Knowledge Enhanced Masked Language Model for Stance Detection

Kornraphop Kawintiranon and **Lisa Singh**
Department of Computer Science
Georgetown University
Washington, DC, USA
{kk1155,lisa.singh}@georgetown.edu

Abstract

Detecting stance on Twitter is especially challenging because of the short length of each tweet, the continuous coinage of new terminology and hashtags, and the deviation of sentence structure from standard prose. Fine-tuned language models using large-scale in-domain data have been shown to be the new state-of-the-art for many NLP tasks, including stance detection. In this paper, we propose a novel BERT-based fine-tuning method that enhances the masked language model for stance detection. Instead of random token masking, we propose using a weighted log-odds-ratio to identify words with high stance distinguishability and then model an attention mechanism that focuses on these words. We show that our proposed approach outperforms the state of the art for stance detection on Twitter data about the 2020 US Presidential election.

1 Introduction

Stance detection refers to the task of classifying a piece of text as either being in support, opposition, or neutral towards a given target. While this type of labeling is useful for a wide range of opinion research, it is particularly important for understanding the public's perception of given targets, for example, candidates during an election. For this reason, our focus in this paper is on detecting stance towards political entities, namely Joe Biden and Donald Trump during the 2020 US Presidential election.

Stance detection is related to, but distinct from the task of sentiment analysis, which aims to extract whether the general tone of a piece of text is positive, negative, or neutral. Sobhani and colleagues (Sobhani et al., 2016) show that measures of stance and sentiment are only 60% correlated. For example, the following sample tweet[1] has an obvious positive sentiment, but an opposing stance towards Donald Trump.

> *I'm so happy Biden beat Trump in the debate.*

Stance detection is an especially difficult problem on Twitter. A large part of this difficulty comes from the fact that Twitter content is short, highly dynamic, continually generating new hashtags and abbreviations, and deviates from standard prose sentence structure. Recently, learning models using pre-training (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Yang et al., 2019) have shown a strong ability to learn semantic representation and outperform many state-of-the-art approaches across different natural language processing (NLP) tasks. This is also true for stance detection. The strongest models for stance detection on Twitter use pre-trained BERT (Ghosh et al., 2019; Sen et al., 2018).

A recent study that proposed models for sentiment analysis (Tian et al., 2020) showed that focusing the learning model on some relevant words, i.e. sentiment words extracted using Pointwise Mutual Information (PMI) (Bouma, 2009), performed better than using the standard pre-trained BERT model. We are interested in understanding whether or not focusing attention on specific stance-relevant vocabulary during the learning process will improve stance detection. To accomplish this, we consider the following two questions. First, how do we identify the most important stance-relevant words within a data set? And second, how much attention needs to be paid to these words versus random domain words? Toward that end, we propose building different knowledge enhanced learning models that integrate an understanding of important context-specific stance words into the pre-training process.

[1] All of the sample tweets in this paper are invented by the authors. They are representative of real data, but do not correspond to any actual tweet in the data set in order to preserve the privacy of Twitter users.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4725–4735
June 6–11, 2021.

While we consider PMI as a way to identify important stance words, we find that using the log-odds ratio performs better.

We also consider different options for fine-tuning an attention-based language model. To fine-tune an attention-based language model to a specific task, the most common approach is to fine-tune using unlabeled data with random masking (Devlin et al., 2019; Liu et al., 2019). Because of the noise within social media posts, random tokens that are not task-relevant can impact sentence representation negatively. Therefore, instead of letting the model pay attention to random tokens, we introduce *Knowledge Enhanced Masked Language Modeling (KE-MLM)*, where significant tokens generated using the log-odds ratio are incorporated into the learning process and used to improve a downstream classification task. To the best of our knowledge, this is the first work that identifies significant tokens using log-odds-ratio for a specific task and integrates those tokens into an attention-based learning process for better classification performance.

In summary, we study stance detection on English tweets and our contributions are as follows. (i) We propose using the log-odds-ratio with Dirichlet prior for knowledge mining to identify the most distinguishable stance words. (ii) We propose a novel method to fine-tune a pre-trained masked language model for stance detection that incorporates background knowledge about the stance task. (iii) We show that our proposed knowledge mining approach and our learning model outperform the fine-tuned BERT in a low resource setting in which the data set contains 2500 labeled tweets about the 2020 US Presidential election. (iv) We release our labeled stance data to help the research community continue to make progress on stance detection methods.[2]

2 Related Work

In the NLP community, sentiment analysis is a more established task that has received more attention than stance detection. A sub-domain of sentiment analysis is target-directed or aspect-specific sentiment, which refers to the tone with which an author writes about a specific target/entity or an aspect of a target (Mitchell et al., 2013; Jiang et al., 2011). One common use case is breaking down sentiment toward different aspects of a product in reviews, e.g., the price of a laptop versus its CPU performance (Schmitt et al., 2018; Chen et al., 2017; Poddar et al., 2017; Tian et al., 2020). Different approaches have been proposed to tackle this problem. Chen and colleagues combine attention with recurrent neural networks (Chen et al., 2017). Schmitt and colleagues propose combining a convolutional neural network and fastText embeddings (Schmitt et al., 2018). A recent study proposes modifying the learning objective of the masked language model to pay attention to a specific set of sentiment words extracted by PMI (Tian et al., 2020). The model achieves new state-of-the-art results on most of the test data sets. Because stance is a different task,[3] we will adjust their target-directed sentiment approach for stance and compare to it in our empirical evaluation.

The most well-known data for political stance detection is published by the SemEval 2016 (Mohammad et al., 2016b; Aldayel and Magdy, 2019). The paper describing the data set provides a high-level review of approaches to stance detection using Twitter data. The best user-submitted system was a neural classifier from MITRE (Zarrella and Marsh, 2016) which utilized a pre-trained language model on a large amount of unlabeled data. An important contribution of this study was using pre-trained word embeddings from an auxiliary task where a language model was trained to predict a missing hashtag from a given tweet. The runner-up model was a convolutional neural network for text classification (Wei et al., 2016).

Following the MITRE model, there were a number of both traditional and neural models proposed for stance detection. A study focusing on traditional classifiers proposed using a support vector machine (SVM) with lexicon-based features, sentiment features and textual entailment feature (Sen et al., 2018). Another SVM-based model consisted of two-step SVMs (Dey et al., 2017). In the first step, the model predicts whether an input sequence is relevant to a given target. The next step detects the stance if the input sequence is relevant. Target-specific attention neural network (TAN) is a novel bidirectional LSTM-based attention model. In this study, Dey and colleagues trained it on unpublished unlabeled data to learn the domain context (Du et al., 2017). Recently,

[2] https://github.com/GU-DataLab/stance-detection-KE-MLM

[3] Stance detection aims to detect the opinion s to the specific target e, aspect-based sentiment focuses on extracting the aspect a towards the target e and corresponding opinion s (Wang et al., 2019).

a neural ensemble model consisting of bi-LSTM, nested LSTMs, and an attention model was proposed for stance detection on Twitter (Siddiqua et al., 2019). The model's embedding weights were initialized with the pre-trained embeddings from fastText (Bojanowski et al., 2017).

The emergence of transformer-based deep learning models has led to high levels of improvement for many NLP tasks, including stance detection (Ghosh et al., 2019; Küçük and Can, 2020; AlDayel and Magdy, 2020). BERT (Devlin et al., 2019) is the most used deep transformer encoder. More specifically, BERT uses Masked Language Modeling (MLM) to pre-train a transformer encoder by predicting masked tokens in order to learn the semantic representation of a corpus. Ghosh and colleagues (Ghosh et al., 2019) show that the original pre-trained BERT without any further fine-tuning outperforms other former state-of-the-art models on the SemEval set including the model that utilizes both text and user information (Del Tredici et al., 2019). Because we are interested in the 2020 US Presidential election and many temporal factors relevant to stance exist (e.g. political topics), we introduce a new Election 2020 data set. For our empirical analysis, we will use this data set, and compare our approach to other state-of-the-art methods that used the SemEval data set. Our data sets are described in Section 5.1.

Inspired by BERT, different variations of BERT have been proposed to solve different specific NLP tasks. SpanBERT (Joshi et al., 2019) masks tokens within a given span range. ERNIE (Sun et al., 2019) finds and masks entity tokens achieving new state-of-the-art results on many Chinese NLP tasks, including sentiment analysis. GlossBERT (Huang et al., 2019) uses gloss knowledge (sense definition) to improve performance on a word sense disambiguation task. SenseBERT (Levine et al., 2020) aims to predict both masked words and the WordNet super-sense to improve word-in-context tasks. Zhang and colleagues introduce entity token masking (Zhang et al., 2019) for relation classification where the goal is to classify relation labels of given entity pairs based on context. A number of studies have been working on adjusting transformers for sentiment analysis tasks. A recent study (Tian et al., 2020) proposes a sentiment knowledge enhanced pre-training method (SKEP). It shows that masking sentiment words extracted by PMI guides the language model to learn more sentiment knowledge resulting in better sentiment classification performance. SentiLARE (Ke et al., 2020) uses an alternative approach that injects word-level linguistic knowledge, including part-of-speech tags and sentiment polarity scores obtained by SentiWordNet (Guerini et al., 2013), into the pre-training process. Following these works, SENTIX (Zhou et al., 2020) was proposed to incorporate domain-invariant sentiment knowledge for cross-domain sentiment data sets. Our work differs because our task is stance detection and we employ a novel knowledge mining step that uses log-odds-ratio to determine significant tokens that need to be masked.

3 KE-MLM: Knowledge Enhanced Masked Language Modeling

We propose Knowledge Enhanced Masked Language Modeling (KE-MLM), which integrates knowledge that enhances the classification task in the fine-tuning process. We identify task-relevant tokens using text mining (Section 3.1). We then use these discovered tokens within a masked language model (Section 3.2).

3.1 Knowledge Mining for Classification

While TF-IDF is the preferred method for identifying important words in a corpus, we are interested in identifying important words for distinguishing stance, not just words that are important within the corpus. Therefore, we propose using the weighted log-odds-ratio technique with informed Dirichlet priors proposed by Monroe and colleagues (Monroe et al., 2008) to compute significant words for each stance class. Intuitively, this measure attempts to account for the amount of variance in a word's frequency and uses word frequencies from a background corpus as priors to reduce the noise generated by rare words. This technique has been shown to outperform other methods that were designed to find significant words within a corpus such as PMI and TF-IDF (Monroe et al., 2008; Jurafsky et al., 2014; Budak, 2019).

More formally, we compute the usage difference for word w among two corpora using the log-odds-ratio with informative Dirichlet priors as shown in the Equation 1, where n^i is the size of corpus i and n^j is the size of corpus j. y_w^i and y_w^j indicate the word count of w in corpus i and j, respectively. α_0 is the size of the background corpus and α_w is the word count of w in the background corpus.

$$\delta_w^{(i-j)} = \log \frac{y_w^i + \alpha_w}{n^i + \alpha_0 - y_w^i - \alpha_w} - \log \frac{y_w^j + \alpha_w}{n^j + \alpha_0 - y_w^j - \alpha_w} \quad (1)$$

To measure the significance of each word, we first compute the variance (σ^2) of the log-odds-ratio using Equation 2, and then compute the Z-score using Equation 3. A higher score indicates more significance of word w within corpus i compared to corpus j. A lower score means more significance of word w within corpus j compared to corpus i.

$$\sigma^2(\delta_w^{(i-j)}) \approx \frac{1}{y_w^i + \alpha_w} + \frac{1}{y_w^j + \alpha_w} \quad (2)$$

$$Z = \frac{\delta_w^{(i-j)}}{\sqrt{\sigma^2(\delta_w^{(i-j)})}} \quad (3)$$

Since stance has three different classes (support, opposition and neutral), we need to adjust the log-odds-ratio technique in order to obtain a set of significant stance words. Using a training set, we find stance tokens which are significant tokens for support/non-support or opposition/non-opposition as follows:

- **Supportive & Non-supportive tokens** are the highest and lowest Z-score tokens, respectively when i only contains the support class and j contains only the opposition and neutral classes.
- **Opposing & Non-opposing tokens** are the highest and lowest Z-score tokens, respectively when i only contains the opposition class and j only contains the support and neutral classes.

We select the highest and lowest k tokens based on Z-score from each token list above. This results in four k-token lists. The combined tokens of these lists after removing duplicates are defined to be the *stance tokens*. We hypothesize that these stance tokens will play a key role during stance detection.

3.2 Significant Token Masking

There are two main approaches to train a transformer encoder, Causal Language Modeling (CLM) and Masked Language Modeling (MLM). CLM has a standard language modeling objective, predicting the next token given all previous tokens in the input sequence. This means that it needs to learn tokens in order and can only see the previous tokens. On the other hand, MLM uses a masking technique that is more flexible, allowing researchers to explicitly assign which tokens to mask. The other tokens are used for masked token recovery. Intuitively, a language model that learns to recover a specific set of tokens well will tend to produce a better semantic representation for sequences containing those tokens (Tian et al., 2020; Ke et al., 2020; Zhou et al., 2020). Generally, randomly masking tokens is preferred when the task requires the language model to learn to recover all tokens equally. This tends to result in a semantic representation that is equally good for any input sequences.

In many BERT-based models, when training the transformer encoder with masked language modeling, the input sequence is modified by randomly substituting tokens of the sequence. Specifically, BERT uniformly chooses 15% of input tokens of which 80% are replaced with a special masked token $[MASK]$, 10% are replaced with a random token, and 10% are not replaced and remain unchanged. The goal of significant token masking is to produce a corrupted version of the input sequence by masking the significant tokens rather than random tokens. We keep the same ratio of masked tokens by masking up to 15% of the significant tokens. If fewer than 15% of the tokens are significant, we randomly mask other tokens to fill up to 15%.[4]

Formally, significant word masking creates a corrupted version $\acute{X}$ for an input sequence X that is influenced by the extracted knowledge G. Tokens of sequences X and $\acute{X}$ are denoted by x_i and $\acute{x_i}$, respectively. In the fine-tuning process, the transformer encoder is trained using a masked word prediction objective that is supervised by recovering masked significant words using the final state of the encoder $\acute{x_1}, ..., \acute{x_n}$, where n is the length of the sequence.

After constructing this corrupted version of the sequence, MLM aims to predict the masked tokens to recover the original tokens. In this paper, we inject knowledge for our specific classification task during MLM, causing the model to pay more attention to stance tokens instead of random tokens.

[4]With a set of 20-40 significant words, their word counts are roughly 1% of the total number of tokens of the unlabeled data that we trained the language model on.

Table 1: Example sets of strong supportive and opposing tokens for both candidates on Twitter.

	Biden	Trump
Support	administration, ballot, bluewave, early, kamala, rule, safe, show, trust, voteblue	americafirst, follow, help, ifbap, kag, maga, patriots, retweet, thanks, votered
Oppose	bernie, black, blah, cities, kag, maga, money, patriots, woman, wwg1wga	bluewave, consequences, demconvention, division, liar, make, republicans, resist, stand

These results are based on real data. Tokens are sorted alphabetically.

Formally, we get an embedding vector $\tilde{x_i}$ from the transformer encoder by feeding the corrupted version $\acute{X}$ of input sequence X. Next, the embedding vector is fed into a single layer of neural network with a softmax activation layer in order to produce a normalized probability vector $\hat{y_i}$ over the entire vocabulary as shown in Equation 4, where W is a weight vector and b is a bias vector. Therefore, the prediction objective L is to maximize the probability of original token x_i computed in Equation 5, where $m_i = 1$ if the token at the i-th position is masked, otherwise $m_i = 0$ and y_i is a one-hot representation of the original token.

$$\hat{y_i} = softmax(\tilde{x_i}W + b) \tag{4}$$

$$L = -\sum_{i=1}^{i=n} m_i \times y_i \log \hat{y_i} \tag{5}$$

Finally, we fine-tune a pre-trained BERT with unlabeled in-domain (election 2020) data. The representation learned by the language model is expected to be customized for the stance detection task.

4 Experimental Design

In this section we describe our experimental design, beginning with the knowledge mining decisions, followed by the decisions and parameters used for the language models.

4.1 Stance Knowledge Mining

We begin by determining the number of significant stance words to identify. Based on a sensitivity analysis, we set $k = 10$ to extract the top-10 significant words for each stance category as described in Section 3.1 (support, non-support, oppose, non-oppose). Examples of significant tokens from the strong supportive/opposing stance are shown in Table 1. Our stance detection models are independently trained for each candidate, so overlapping tokens are allowed (e.g. the word *patriots* tends to support Trump but oppose Biden). Once we have a set of tokens for the four categories, we union these four token sets. After removing duplicates, there are roughly 30 stance tokens for each candidate.

4.2 Language Models

Because the state-of-the-art models for stance detection are neural models with pre-trained language models on a large amount of in-domain data, (Zarrella and Marsh, 2016; Küçük and Can, 2020), we use both original pre-trained BERT and BERT fine-tuned on the unlabeled election data as our benchmarks. We fine-tuned BERT for two epochs since it gives the best perplexity score[5]. For KE-MLM, we first initialize the weights of the model using the same values as the original BERT, then we fine-tune the model with unlabeled election data using the identified stance tokens masked. We exhaustively fine-tuned KE-MLM to produce the language model that focuses attention on the stance tokens from the training set.

Because BERT's tokenizer uses WordPiece (Wu et al., 2016), a subword segmentation algorithm, it cannot learn new tokens after the pre-training is finished without explicitly specifying it. However, adding new tokens with random embedding weights would cause the pre-trained model to work differently since it was not pre-trained with those new tokens. We realize that some significant tokens for the stance of Election 2020 are new to the BERT and were not in the original BERT pre-training process. Therefore, we consider adding all the stance words to the BERT tokenizer. We hypothesize that adding such a small number of tokens will barely affect the pre-trained model. To test the effect of adding stance tokens into the normal fine-tuning process, we train language models in which stance tokens are added, but we fine-tune them with the normal random masking method. We refer to this model as *a-BERT*, where stance tokens

[5]Perplexity is a performance measurement of the masked language model, a lower score is better.

are added to the BERT tokenizer, but only the standard fine-tuning method is performed. To compare our performance to the sentiment knowledge enhanced pre-training method or SKEP (Tian et al., 2020), we use the pre-training method proposed in their paper and then fine-tune the model using our election 2020 data (SKEP).

We hypothesize that applying KE-MLM may guide the language model to focus too much attention on the stance knowledge and learn less semantic information about the election itself. Therefore, we consider a hybrid fine-tuning strategy. We begin by fine-tuning BERT for one epoch. Then we fine-tune using KE-MLM in the next epoch. This hybrid strategy forces the model to continually learn stance knowledge along with semantic information about the election. We expect that this dual learning will construct a language model biased toward necessary semantic information about the election, as well as the necessary embedded stance knowledge. We refer to this approach as our KE-MLM (with continuous fine-tuning), while KE-MLM– refers to a model that is overly fine-tuned with only stance token masking.

To summarize, the language models we will evaluate are as follows: the original pre-trained BERT (o-BERT), a normally fine-tuned BERT that uses our election data (f-BERT), a normally fine-tuned BERT that uses stance tokens as part of its tokenizer (a-BERT), a fine-tuned BERT using the SKEP method (Tian et al., 2020) (SKEP), our overly fine-tuned model (KE-MLM–), and our hybrid fine-tuned model (KE-MLM). For all the language models, we truncate the size of an input sequence to 512 tokens. The learning rate is constant at $1e-4$ and the batch size is 16.

4.3 Classification Models

In masked language modeling, we fine-tune the model using a neural layer on top with the learning objective to predict masked tokens. In this step, we substitute that layer with a new neural layer as a stance classifier layer. Its weights are arbitrarily initialized. The prediction equation is similar to Equation 4 but now the input is not corrupted, and the output is a vector of the normalized probability of the three stance classes. We use a cross-entropy loss function and the objective is to minimize it. We use the Adam optimizer (Kingma and Ba, 2015) with five different learning rates, including $2e-5$, $1e-5$, $5e-6$, $2e-6$ and $1e-6$. The batch size is constantly set to 32 during the classification learning process.

We train and test our models on each candidate independently with five different learning rates. The best model is determined by the best macro average F1 score over three classes among five learning rates. Because the weights of the classifier layer are randomly initialized, we run each model five times. The average F1 score is reported in Table 2 as the classification performance.

5 Empirical Evaluation

After describing our data set (Section 5.1), we present our experimental evaluation, both quantitative (Section 5.2), and qualitative (Section 5.3).

5.1 Data Sets

For this study, our research team collected English tweets related to the 2020 US Presidential election. Through the Twitter Streaming API, we collected data using election-related hashtags and keywords. Between January 2020 and September 2020, we collected over 5 million tweets, not including quotes and retweets. These unlabeled tweets were used to fine-tune all of our language models.

Our specific stance task is to determine the stance for the two presidential candidates, Joe Biden and Donald Trump. For each candidate, we had three stance classes: support, opposition, and neutral.[6] We consider two stance-labeled data sets, one for each candidate, Biden and Trump. Our data were labeled using Amazon Mechanical Turk (MTurk) workers (Crowston, 2012). These workers were not trained. Instead, we provided a set of examples for each stance category that they could refer to as they conducted the labeling task. Examples of statements presented to MTurk workers are presented in Table 3. We asked annotators to carefully review each tweet t_c^i from the tweet set $T_C = \{t_c^1, t_c^2, ...\}$ and determine whether the tweet t_c^i is *(i)* clearly in support of C, *(ii)* clearly in opposition to C or *(iii)* not clearly in support or opposition to C, where $t_c^i \in T_C$ and $C \in \{\text{Donald Trump}, \text{Joe Biden}\}$. To increase the labeling yield, we verify that two tweet sets $T_{C=\text{Donald Trump}}$ and $T_{C=\text{Joe Biden}}$ are mutually exclusive. Each tweet was labeled by three annotators and the majority vote is considered to be the true label. If all three annotators vote for three differ-

[6] Our definition of stance labels is consistent with the definition from (Mohammad et al., 2016a)

Table 2: The average F1 scores over five runs. The confidence intervals for the macro F1 scores are computed based on a significance level of 0.05, meaning a 95% confidence level. The highest scores are shown in boldface.

	Biden				Trump			
Model	F1-Support	F1-Oppose	F1-Neutral	F1-macro	F1-Support	F1-Oppose	F1-Neutral	F1-macro
o-BERT	0.7324	0.6875	0.7151	0.7117 (±0.0063)	0.7574	0.8101	0.6955	0.7543 (±0.0069)
f-BERT	0.7743	0.7226	0.7347	0.7439 (±0.0049)	0.7921	0.8147	0.6961	0.7677 (±0.0084)
a-BERT	0.7905	0.7234	0.7432	0.7523 (±0.0049)	0.8090	0.8154	0.6926	0.7724 (±0.0078)
SKEP	0.7923	0.7153	0.7349	0.7475 (±0.0047)	0.7852	0.8169	0.7151	0.7724 (±0.0067)
KE-MLM–	0.7618	0.7303	0.7380	0.7434 (±0.0040)	0.7854	0.7968	0.7083	0.7635 (±0.0081)
KE-MLM	**0.7927**	**0.7329**	**0.7475**	**0.7577 (±0.0032)**	**0.8094**	**0.8184**	**0.7354**	**0.7877 (±0.0075)**

Table 3: Sample of stance examples presented to MTurk labelers.

Candidate	Statement	Stance
Biden	Biden will be a great president. I am voting for him in November.	Support
	Biden has handled the pandemic poorly.	Oppose
	Biden spoke in Pennsylvania.	Neutral
Trump	Trump has been a great president. I am voting for him in November.	Support
	Trump has handled the pandemic poorly.	Oppose
	Trump held a rally yesterday.	Neutral

ent classes, we assume the tweet's label is neutral because the stance is ambiguous.

Our data set contains 1250 stance-labeled tweets for each candidate. The stance label distributions are shown in Table 4. The distributions of both candidates are skewed towards the opposition label. Overall, the stance class proportions vary from 27% to 39%. The inter-annotator agreement scores from different metrics are shown in Table 5. The task-based and worker-based metrics are recommended by the MTurk official site (Amazon, 2011), given their annotating mechanism. All scores are range from 86% up to 89%, indicating the high inter-rater reliability for these data sets.

Table 4: Stance distribution for Biden and Trump.

	%SUPPORT	%OPPOSE	%NEUTRAL
Biden	31.3	39.0	29.8
Trump	27.3	39.9	32.8

Table 5: Mechanical Turk inter-annotator agreement for Biden and Trump.

Metric	Biden	Trump
Task-based	0.8693	0.8920
Worker-based	0.8915	0.8969

5.2 Experimental Results

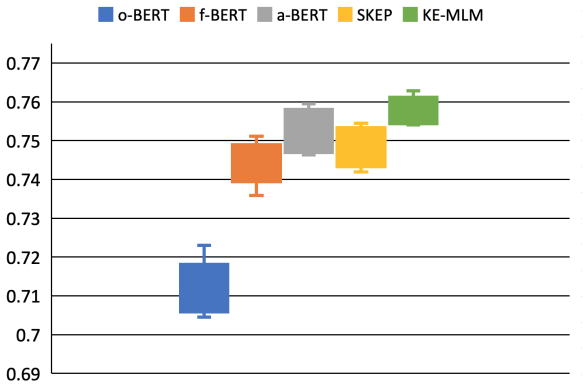
We conducted experiments on train-test sets using a 70:30 split for both the Biden and Trump data sets.[7] We evaluate the classification performance using the macro-average F1 score along with the F1 score of each class. The results presented in Table 2 show the average F1 scores over five runs with different random seeds. The highest score for each evaluation metric is highlighted in bold.

For Biden-stance, every fine-tuning method (f-BERT, a-BERT, SKEP, KE-MLM– and KE-MLM) improves the average F1 score from the original pre-trained model by 3.2%, 4.1%, 3.6%, 3.2% and 4.6%, respectively. For Trump-stance, the average F1 scores are also improved by 1.3%, 1.8%, 1.8%, 0.9% and 3.3%. The improvement is twice as much for Biden than for Trump. This is an indication that the additional background knowledge is more important for detecting stance for Biden than for Trump. In general, our knowledge enhanced model performs better than all the other models and outperforms the original BERT by three to five percent. a-BERT performs similarly to SKEP for Trump, but its performance is better for Biden. The model's overall performances are second-best with only a difference of 0.5% and 1.5% in the average F1-macro score when compared to KE-MLM for Biden and Trump, respectively. These results further highlight the importance of incorporating stance tokens into the tokenizer. While adding stance to the tokenization is important, the additional improvement of KE-MLM comes from focusing attention on both the stance tokens and the general election data. The result also supports our hypothesis that training KE-MLM– alone for two epochs would result in better accuracy than original BERT (o-BERT), but a lower accuracy than normally fine-tuned BERT (f-BERT) because it learns stance knowledge but lacks in-domain election knowledge.

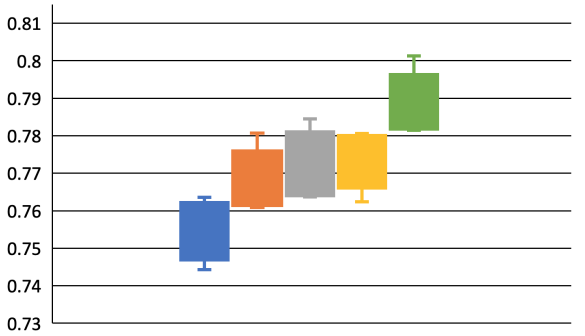
To better understand the robustness of our models, we analyze the variance in the F1 scores across

[7] Because we do not have sufficient unlabeled election data from 2016, we cannot fairly test our model with the SemEval 2016 stance data.

the different runs. Figure 1 shows the box plots of the macro average F1 scores for each model. The scores of both candidates follow a similar pattern. For Biden, the highest F1 score and the lowest variance is KE-MLM. For Trump, the highest F1 score is KE-MLM, but the variance is comparable to the other models. The model with the lowest variance is SKEP. These figures further emphasize KE-MLM's ability to detect stance better than normally fine-tuning methods. Interestingly, a-BERT performs second-best (see gray boxes in Figure 1), further highlighting the importance of not ignoring stance tokens. Forcefully adding unseen stance tokens to the BERT tokenizer with random initial weights benefits overall classification performance.



(a) Biden



(b) Trump

Figure 1: The distribution of macro average F1 scores from five independent runs.

Additionally, we conducted a sensitivity analysis on different sizes of unlabeled data for pre-training to verify that the large unlabeled data is actually beneficial. We fine-tune f-BERT using different sizes of data (100K, 500K, 1M, 2M) and compare the results to those of BERT with zero-fine-tuning (o-BERT) and fine-tuning using the entire 5M tweets (f-BERT). We train each pre-trained language model on training and test on testing data set five times. The average F1 scores are shown in Fig 2. For Biden, the average F1 score is 3% lower when there is no fine-tuning compared to using all 5M tweets. For Trump, the score only improves a little over 1%. Interestingly, as the size of the unlabeled data increases, the F1 score also increases even though the increase is not always large. Therefore, pre-training using a smaller size unlabeled data set does still produce benefits, but when possible, using a large sample does lead to improvement.

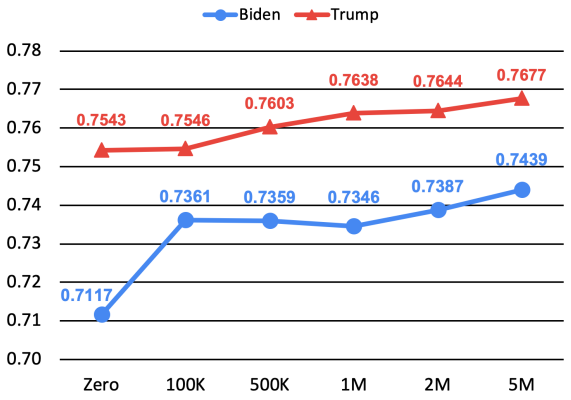


Figure 2: The model performance by f-BERT pre-trained on different sizes of unlabeled data. We train each model five times and report the average F1 scores.

5.3 Qualitative Analysis of the Effect of Stance Knowledge

While we see from Table 2 that KE-MLM outperforms all baselines on average, we are interested in understanding when there is labeling disagreement between other methods and KE-MLM, what features are driving the disagreement. Therefore, we manually investigate samples in which f-BERT and a-BERT produced incorrect predictions, while KE-MLM produced correct ones. On average over multiple runs, 28.8% and 38.5% of misclassified tweets by f-BERT are correctly predicted by KE-MLM for Biden and Trump, respectively. For a-BERT, they are 22.5% and 25.7% on average. As a case example, Table 6 illustrates the attention distribution of the sequence representation learned by each language model for a few mislabeled tweets. Significant words are colored. The color darkness is determined by the attention weights of the representation learned for the classification token.[8] The

[8]The representation of classification tokens produced by a transformer encoder is usually referred to as $[CLS]$. Please see (Devlin et al., 2019) for details about the attention weight calculation.

Table 6: Visualization of selected samples with attention weight distribution by color darkness.

Candidate	Model	Sampled Sentence	Prediction
Biden	f-BERT	The democrats and @joebiden believe in the power of the government. The #gop and @realdonaldtrump believe in the power of the american people. #maga	Neutral
	a-BERT	The democrats and @joebiden believe in the power of the government. The #gop and @realdonaldtrump believe in the power of the american people . #maga	Neutral
	KE-MLM	The democrats and @joebiden believe in the power of the government. The #gop and @realdonaldtrump believe in the power of the american people. #maga	Opposition
Trump	f-BERT	Covid -19 was Trump's biggest test. He failed miserably . #demconvention	Neutral
	a-BERT	Covid -19 was Trump's biggest test. He failed miserably. #demconvention	Neutral
	KE-MLM	Covid -19 was Trump's biggest test. He failed miserably . #demconvention	Opposition

darker the color the more important the word. From the selected samples, we know from the knowledge mining step that the word "maga" and "demconvention" are two of the most distinguishing stance words (see Table 1), but both f-BERT and a-BERT fail to identify these strong stance words and therefore, produced incorrect predictions. In contrast, KE-MLM produces the correct predictions by paying reasonable attention to the stance information, further supporting the notion that KE-MLM is using meaningful, interpretable tokens.

6 Conclusions and Future Directions

Intuitively, a language model fine-tuned using in-domain unlabeled data should result in better classification performance than using the vanilla pre-trained BERT. Since our goal is to maximize the accuracy of a specific classification task, we train an attention-based language model to pay attention to words that help distinguish between the classes. We have shown that for stance detection, using the log-odds-ratio to identify significant tokens that separate the classes is important knowledge for this classification task. Once these important tokens are identified, forcing the language model to pay attention to these tokens further improves the performance when compared to using standard data for fine-tuning. To the best of our knowledge, our approach is better than the other state-of-the-art approaches for stance detection. Additionally, we are releasing our data set to the community to help other researchers continue to make progress on the stance detection task. We believe this is the first stance-labeled Twitter data for the 2020 US Presidential election.

There are several future directions of this work. First, to relax the trade-off between learning election semantics in general and learning stance knowledge, instead of fine-tuning one epoch with the normal fine-tuning method and another epoch with KE-MLM, we could reduce the masking probability of stance distinguishing words from 100% to something lower based on the distinguishability of the token. Theoretically, this would give a higher weight to words that are more polarizing. This also relaxes the potential overfitting that may occur when learning only stance knowledge and lets the model randomly learn more tokens. Another future direction is to test our language modeling method on other classification tasks (e.g. sentiment analysis, spam detection). Also, this paper uses BERT as the base language model. There are many variations of BERT that can be further investigated (e.g. RoBERTa). Finally, we view stance as an important task for understanding public opinion. As our models get stronger, using them to gain insight into public opinion on issues of the day is another important future direction.

Acknowledgements

This research was funded by National Science Foundation awards #1934925 and #1934494, and the Massive Data Institute (MDI) at Georgetown University. We would like to thank our funders, the MDI staff, and the members of the Georgetown DataLab for their support. We would also like to thank the anonymous reviewers for the detailed and thoughtful reviews.

References

Abeer Aldayel and Walid Magdy. 2019. Your stance is exposed! analysing possible factors for stance detection on social media. *The ACM on Human-Computer Interaction*, 3(CSCW):1–20.

Abeer AlDayel and Walid Magdy. 2020. Stance detection on social media: State of the art and trends. *arXiv preprint arXiv:2006.03644*.

Amazon. 2011. Hit review policies. https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_HITReviewPolicies.html.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics (TACL)*, 5:135–146.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the biennial Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*.

Ceren Budak. 2019. What happened? the spread of fake news publisher content during the 2016 us presidential election. In *Proceedings of the World Wide Web Conference (WWW)*.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kevin Crowston. 2012. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches*.

Marco Del Tredici, Diego Marcheggiani, Sabine Schulte im Walde, and Raquel Fernández. 2019. You shall know a user by the company it keeps: Dynamic representations for social media users in NLP. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Kuntal Dey, Ritvik Shrivastava, and Saroj Kaushik. 2017. Twitter stance detection—a subjectivity and sentiment polarity inspired two-phase approach. In *IEEE international conference on data mining workshops (ICDMW)*.

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*.

Shalmoli Ghosh, Prajwal Singhania, Siddharth Singh, Koustav Rudra, and Saptarshi Ghosh. 2019. Stance detection in web and social media: a comparative study. *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 75–87.

Marco Guerini, Lorenzo Gatti, and Marco Turchi. 2013. Sentiment analysis: How to derive prior polarities from SentiWordNet. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL)*, 8:64–77.

Dan Jurafsky, Victor Chahuneau, Bryan R Routledge, and Noah A Smith. 2014. Narrative framing of consumer sentiment in online restaurant reviews. *First Monday*.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 2020. SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Computing Surveys (CSUR)*, 53(1).

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving some sense into BERT. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016a. A

dataset for detecting stance in tweets. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016b. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Lahari Poddar, Wynne Hsu, and Mong Li Lee. 2017. Author-aware aspect topic sentiment model to retrieve supporting opinions from reviews. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI*.

Martin Schmitt, Simon Steinheber, Konrad Schreiber, and Benjamin Roth. 2018. Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Anirban Sen, Manjira Sinha, Sandya Mannarswamy, and Shourya Roy. 2018. Stance classification of multi-perspective consumer health information. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*.

Umme Aymun Siddiqua, Abu Nowshed Chy, and Masaki Aono. 2019. Tweet stance detection using an attention based neural ensemble model. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Parinaz Sobhani, 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 (*SEM)*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Rui Wang, Deyu Zhou, Mingmin Jiang, Jiasheng Si, and Yang Yang. 2019. A survey on opinion mining: From stance to product aspect. *IEEE Access*, 7:41101–41124.

Wan Wei, Xiao Zhang, Xuqin Liu, Wei Chen, and Tengjiao Wang. 2016. pkudblab at SemEval-2016 task 6 : A specific convolutional neural network system for effective stance detection. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Guido Zarrella and Amy Marsh. 2016. Mitre at semeval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jie Zhou, Junfeng Tian, Rui Wang, Yuanbin Wu, Wenming Xiao, and Liang He. 2020. SentiX: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.