

Using Sentiment Scores for Stance Detection Of Climate-Change Text

Michael Zhu

UC Berkeley

michael.zhu@berkeley.edu

W266 Spring 2021

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 or identifying media outlets that consistently reject or downplay the validity of scientific evidence. Stance detection is a popular 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 sought to improve stance detection performance 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 to BERT embeddings would improve classification accuracy and macro-F1 score. Using two different methods for sentiment scoring, I find that neither were able to reliably improve performance above a baseline model, which used only the pooled output embedding from BERT-base-uncased (0.70 accuracy and 0.71 macro-F1). I discuss possible reasons for this null result as well as broader connections between stance detection and sentiment analysis.

1 Introduction

Support for climate change policy is becoming an increasingly partisan issue in the United States over recent years [1]. Accordingly, many studies have investigated the effects of using different methods to communicate climate-change-related information to the public [2, 3, 4]. 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 stance detection on the same data set as in Luo et al., 2020.

2 Background and Methods

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” [6]. Therefore, much of the recent literature on stance detection has aimed to identify misinformation 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 [7], composed of tweets, and the 2017 Fake News Challenge (FNC) Stage 1 [8], composed of news articles. Notably, in contrast to these data sets that use numerous pairs of candidate and target statements, 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. 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 [9], building a “joint neural network model” [10], and using a multi-task learning architecture that performs both sentiment classification and stance detection [11]. 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, I tried 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 [8, 12, and see 13 for review], a recent study focused on model comparison for stance detection showed that BERT outperformed previous SOTA models [14]. 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 make primarily two contributions to previous work:**

- 1. Experimenting with 3 new features used for stance detection on DeSMOG data**
 - (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**

3 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.

Target Statement: “Climate change/global warming is a serious concern.”

Label	Example Sentence
neutral	If carbon dioxide emissions continue to rise beyond 2020, or even remain level, the temperature goals set in Paris become almost unattainable.
disagree	The study is one more example that you can get any answer you want when the thermometer data errors are larger than the global warming signal you are looking for.
agree	Millions more people around the world are threatened by river floods in coming decades due to climate change.

neutral	45 % of the general public view perceived global warming as caused by humans.
agree	Two billion people may be displaced by rising sea levels by the turn of the next century.

Table 1. Target statement and 5 example sentences with labels from the DeSMOG data set.

Exploratory Data Analysis

The data showed a relatively large class 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 account for effect 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. 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 (Appendix A).

Overall, from these exploratory analyses, I decided to implement the following for stance detection classification: (1) use class weights in the loss function 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 \pm 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. 1). 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; therefore, 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. 2, *Top Row*).

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 *Appendix Methods*) to use as a second sentiment feature for stance detection classification (Fig. 2, *Bottom Row*).

Though the shapes of the distributions for these two sentiment features are starkly different (Fig. 3), 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, $p < 1 \times 10^{-9}$), 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; the only change was

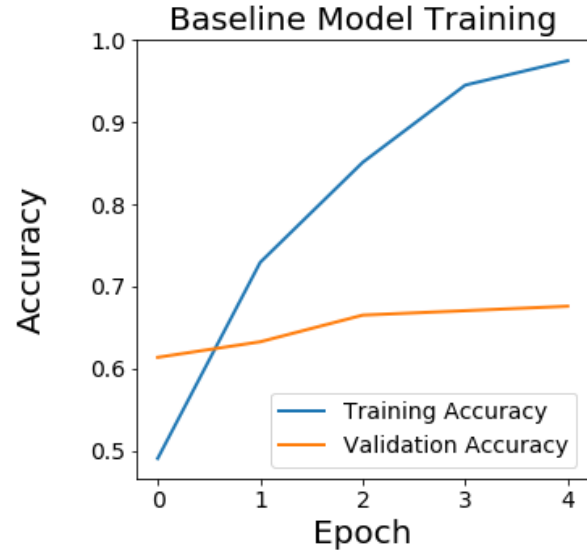


Figure 1. Training and validation accuracy of the baseline model using BERT-base-uncased.

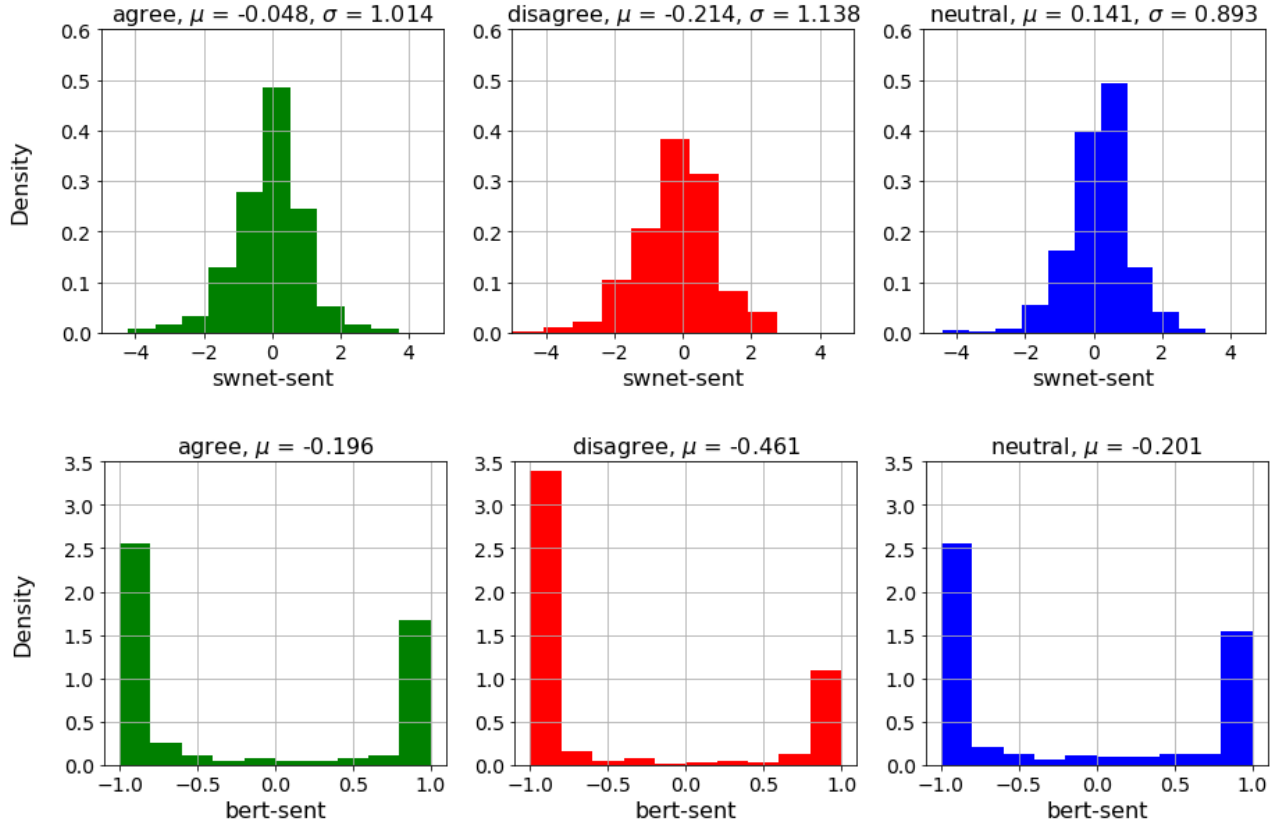


Figure 3. Distributions of the two sentiment features, *swnet-sent* (top row) and *bert-sent* (bottom row) separated by class.

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: three of them used each of the engineered features individually and the fourth used all three engineered features together. Table 2 shows the results of all models.

Overall, all models performed within a relatively narrow range of each other and were far better than majority-class prediction. The model using only *bert-sent* performed best on validation data by a small margin with 0.69 ± 0.02 (std) accuracy and 0.66 ± 0.03 macro-F1. However, qualitatively, this performance did not seem reliably better than the baseline model. Running more cross-validation training sessions would be necessary to determine whether the difference in evaluation metrics is significant. Evaluating predictions on the held-out test set using the best performing model achieved 0.71 accuracy and 0.70 macro-F1, which falls short of the 0.75 accuracy and 0.73 macro-F1 previously achieved by Luo et al., 2020 (who did not use any explicit sentiment features). Based on these results, using sentiment features did not have a large impact on accuracy or macro-f1 for stance detection on the DeSMOG data set.

Model	Val_Accuracy	Macro-F1
Majority Class	0.43	0.17
BERT (baseline)	0.67 ± 0.02	0.64 ± 0.03
BERT With ONLY num_tokens	0.64 ± 0.03	0.65 ± 0.03
BERT With ONLY swnet-sent	0.66 ± 0.02	0.65 ± 0.03
BERT With ONLY bert-sent	0.69 ± 0.02	0.66 ± 0.03
BERT With 3 Engineered Features	0.67 ± 0.03	0.65 ± 0.04

Table 2. Summary of model performance: mean validation accuracy and mean macro-F1 score across 5-fold cross-validation for each model.

Error Analysis

To better understand the model’s classification patterns and provide insight for future analyses, I examined the confusion matrix of the best performing model evaluated on the held-out test set (Fig. 3). By far the most often misclassification (11% of all test examples) was predicting *neutral* when the true label was *agree*. This result corroborates with findings reported in Luo et al., 2020, who found that F1 score for the *agree* class was the lowest out of the three classes (i.e. likely due to more false negatives). Qualitatively, from looking at some of the misclassified examples it seems like the model doesn’t effectively learn relatively abstract relationships between climate change and related entities (eg. “We can expect the Arctic to be ice-free in summer within 20 years” and “Coal would have to be phased out even before the Paris Agreement to combat climate change.” were both labeled *neutral*). In these examples, relationships like climate change and ice melting or coal usage are not obvious and require a certain degree of previous knowledge and abstraction. Therefore, (as with likely most NLP tasks) future models could benefit greatly from developing methods to more effectively learn abstract relationships based only on text data.

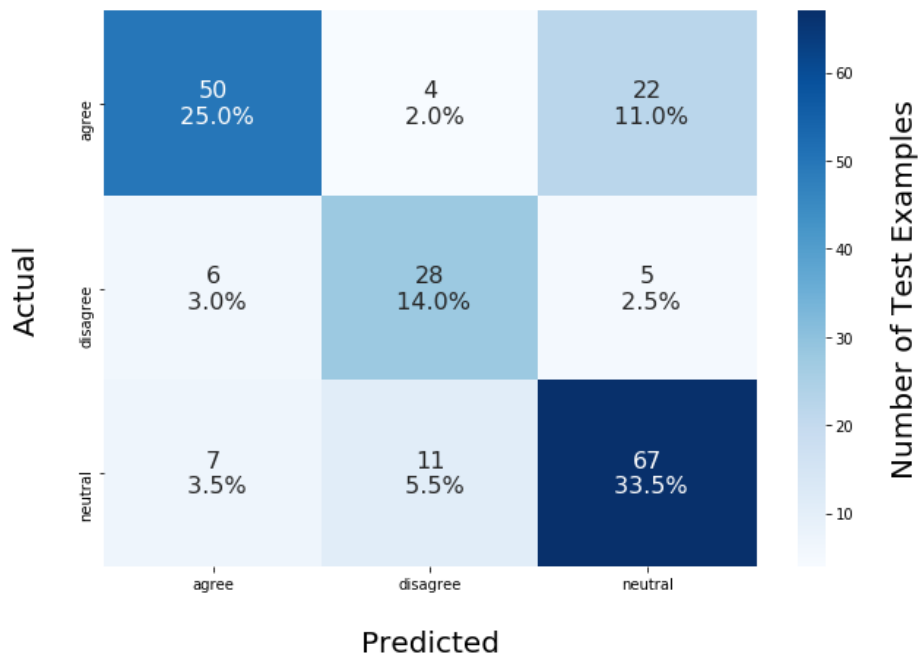


Figure 3. Confusion matrix of predictions on the held-out test set (using the model that performed best based on mean 5-fold cross-validation accuracy).

Discussion and Conclusion

My initial hypothesis was that authors who are more opinionated on climate change also write with stronger sentiment. Overall, I find mixed results with respect to this hypothesis: opinions that are labeled *disagree* do have significantly lower sentiment scores than opinions labeled *agree*, however, using sentiment scores for classification was not able to improve performance on stance detection. Furthermore, there are two important limitations of this study: (1) models tend to overfit to the training data and (2) the engineered sentiment features use relatively crude methods to capture sentiment information; simply concatenating a scalar sentiment score onto a 768-dimensional embedding may make any information contained in the sentiment score too sparse for the model to potentially extract useful information for separating between classes. Therefore, future analyses may be able to overcome these limitations and develop more effective methods to use sentiment information.

References:

1. Pew Research Center. 2020. As economic concerns re-cede, environmental protection rises on the public's policy agenda.
2. Bolsen, T., Shapiro, M. A., Bolsen, T., & Shapiro, M. A. (2017). Strategic Framing and Persuasive Messaging to Influence Climate Change Perceptions and Decisions. In Oxford Research Encyclopedia of Climate Science. Oxford University Press.
3. Bolsen, T., Palm, R., & Kingsland, J. T. (2019). The Impact of Message Source on the Effectiveness of Communications About Climate Change. *Science Communication*, 41(4), 464–487.
4. Palm, R., Bolsen, T., & Kingsland, J. T. (2020). “Don’t Tell Me What to Do”: Resistance to Climate Change Messages Suggesting Behavior Changes. *Weather, Climate, and Society*, 12(4), 827–835.
5. Luo, Y., Card, D., & Jurafsky, D. (2020). DeSMOG: Detecting Stance in Media On Global Warming. arXiv preprint arXiv:2010.15149, 2020.
6. Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., & Zittrain, J. L. (2018). The science of fake news: Addressing fake news requires a multidisciplinary effort. *Science*, 359(6380), 1094–1096.

7. Mohammad, S. M., Kiritchenko, S., Sobhani, P., Zhu, X., & Cherry, C. (2016). SemEval-2016 Task 6: Detecting Stance in Tweets.
8. Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., & Gurevych, I. (2018). A Retrospective Analysis of the Fake News Challenge Stance Detection Task.
9. Sobhani, P., Mohammad, S., & Kiritchenko, S. (2016). Detecting stance in tweets and analyzing its interaction with sentiment. In Proceedings of the fifth joint conference on lexical and computational semantics, pp. 159–169.
10. Sun, Q., Wang, Z., Li, S., Zhu, Q., & Zhou, G. (2019). Stance detection via sentiment information and neural network model. *Frontiers of Computer Science*, 13(1), 127–138.
11. Y. Li, and C. Caragea, “Multi-Task Stance Detection with Sentiment and Stance Lexicons,” In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6299–6305, 2019
12. S. B. Neel Rakholia, ““is it true?” - deep learning for stance detection in news,” tech. rep., Stanford University, 2017.
13. Küçük, D., & Fazli, C. A. N. (2020). Stance detection: A survey. In *ACM Computing Surveys* (Vol. 53, Issue 1). Association for Computing Machinery.
14. Ghosh, S., Singhanian, P., Singh, S., Rudra, K., & Ghosh, S. (2020). Stance Detection in Web and Social Media: A Comparative Study.

Appendix A

Fig. S1 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).

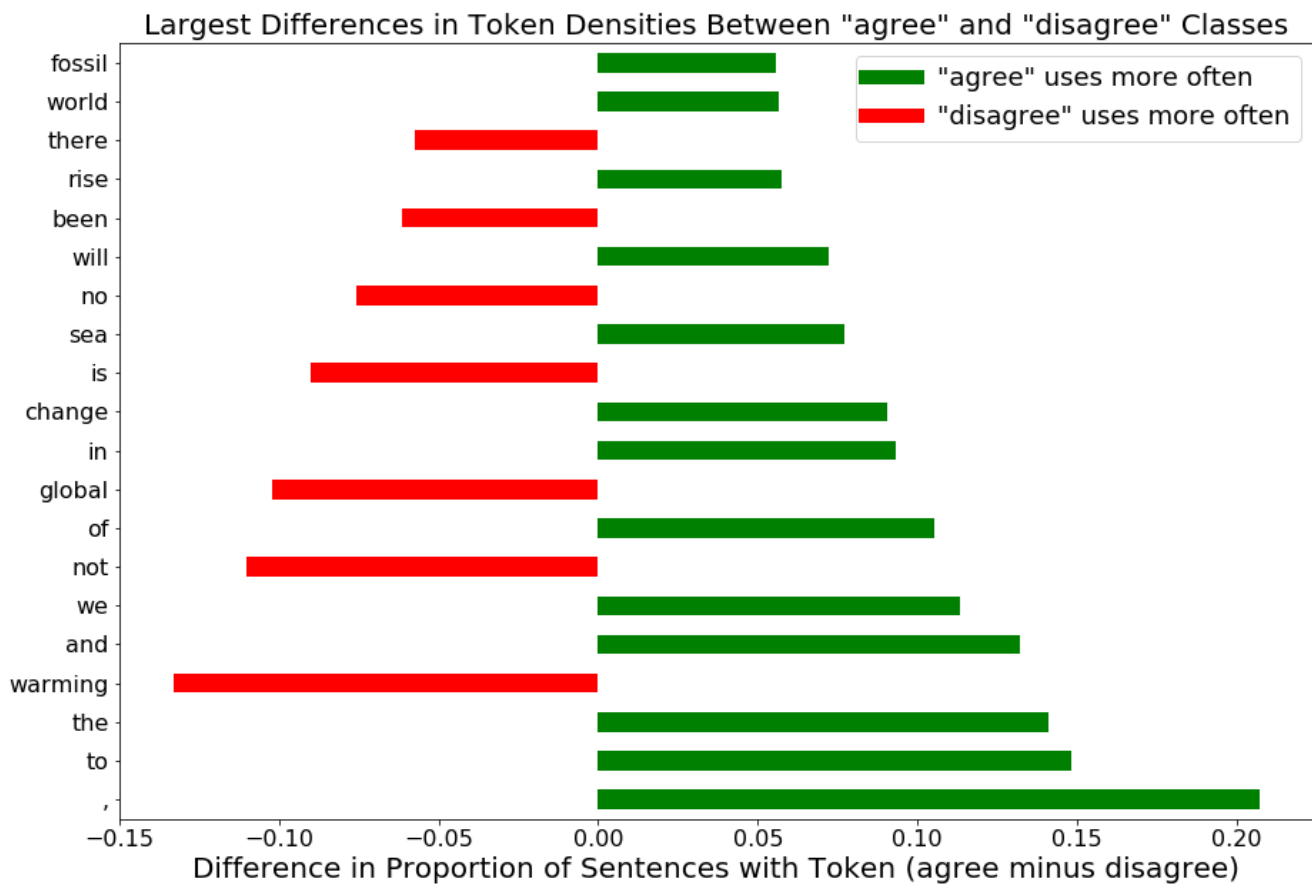


Figure S1. Tokens with the largest differences in frequency used between *agree* and *disagree* examples. Proportions are first calculated within-class and then subtracted.

Appendix Methods

Data

All data were obtained from the public github repository of Luo et al., 2020 (<https://github.com/yiweiluo/DeSMOG>). The data set contains 2,050 global-warming-related sentences scraped from news articles from both “left-leaning” and “right-leaning” media. The most common news sources scraped from include The New York Times, Vox, Mother Jones, Washington Post, The Nation, Guardian (US), Newsmax, Fox, Redstate, Washington Examiner, Washington Times, and Breitbart.

Sentiment Features

SentiWordNet-based scores (swnet-sent) - swnet-sent values were calculated using the following procedure: for each example in the data set, (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 can likely improve the validity of swnet-sent scores by first pos-tagging all tokens for each example sentence individually.

BERT-based scores (bert-sent) - The “sentiment-analysis” pipeline from the Transformers library was used with default settings. Values for the bert-sent feature were calculated using the following procedure: for each example in the data set, (1) use the pipeline to output two probabilities, one for each of the classes: *positive* and *negative*; (2) to capture a “net sentiment” subtract the probability *negative* from probability *positive* (so when the model is more uncertain, the value is closer to 0); (3) change the sign of all examples labeled *negative* by the pipeline to have negative values (i.e. to capture “negative” sentiment).

Modeling

All models used the following hyperparameters: {BERT max token length: 90, learning rate: 2e-5, number of epochs: 7, batch size: 16, number of neurons in dense layer: 256, dropout rate in dense layer: 0.1}. Accuracy and Macro-F1 were chosen as the primary evaluation metrics due to the class imbalance as well as based on previous work using this data set (Luo et al., 2020).

All models were trained and evaluated on a NVIDIA GeForce RTX 2060 (6GB VRAM) using the NVIDIA CUDA interface.