Stance Detection in Tweets

W266 Natural Language Processing - Final Project

Alex Dessouky & Tim Spittle

December 7, 2019

Abstract

TBD - Abstract goes here

1 Introduction

1.1 Background

Stance Detection is a natural language processing (NLP) task that seeks to automatically determine the "stance" of a statement in relation to a "target". The stance the author of that statement can take is either in favor of the target, against the target, or neutral toward the target. A target can be anything: a proposition, an idea, a person, etc. Since the target against which a statement is evaluated can be agnostic of the content of the statement itself (that is, the statement may not address the target at all), it is possible that it cannot be determined from the statement if the author has a stance toward the target of evaluation.

Given an abundance of online discourse on social media, automatic stance detection of social media content is an appealing task with widespread potential applications. In particular, Twitter is a social media platform with an incredibly fast-paced environment where many tweets are shared back-and-forth addressing a variety of topics and issues.

1.2 Objective

The objective of this paper is to explore new methodologies for detecting stance of tweets and attempt to design a classification model that outperforms existing models.

2 Data, Task, & Evaluation

We will rely on an existing, pre-labeled dataset in order to leverage data that has been analyzed by competitor models against which we can compare our results. Specifically, we will use a dataset originally compiled for the International Workshop on Semantic Evaluation (SemEval 2016) where it was used in an evaluation exercise (task 6). (Mohammad et al. 2016b)

This dataset consists of tweet-target pairs annotated for both stance and sentiment. The targets may or may not be referred to in the tweets. While sentiment is shown to be beneficial for stance classification, it was not available to competitors at SemEval and pre-labeled sentiment would not be available in a production environment and is therefore ignored for this classification model. [sobhani-etal-2016-detecting]

The process for collecting and annotating these tweets is detailed in Mohammad et al. (2016a). The authors queried Twitter for topic-specific hashtags to collect a full corpus of tweets related to the intended targets. These targets include:

- Atheism

- Climate Change is a Real Concern
- Feminist Movement
- Hillary Clinton
- Legalization of Abortion

After extracting tweets containing the query hashtags, the authors only kept tweets with that hashtag at the end and then removed those hashtags from the tweet text to exclude obvious clues for the classification task. Note that while this is a necessary step for the NLP task, removal of the query hashtag may actually change the stance of the tweet (e.g. if the hashtag was a negation of the preceding text).

Tweets were then annotated for stance using CrowdFlower [http://www.crowdflower.com], a website for crowd-sourced data annotation, where each tweet was annotated by at least eight reviewers. The raw inter-annotater agreement was 73.1%, however a cutoff was applied where only tweets with >60% agreement were used, which decreased the corpus and increased the average agreement to 81.85%. Lastly, these tweets were split into train and test sets chronologically. (See the **Limitations** section for more on how this process may affect the classifcation task)

Model results were evaluated using a modified F1 score which took the macro-average of the "favor" and "against" stance category F1-scores. This method was chosen to treat the "none" stance as a negative class, though it still will affect scores. *[NEED Information Retrieval Source]* For evaluation of the models across targets, the micro-average of F-scores across all tweets, regardless of the topic, was evaluated (referred to as **F-microT**). This was the score used for the competition metric. Alternatively, the macro-average of the F1-scores achieved for each topic can also be calculated (referred to as **F-macroT**).

3 Related Work

The baseline model for the SemEval-2016 Task 6 competition was a linear-kernel Support Vector Machine (SVM) classifier, with 3 different specifications and features sets used:

- **Unigrams**: five SVM classifiers (one per target) trained using unigram features
- **N-grams**: [*best performer*] five SVM classifiers (one per target) trained using word n-grams of 1-, 2-, and 3-gram length and character n-grams of 2-, 3-, 4-, and 5-gram length
- **N-grams-combined**: one SVM classifier trained on the combined data for all targets, using the say n-gram features as described above

SVM was chosen as it is effective on text categorization tasks and robust on large feature spaces. Sobhani, Mohammad, and Kiritchenko (2016) expanded on this n-gram model to also include SVM models trained using word embedding and sentiment lexicon features. Note that the word embeddings in this model were trained using the "full corpus" of nearly 2 million tweets extracted by the authors that compiled the original dataset using the query hashtags. These tweets were not provided with the data, therefore the strength of this model is artificially inflated by the context the word embeddings were able to capture by virtue of being trained with exclusive data that aligns precisely with the train and test labeled data

Several of the competing teams within the SemEval-2016 Task 6 competition took a 'neural network' approach to the challenge. [Zarella], the winner of the competition, leveraged transfer learning to perform the classification task. The team sampled 200 million + relevant Tweets to create an auxiliary 'hashtag prediction task', which was used as a projection into a LSTM-RNN layer. [Wei] and [Vijayaraghavan] both implemented convolutional neural networks, however, did not achieve the same level of success as [Zarella].

There were also a handful of teams that took a 'featured-based' modelling approach. [Tutek] used 'lexical features' such as n-grams, average word length, and number of hashtags as inputs to an ensemble classifier rooted inlogistic regression, gradient boosting machines, random forests, and support vector machines. Similarly, [Bohler] used a voting classifier that took input from two multinomial Naïve Bayes classifiers, one trained on word bi-grams while the other was trained on character tri-grams, and a logistic regression classifier trained on GloVe vectors. [Bohler] also experimented with various additional features such as the presence or absence of negation and length of tweets.

Since the time the conference has ended, there have been two additional contributions to this task. Both [Sun] and [Du] leveraged neural attention networks to perform stance classification, with each team achieving higher final F1 scores than the winner of the initial competition. Although several of the competing teams found effective models, none were able to achieve higher final F1 scores than the SVM baseline [Muhammed et al.]. This also holds for [Sun] and [Du]'s work after the conference ended. We will present our model's results in the context of performance against all SemEval-2016 Task 6 competing teams, [Muhammed et al.]'s baseline, and the succeeding work by [Sun] and [Du].

Outside the context of SemEval-2016 conference, the BERT language model from Devlin et al. (2018) has achieved state-of-the-art performance on several common natural language processing tasks including question answering, sentiment analysis, and named entity recognition (see *Figure 1* from Devlin et al. 2018). [Mayfield] applied [Devlin et. al]'s BERT contextualized word embeddings to a stance classification task attempting to predict whether a Wikipedia user preferred to 'Keep' or 'Delete' a specific post based on their comments. [Mayfield]'s results show that BERT out-performed their baseline classifiers, which used GloVe and Bag-of-Words embeddings as inputs.

Further, [Ma] experimented with various infrastructures built on top of BERT outputs for a binary classification task using Twitter data. The overall goal of [Ma]'s experiment was to determine whether a given tweet was 'on-topic' or 'off-topic' in the context of disaster management. [Ma] found that a bi-directional LSTM taking in the full BERT sequential output outperformed the standard BERT pooled output used for classification tasks.

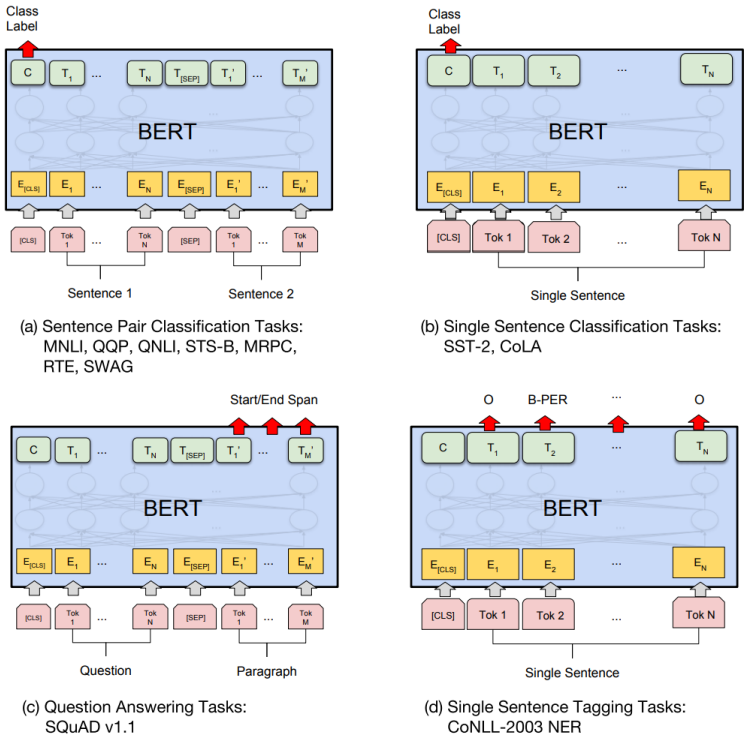

Figure 1: BERT by Task (Source: @devlin2018bert)

4 Model

Next, we describe the model we implemented to approach the SemEval-2016 Task 6 competition. After reviewing the work performed by contributing members of the competition, we hypothesized that a neural network which effectively captures the context of the limited characters and words contained in a tweet would perform well in classifying stance from Twitter data. Words contained in tweets cannot always be taken at face value. Hashtags, sarcasm, irony, and other forms of rhetoric are frequently found in tweets, and we would want our model to effectively capture these nuances since it contains valuable information about the author's stance.

4.1 Infrastructure

Given the state-of-the-art performance BERT has seen across many natural language processing tasks, including stance classification [Mayfield] and a binary classification task leveraging Twitter data [Ma], we elected to leverage BERT for contextualized word embeddings. Rather than using the pooled output captured by BERT's '[CLS]' token, our model connects the full sequential output from BERT to a 128-unit LSTM layer. This infrastructure was motivated by the success of the BERT-LSTM integrated layers in [Ma]'s binary Twitter classification task as well as [Zarella]'s competition winning model that leveraged an auxiliary 'hashtag prediction task' as a projection into a LSTM-RNN layer. The LSTM layer output is fed into a densely connected layer consisting of 64 units using 'relu' activation function and He initialization. The densely connected layer feeds the classification layer which uses 'softmax' activation to predict the probability of each of the three stance labels in our data set: FAVOR/NONE/AGAINST. Dropout is performed at two points: 1) prior to the LSTM layer and 2) prior to the classification layer. Refer to *Figure 2* for a diagram of our model infrastructure.

As previously mentioned, we expect a model that has the capacity to capture the unique rhetoric seen in

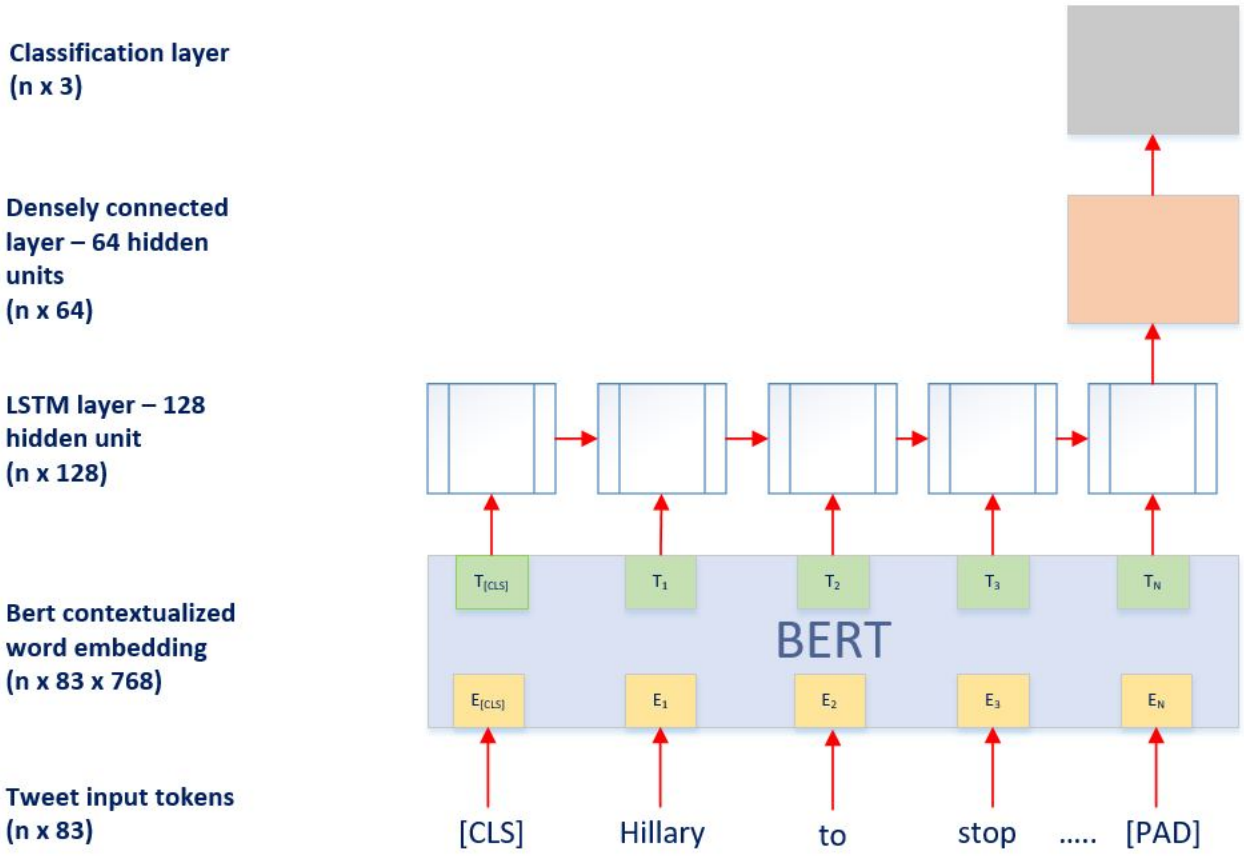

Figure 2: Model Diagram

tweets to perform well in stance classification. We felt that the combined use of BERT with a LSTM layer would provide the infrastructure suitable to capture the nuances of hashtags, sarcasm, and irony frequently used on Twitter.

4.2 Hyperparameter Tuning

Since independent models were trained for each of the five topics, we tuned the model's hyperparameters on a single topic, Hillary Clinton, and then applied the results to the remaining topics. Hillary Clinton was selected because it contained the most consistent balance of classes within both the training and test sets. In addition, its majority class was the 'AGAINST' stance label, which was consistent with all other topics, excluding Climate Change. Our overall approach for hyperparameter tuning was trial and error. Table XX below details the hyperparameters we focused on as well as the final value implemented in our models.

Table 1: Hyperparameters tuned and final values used

Hyperparameter	Final Value
Learning Rate	0.001
Dropout Rate	50%
Batch Size	32
# of Bert Fine Tune Layers	6 layers
Tweet Pre-processing	None

Note we ran our model both with tweets pre-processed (all words lower-cased and punctuation and digits removed) as well as without pre-processing. The model performed consistently better without pre-processing applied, and thus, we elected to train all topic models on the raw tweets.

4.3 Training

As mentioned above, we leveraged a batch size of 32 tweets for training. Adam optimization with a learning rate of 0.001 was used with categorical cross entropy loss. We used 20 epochs for training on all topics, except for Climate Change. Climate Change had a considerably smaller sample size, and thus, required fewer epochs (7 epochs) before overfitting the model to the training data. Since no development data was provided, we carved out 15% of the training data to use as dev data to determine whether we were overfitting during training.

5 Results

Our model's results are summarized in the 2 tables below. Table X1 shows the final ranking of our model in the context of 1) the competing teams in the SemEval-2016 conference and 2) all contributions to the task, including the SVM baseline and post-conference work. Table X2 shows our F1 scores for each topic in comparison to all teams that achieved a high score in at least 1 topic as well as the SVM baseline, [Sun], and [Du].

Table 2: Final F1 Scores with Ranks

Topic	F1 Score	SemEval-2016 Rank	Overall Rank
Overall Macro F1	60.88	1	2
Overall Micro F1	69.72	1	3
Atheism	69.66	1	2

Topic	F1 Score	SemEval-2016 Rank	Overall Rank
Hillary Clinton	64.48	2	3
Abortion	64.06	1	3
Climate Change	48.47	3	5
Feminism	57.76	2	3

Table 3: F1 Scores for all state-of-the-art models (overall or in at least 1 target)

Team	Overall (F-micro)	Overall (F-macro)	Atheism	Climate	Feminism	Hillary	Abortion
Muhammed et. al	70.32	59.01	69.19	43.80	58.72	61.74	66.91
Sun	69.79	61.00	70.53	49.56	57.50	61.84	66.16
Dessouky & Spittle	69.72	60.88	69.66	48.47	57.76	64.48	64.06
Du	68.79	59.56	59.77	53.59	55.77	65.38	63.72
MITRE	67.82	56.03	61.47	41.63	62.09	57.67	57.28
pkudblab	67.33	58.57	63.34	52.69	51.33	64.41	61.09
TakeLab	66.83	58.00	67.25	41.25	53.01	67.12	61.38
Majority Class	65.22	40.09	42.11	42.12	39.10	36.83	40.30
DeepStance	63.54	52.86	52.90	40.40	52.34	55.35	63.32
IDI@NTNU	62.47	55.08	59.59	54.86	48.59	57.89	54.47

The results in the above tables demonstrate the strength of our model. When comparing against teams participating in the SemEval2016 competition, we achieved the highest overall micro F1and macro F1 scores. We also achieved the top F1 score in the Atheism and Abortion topics, while finishing second in Hillary Clinton and Feminism, and third in Climate Change. When including [Sun] and [Du]'s post-conference contributions as well as the SVM baseline, we do not have any top scores. However, we still achieved the second highest macro F1 score, trailing [Sun] by only 0.12. We also achieved the third highest micro F1 score, behind [Sun] and [Muhammed et. al]. Note that the micro F1 score tends to be higher for models that perform better on the majority class, while macro F1 score tends to be higher for models that perform well on all classes. This can be seen in the comparison between the majority classifier's micro and macro F1 scores. The majority classifier scores a 65.22 on micro F1 score, but only 40.09 on macro F1. Our model achieved universally higher results than the majority classifier.

6 Conclusion

TBD Did we identify any state-of-the-art methodologies? No
What did we learn about Stance Detection? Messy

7 Limitations

Small amount of labeled data available
- High level of disagreement among encoders = expected ML classifier error
- Removed key hashtags from training/test data (would be incredibly valuable in the real world)
- Reproducibility - re-running the same model produces slightly different results
- Given such a small amount of data, training on separate models can be fit to 100% accuracy and produce different weights and therefore different results on test set

	Precision	Recall	F1.Score
Against	81.73	67.55	73.97
None	50.34	64.78	56.65
Favor	60.22	71.71	65.47

	F1.Score
F−micro	69.72
F−macro	60.88
Atheism	69.66
Climate Change	48.47
Feminism	57.76
Hillary Clinton	64.48
Abortion	64.06

Figure 3: F-Score Tables

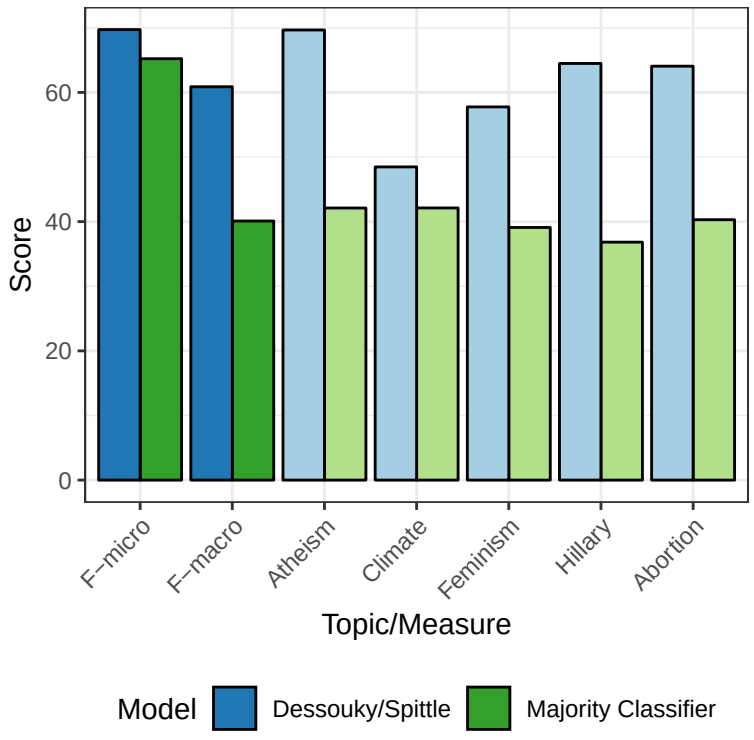

Figure 4: Results versus Majority Classifier

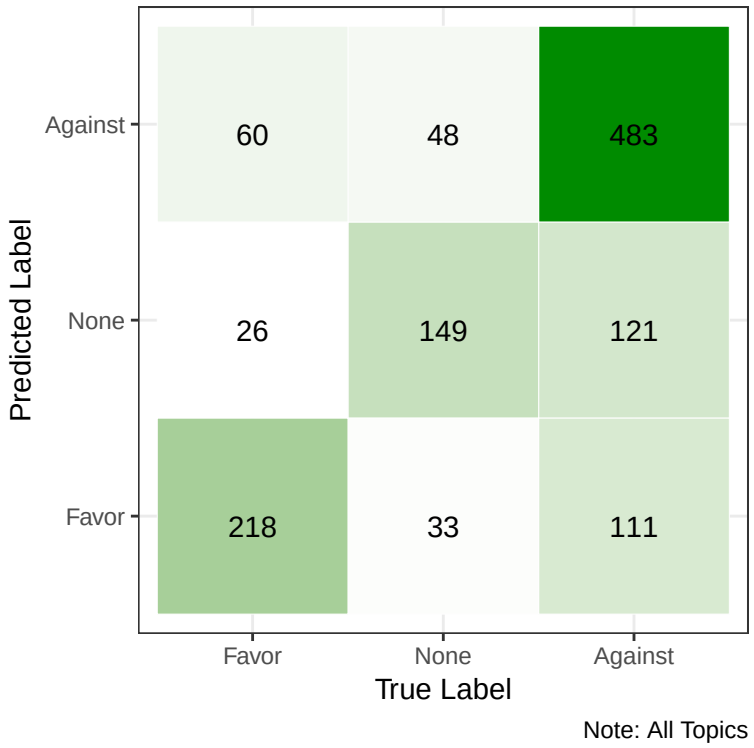

Figure 5: Confusion Matrix

7.1 Error Analysis

See *Figure 1* for confusion matrix ...

Errors for target **Atheism** were manually reviewed by the authors and categorized according to whether the purported "true" label or the predicted label from our model appeared to be correct (or if neither appeared correct).

It was surprising how many tweets appeared to have the incorrect true label associated with it. Some were blatantly and unambiguously wrong. Others has common challenges with the tweet text that may have caused some of these errors in annotation [*TO DO: add examples*]:

- *Referential*: Some tweets referenced very specific people or things that may be hard to align with the target or require a leap in assumption about connection of target (e.g. does supporting the NRA make you pro-choice, or are they just correlated)
- *Missing Context*:Some tweets were in reply to other tweets, whose content we cannot see, while others may suffer from missing the query hashtag
- *Sarcsm*: TBD

This apparent mis-labeling obviously causes some concern about the ability to train a model consistently that performs well on unseen data. However, there were additionally instances where our model made obvious mistakes. There were no systematic issues and it is difficult to determine what made these mis-classified tweets appear to have another stance. [**Not really sure what else to say here - reviewing errors not very enlightening**] It is notable that our model made consistent errors across true label categories, however the annotaters made errors exlusively in the against stance. The against stance constitutees 73% of test set

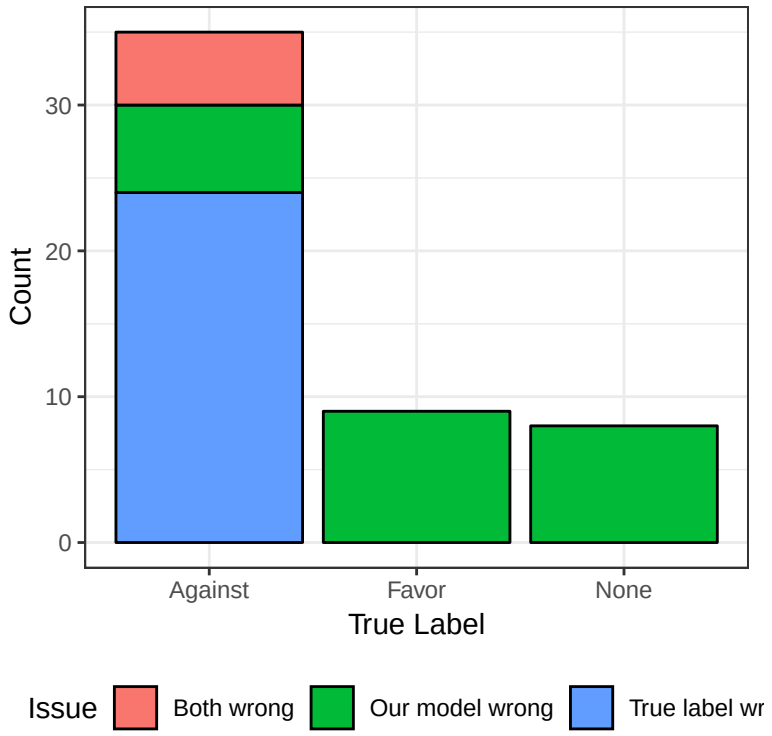

Figure 6: Atheism Error Analysis Summary of Issues

data observations for the atheism target, indicating that some amount of the errors may be attributable to fatigue, perhaps, or some factor that causes repeated annotation to lean toward the majority class.

7.2 Further Work

[*Ideas on gaps in our approach and/or logical next steps*]

References

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." http://arxiv.org/abs/1810.04805.

Mohammad, Saif, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016a. "A Dataset for Detecting Stance in Tweets." In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 3945–52. Portorož, Slovenia: European Language Resources Association (ELRA). https://www.aclweb.org/anthology/L16-1623.

———. 2016b. "SemEval-2016 Task 6: Detecting Stance in Tweets." In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 31–41. San Diego, California: Association for Computational Linguistics. https://doi.org/10.18653/v1/S16-1003.

Sobhani, Parinaz, Saif Mohammad, and Svetlana Kiritchenko. 2016. "Detecting Stance in Tweets and Analyzing Its Interaction with Sentiment." In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, 159–69. Berlin, Germany: Association for Computational Linguistics. https://doi.org/10.18653/v1/S16-2021.