicate climate-change-related information to the public (Bolsen et al., 2017, Bolsen et al., 2019, Palm et al., 2020). A recent study that investigated differences in word choice between various news sources developed a supervised model to perform stance detection (Luo et al., 2020). The authors used the model to label sentences from news articles as *agree, disagree,* or *neutral* with respect to a single target statement: “Climate change/global warming is a serious concern.” The authors find that a BERT model performed best at this classification task with 0.75 accuracy and 0.73 macro-F1 score. In this project, I aimed to improve upon the performance of BERT by using sentiment scores as an additional feature (along with the pooled output embedding from BERT) to perform the same stance classification task on the same data set as in Luo et al., 2020.

Background

Stance detection is a natural language processing (NLP) task that aims to determine the opinion, or "stance," of a piece of text. Specifically, stance detection is most commonly performed by classifying the relationship between a candidate piece of text and a target statement as falling into one of three classes: {*agree, disagree, neutral*}. This task has received increased attention in recent years as concerns grow over the spread of misinformation, popularly known as "fake news" (Lazer et al., 2018). Therefore, much of the recent literature on stance detection has aimed to identify misinformation (eg. stances that disagree with factual target statements) in text collected from news sources, including conventional news websites as well as social media platforms. In particular, two of the most popular stance detection competitions were the SemEval-2016 Task 6 (Mohammad et al., 2016), which used a data set consisting of tweets from Twitter, and the 2017 Fake News Challenge (FNC) Stage 1 (Hanselowski et al., 2018), which used text data collected from news articles. Notably, in contrast to these data sets that use numerous pairs of candidate and target statements (eg. different headline and body text pairs for the FNC-1 data set), all examples in the data set used in this study were labeled relative to a single target statement: "Climate change/global warming is a serious concern."

In addition to its potential for detecting misinformation, an effective stance detection model can have various other applications for downstream analyses, such as gauging public opinion on a specific topic or researching how supporters of different sides of an argument may present information differently. An example of the latter: Luo et al., 2020 developed and used a stance detection model to label stances of over 500,000 opinions related to global warming scraped from news articles. The authors used the labels assigned by the stance detection model to divide the data set into two groups, "global-warming-accepting media" and "global-warming-skeptic media," and then studied differences in word choice between these two groups; among other findings, they discovered that global-warming-skeptic sources tend to more often use words that cast "opponent-doubt," "(eg. pretend, claim; inaccurate, alleged)." Critically, the authors were only able to perform the primary analyses of their study because of the automatic labeling made possible by the stance detection model. Therefore, just as with detecting misinformation, many future applications of stance detection models would benefit greatly from the model itself being as accurate and precise as possible; i.e., using a more accurate model would give more confidence in future analyses that are built upon the assumption that the model accurately labeled the stance of the text being analyzed. So the primary focus of this study is to improve the classification performance (measured by accuracy and macro-F1 score) of the model developed by Luo et al., 2020 by training on the same data set released by the authors.

In particular, I sought to improve stance detection performance by testing the hypothesis that authors who are more opinionated on climate change also write with stronger sentiment. If this were true, then perhaps the sentiment of a piece of text would add useful information for a model to distinguish between different stances. Therefore, a major portion of my analysis focused on deriving sentiment scores for text data and using the scores as features to improve stance detection performance. This idea of using sentiment-related information to improve stance detection has been tested before, but not specifically with climate-change-related text. Many previous studies used the SemEval-2016 Twitter data set and tried a variety of approaches to incorporate sentiment-related information, including using sentiment lexicons to derive features for an SVM classifier (Sobhani et al., 2016), building a "joint neural network model" (Sun et al., 2019), and using a multi-task learning architecture that performs both sentiment classification and stance detection (Li & Caragea, 2019). Overall, these studies reported that incorporating sentiment information into stance detection improved performance beyond state-of-the-art (SOTA) models for the SemEval-2016 data set. Therefore, it seemed to be worthwhile to try using sentiment scores as features for stance detection on the DeSMOG data set by Luo et al., 2020.

For deciding which general model architecture to use, the stance-detection literature seems to clearly point to transformers, specifically BERT, as performing best. Although some of the most popular models used for stance detection historically were SVMs and RNNs (Rakholia & Bhargava, 2016, Hanselowski et al., 2018, see Kucuk & Can, 2020 for review), a recent study focused on model comparison for stance detection showed that BERT outperformed previous SOTA models (Ghosh et al., 2020). Importantly, for the data set I am using in this study, DeSMOG, the authors also reported that a BERT-base model performed best at stance detection with 0.75 accuracy and 0.73 macro-F1 score. Therefore, in this study I compared different variants of BERT-based models with and without sentiment features. Specifically, I making primarily two contributions to previous work:

1. Experimenting with 3 new features used for classification
   (1) Sentiment scores from using SentiWordNet
   (2) Sentiment scores derived from BERT sentiment probabilities
   (3) Number of tokens
2. Detecting statistical significance of differences in these features between classes

Results

Data

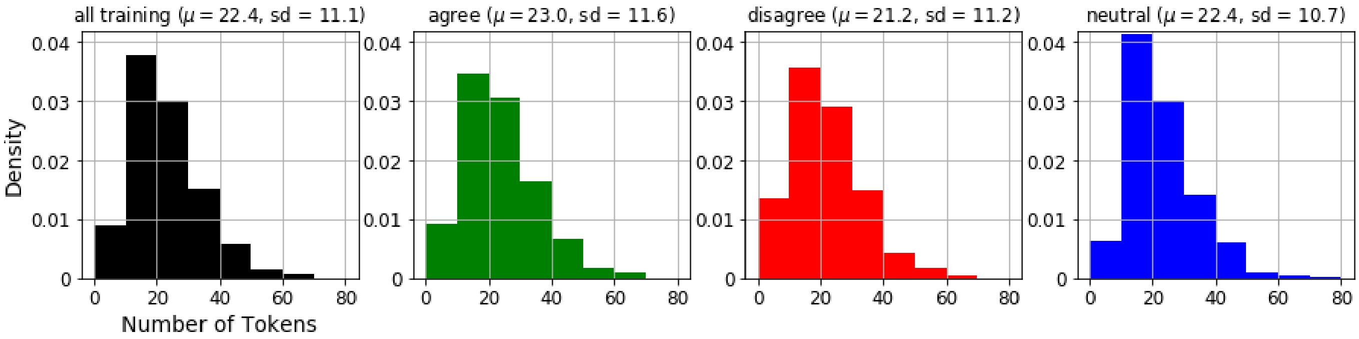
The data used for all analyses were obtained from the public github repository of Luo et al., 2020. The data set contains 2,050 global-warming-related sentences from various news websites. For modeling, 200 sentences were left out to use as a held-out test set (stratified by label and political leaning of the news source), and the remaining 1,850 sentences were used for training. Some example sentences and corresponding labels are displayed in Table 1.

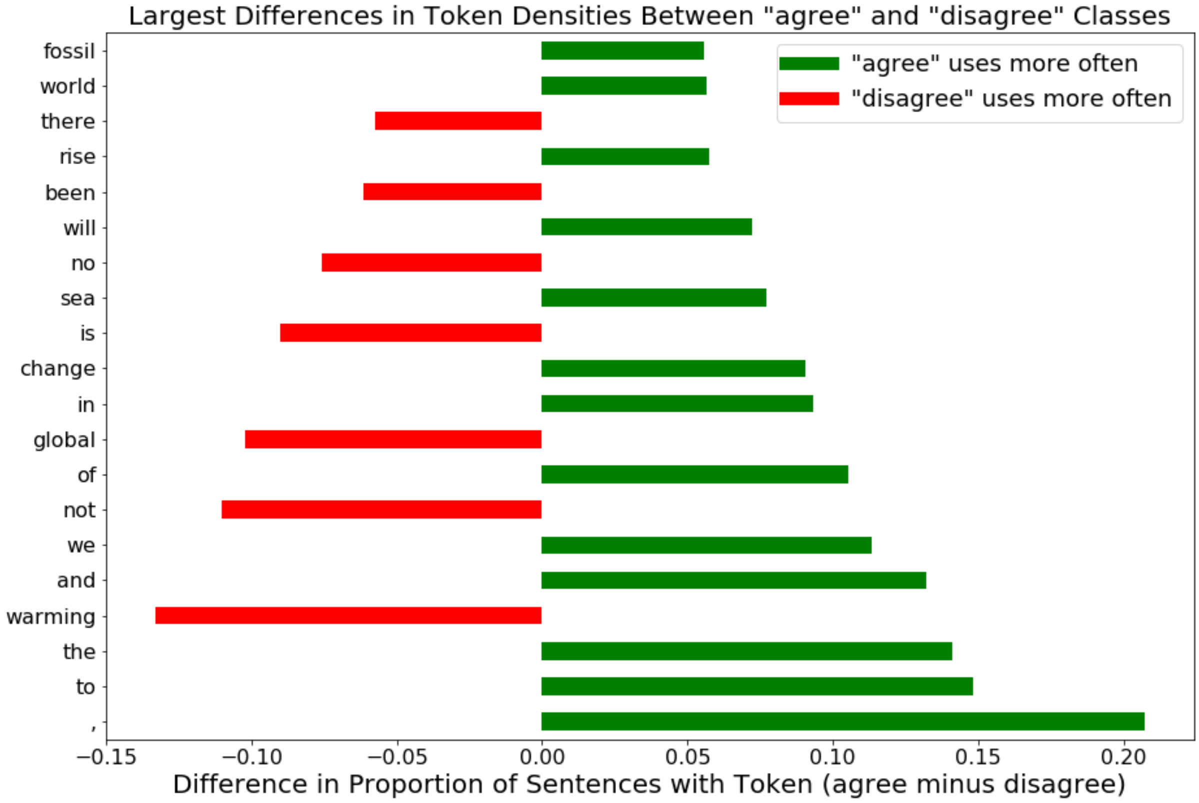
Exploratory Data Analysis

I first looked for class imbalance in the data set by checking the proportion of examples in each class. Indeed, there is a relatively large imbalance with *disagree* stances occurring around half as often as the other two classes {*agree*: 0.37, *disagree*: 0.20, *neutral*: 0.43}. According to previous work on this data set, resampling techniques tended to perform poorly compared to using class weights (Luo et al., 2020); therefore in my baseline model, I also used class weights to adjust the loss function and try to account for effects of the class imbalance.

Next, I looked at the distribution of token lengths for the training set to get an idea of how much text is in the sentences and if there are differences between classes (Fig. 1, *Top*). The mean token length across all training examples was 22.4 ± 11.1, and sentences with an *agree* stance were significantly longer than those with a *disagree* stance (Wilcoxon rank-sum, p = 0.0035). In addition to overall length of sentences, I also examined the occurrences of specific tokens and how their frequencies differed between classes. Fig. 1, *Bottom* shows the tokens that had the largest differences in occurring between *agree* and *disagree* sentences. The token that was most commonly used more in *agree* sentences than *disagree* sentences was a comma (used ~20% more often), which makes intuitive sense based on the previous analysis that identified that *agree* sentences tend to be longer than *disagree* sentences; the use of more commas may also suggest the use of more complicated sentence structures with multiple clauses. Although some other tokens in the list were interesting (eg. *disagree* uses “global” and “warming” more often and *agree* uses “fossil,” “sea,” “rise,” and “we” more often), I did not explicitly use any of these insights for other analyses in this study due to time constraints. Nonetheless, future work may benefit from applying this analysis as well as its extensions, such as using an n-gram approach as opposed to only using single tokens (i.e. unigrams).

Overall, from these exploratory analyses, I decided to implement the following for stance detection classification: (1) weight the loss function using class weights and (2) use “number of tokens” as an additional feature due to its significant difference between classes.

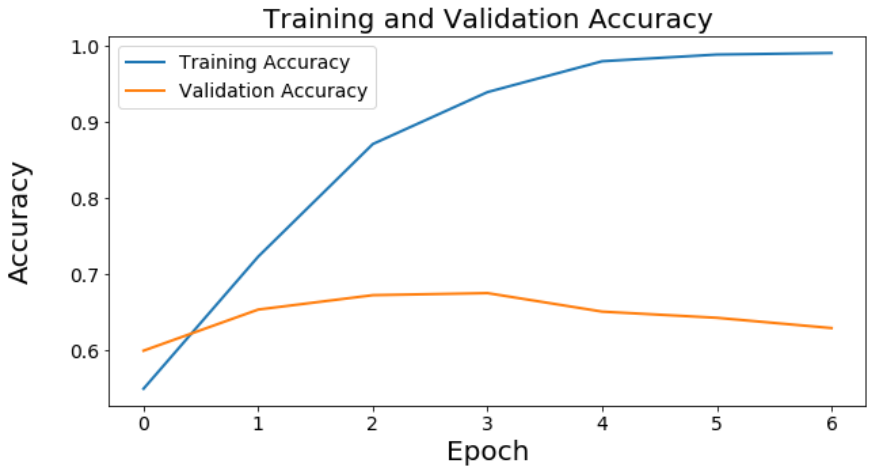




Baseline Stance Detection Model

As a baseline model, I used a neural network with the following architecture (in order of first hidden layer to output): (1) pre-trained BERT-base-uncased model from the Huggingface Transformers library; (2) a single fully-connected dense layer with 256 neurons; (3) a softmax output layer with 3 units (i.e. the 3 classes: *agree, disagree, neutral*). The model was trained using 5-fold cross validation on 1,850 sentences and achieved validation performance of 0.67 ± 0.03 accuracy (mean ± std across folds) and 0.65 ± 0.03 macro-F1. These scores did not match the best performing model from Luo et al., 2020 (0.75 accuracy and 0.73 macro-F1); however, they did come relatively close and outperformed all other linear models implemented by Luo et al, 2020, thus reaffirming the remarkable capabilities of BERT for transfer learning.

The most apparent issue with the baseline model is overfitting to the training data with training accuracy approaching 1 and validation accuracy increasing in initial epochs but then peaking at ~0.65 accuracy usually by the fourth epoch (Fig. 2). To combat this overfitting, I ran a series of experiments including the following: varying the dropout rate of the dense layer, varying the number of neurons in the dense layer, adding an additional dense layer with dropout, lowering the number of epochs, and changing the proportions of the train/dev split. However, none of these methods proved to be very effective at increasing validation accuracy, and the main purpose of this study is to test the use of sentiment information for stance detection; therefor, I proceeded to use the same hyperparameters as in the baseline model, acknowledging that overfitting due to model architecture and/or hyperparameter values may be a confound for interpreting results.



Engineering Sentiment Features

To test the main hypothesis that sentiment scores can be used to improve the performance of BERT for stance detection, I used two different methods to engineer two separate features that aim to capture sentiment information.

The first method uses a lexicon-based approach by using SentiWordNet to assign sentiment scores. SentiWordNet comprises a list of annotations for each WordNet synset according to the degree of positivity, negativity, and neutrality (each one on a scale of 0-1 and all three scores sum to 1). SentiWordNet was implemented using the nltk library. For each example, a single scalar value was calculated to capture the overall sentiment of the sentence (Fig. 3, *Top*).

The second sentiment feature was derived from the probabilities outputted from the default “sentiment-analysis” pipeline of the Transformers library. This implementation uses a pre-trained distilBERT model to classify the sentiment of a piece of text as one of two classes: {*positive, negative*} along with the probability predicted for each class. These probabilities were calculated for each example and then transformed (see *Methods*) to use as a second sentiment feature for stance detection classification (Fig. 3, *Bottom*).

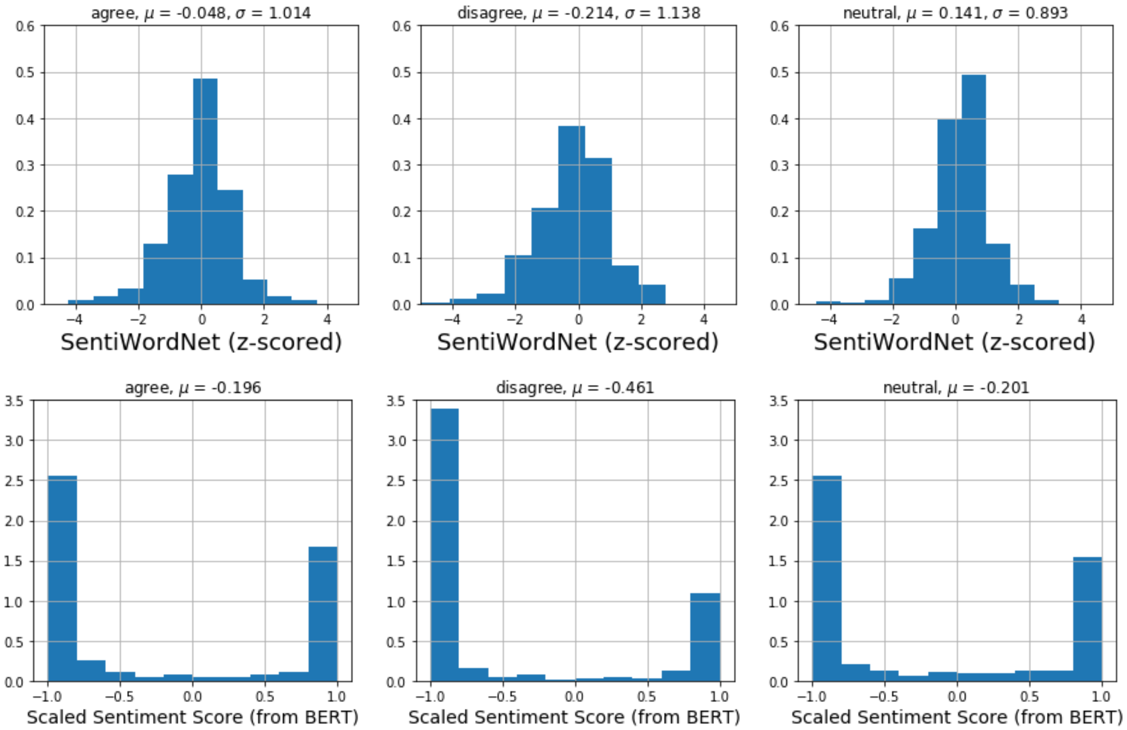


Fig. 3 shows the distributions for both the SentiWordNet-based sentiment scores (swnet-sent) and the sentiment scores derived from distilBERT probabilities (bert-sent). Though the shapes of the distributions are clearly different, both features show that *disagree* sentences tend to have lower sentiment scores than *agree* sentences. These differences are statistically significant for both features (for swnet-sent: unpaired t-test, $p = 0.011$, for bert-sent: Wilcoxon rank-sum, $p < 1x10^{-9}$), though the effect sizes are only small-moderate: for swnet-sent, Cohen's D = 0.154, for bert-sent, Cohen's D = 0.308.

Adding New Features to Baseline Model

In total, 3 features were engineered: (1) number of tokens, (2) swnet-sent, and (3) bert-sent. The latter two were the main variables of interest while "number of tokens" was mainly used for exploratory purposes and to control for adding a scalar value onto the BERT embedding. Specifically, the effects of each feature on stance detection performance was tested by using the same architecture and hyperparameters as in the baseline model, and the only change was concatenating the new feature(s) onto the 768-dimensional pooled output embedding from the pretrained BERT (eg. if a single feature is being tested, a 769-dimensional vector is inputted into the 256-neuron dense layer and then passed to the softmax output layer). In total, I ran 4 experiments, each one consisting of a 5-fold cross validation

Model	Accuracy	Macro-F1
Majority Class	0.43	0.17
BERT (baseline)	0.67 ± 0.02	0.64 ± 0.03
BERT With ONLY SentiWordNet	0.66 ± 0.02	0.65 ± 0.03
BERT With ONLY bert-sentiment	0.69 ± 0.02	0.66 ± 0.03
BERT With 3 Engineered Features	0.67 ± 0.03	0.65 ± 0.04

Error Analysis

Methods

For each example, (1) look up the *first listed* positive and negative SentiWordNet score of each token (tokens not found in the SentiWordNet lexicon were ignored); (2) subtract the negative score from the positive score for each token; (3) calculate the mean of the resulting scores across all tokens, which results in a single scalar sentiment score for each example. All scores across all examples were z-scored to scale the values so that they are comparable to the embedding output of BERT. A key limitation of this method currently is using the *first listed* scores available from SentiWordNet, which does not take into account the part of speech of each token in the specific example. Due to time constraints, I was not able to implement a part of speech tagger in order to assign more accurate SentiWordNet scores, but future work should

With a growing abundance of news outlets, from conventional news websites to social media platforms, there is also an overwhelming amount of text data being generated every moment. Developing systems to automatically extract useful information from this text data would have numerous applications.

Importantly, although sentiment analysis is often framed as a classification task (eg. a piece of text is classified as {*positive, negative, neutral*}), in this study I sought to use a continuous mea

A popular area of research has been

The authors found that both “global-warming-accepting media” and “global-warming-skeptic media” used similar linguistic devices to promote self-affirmation as well as opponent-doubt, but global-warming-skeptic media tended to show relatively more opponent-doubt.

- Luo et al. used a stance detection model to mass label news data and then look at differences between word choice of the labeled articles
    - But to be able to do this and all other tasks better, developing a better performing model is critical, which is the topic of this paper

References

Pew Research Center. 2020.As economic concerns re-cede, environmental protection rises on the public'spolicy agenda