Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 7360–7367
Marseille, 20-25 June 2022

PoliBERTweet: A Pre-trained Language Model for Analyzing Political Content on Twitter

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

Abstract

Transformer-based models have become the state-of-the-art for numerous natural language processing (NLP) tasks, especially for noisy data sets, including social media posts. For example, BERTweet, pre-trained RoBERTa on a large amount of Twitter data, has achieved state-of-the-art results on several Twitter NLP tasks. We argue that it is not only important to have general pre-trained models for a social media platform, but also domain-specific ones that better capture domain-specific language context. Domain-specific resources are not only important for NLP tasks associated with a specific domain, but they are also useful for understanding language differences across domains. One domain that receives a large amount of attention is politics, more specifically political elections. Towards that end, we release PoliBERTweet, a pre-trained language model trained from BERTweet on over 83M US 2020 election-related English tweets. While the construction of the resource is fairly straightforward, we believe that it can be used for many important downstream tasks involving language, including political misinformation analysis and election public opinion analysis. To show the value of this resource, we evaluate PoliBERTweet on different NLP tasks. The results show that our model outperforms general-purpose language models in domain-specific contexts, highlighting the value of domain-specific models for more detailed linguistic analysis. We also extend other existing language models with a sample of these data and show their value for presidential candidate stance detection, a context-specific task. We release PoliBERTweet and these other models to the community to advance interdisciplinary research related to Election 2020.

Keywords: Transformer, BERT, RoBERTa, BERTweet, Language Model, Politics, Election 2020, Twitter

1. Introduction

Transformer-based language models (LM) have become the state-of-the-art algorithms for many NLP tasks. While the most well-known transformers are BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), the number of variants continue to grow (Jiang et al., 2020b; Joshi et al., 2019; Sanh et al., 2019; Lan et al., 2020). In principle, each model utilizes the unlabeled text corpora using different objective learning functions, for example, masked language modeling (MLM), next sentence prediction (NSP), or sentence order prediction (SOP) (Vaswani et al., 2017). All these mechanisms and functions have the same goal – to learn semantic representation with respect to the training corpus and generate contextualized embeddings given a textual input.

Recently, BERTweet, a pre-trained RoBERTa model trained on Twitter data collected from 2012 to 2019 was released (Nguyen et al., 2020). While this language model is valuable as a general-purpose Twitter-based model, we believe that data scientists and computational social scientists will find it valuable to have a model that is focused on recent political and election semantics related to the 2020 US presidential election. We also note that the previous work does not include the 2020 US election.

Pre-trained transformer-based models have been successful for many politics-related tasks on Twitter, including sentiment analysis (Tian et al., 2020; Ke et al., 2020; Zhou et al., 2020), stance detection (Kawintiranon and Singh, 2021; ALDayel and Magdy, 2021; Budak et al., 2020) and fake news detection (Shu et al., 2020; Shaar et al., 2020; Hossain et al., 2020; Jiang et al., 2020a).

Given the current political climate in the US, we anticipate even more interest in the 2020 US Presidential election than in the 2016 one. Examples of relevant election 2016 research include stance detection (ALDayel and Magdy, 2021; Hardalov et al., 2021), fake news detection (Oshikawa et al., 2020; Bozarth and Budak, 2020) and Twitter content-based political science research (Bode et al., 2020; Pasek et al., 2020b; Pasek et al., 2020a; Budak et al., 2016; Wang et al., 2012; Giachanou and Crestani, 2016). While a general purpose Twitter language model would support this research, capturing more detailed domain specific semantic knowledge and language subtlety may be particularly important for tasks involving detailed language analysis like misinformation analysis or election public opinion analysis.

To support and advance interdisciplinary research in this domain, the contributions of the paper are as follows: (1) we release a new English election 2020 language resource, PoliBERTweet[1], focused on capturing language usage on Twitter specific to the 2020 US Pres-

[1] https://github.com/GU-DataLab/PoliBERTweet

idential Election, (2) we conduct a detailed comparison between PoliBERTweet and existing language models on both political and general tasks, demonstrating the value of using a domain-specific language model, and (3) we pre-train three other transformer-based models on English US politics-related Twitter data and analyze them in the context of stance detection.

The remainder of this paper is organized as follows. We review the state-of-the-art language models in Section 2. The details of our pre-training data and models are described in Section 3. To provide a reasonable benchmark for the NLP community, we evaluate all our models on perplexity, masked token prediction and stance detection. The evaluation is explained in Section 4. Finally, we present concluding thoughts in Section 5.

2. Language Models

A number of studies have shown that pre-trained language models (PTMs) are beneficial for downstream NLP tasks. Their goal is to learn universal language representations from large corpora. In doing so, they capture important language semantics that can be readily used for different learning tasks, as opposed to relearning the more general relationships during the training process of each downstream task.

The first generation of PTMs focused on learning context-free word embeddings such as Word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). The next generation moved to learning contextual word embeddings such as ELMo (Peters et al., 2018) and OpenAI-GPT (Radford et al., 2018). Given the recent successes using transformers, many transformer-based models have been proposed to learn contextual word embeddings, including OpenAI-GPT3 (Brown et al., 2020), BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). We refer the reader to survey papers on language models (Qiu et al., 2020; Gou et al., 2021) for more details.

Recently, Barbieri et al. (2020) proposed a new Twitter benchmark data set to evaluate classification models. They provided a pre-trained RoBERTa-based model where their training data set contained 60M English tweets collected from May 2018 to August 2019. Nguyen et al. (2020) developed a set of two pre-trained RoBERTa models on Twitter data collected from 2012 to 2019. The first model was trained on 850M English tweets where initialized weights are from "cased"-sensitive RoBERTa$_{base}$. The second model was re-trained from the first model with 23M COVID-19 tweets.

We focus on these works because we take the same approach, but in a domain specific context. More specifically, we pre-train "cased" models from BERTweet using political tweets.[2] There are a number of reasons for choosing BERTweet's starting weights. First, BERTweet has MLM as its objective function. MLM is more suitable for Twitter as opposed to NSP for pre-training because most tweets are short and composed of a single sentence. Second, BERTweet follows the RoBERTa approach, optimizing the BERT pre-training approach for more robust performance. Third, we assume their models have adequately learned language structure used on Twitter. Our goal is to enhance the model to better represent the US 2020 election period. Towards that end, we evaluate and compare our model to three other models on different NLP tasks and metrics described in Section 4. The models we compare to are: (i) the original RoBERTa$_{base}$ by Liu et al. (2019) (RB), (ii) the pre-trained model from TweetEval by Barbieri et al. (2020) (TE), and (iii) BERTweet by Nguyen et al. (2020) (BT).

3. Resource Construction Details

3.1. Data Collection and Preprocessing

Using the Twitter Streaming API and the Twitter Decahose API,[3] our research team collected Twitter data that contained election and politics-related hashtags and keywords. The set of keywords and hashtags were identified by political and social scientists on our team. They are presented in Table 1 along with the tweet distribution for each keyword. We organize our keywords and hashtags into three categories: candidates, political parties/leanings, and election. These keywords and hashtags are not meant to be a comprehensive list, but rather a representative one.

During the US 2020 presidential election period from January 2020 to February 2021, we collected over 300 million politics-related posts (tweets, retweets and quotes). We preprocessed data by replacing all Twitter usernames with a special token *@USER* and all URLs by *HTTPURL*. We also used the Python `emoji` library to substitute all emoticons with textual ASCII representations (e.g. :smile: for ☺).[4] Next, we applied `TweetTokenizer` from NLTK (Bird and Loper, 2004) for the tokenization step. Finally, we removed non-English and duplicated tweets, retweets and quoted tweets.

Our final data set contains over 83M original and unique English posts (Poli-Full). It contains approximately 1B word tokens and 18B sub-words after tokenization using a vocabulary of 64,000 subword types.[5] A set of 10,000 unique political tweets were sampled

[2] Similar to the previous work, we do not provide an "uncased" model.

[3] The Decahose API is the enterprise-level API managed by Twitter. It delivers a 10% random sample of the real-time Twitter data via a real-time sampling algorithm. See (Twitter, 2021) for more detail.

[4] https://pypi.org/project/emoji

[5] We use the byte-pair-encoding algorithm (BPE) (Sennrich et al., 2016) for the subword tokenization as recommended for RoBERTa-based models (Liu et al., 2019; Nguyen et al., 2020). It tokenizes a sentence into a list of subwords instead of words to mitigate the out-of-vocabulary issue.

	Keyword/Hashtag	Count
Candidates	trump	56,542,935
	biden	31,377,347
	#joebiden	482,903
	#biden2020	328,037
	#donaldtrump	323,607
	#dumptrump	158,569
	#fucktrump	97,800
	#nevertrump	32,920
	#lovetrump	1,093
Parties	#gop	378,131
	#democrat	104,129
	#republican	100,881
	#conservative	42,647
	#liberal	29,795
Election	#voteblue	238,502
	#rnc2020	210,862
	#voteblue2020	143,672
	#dnc2020	104,419
	#dnc	70,534
	#notmypresident	68,037
	#qanon	64,099
	#bluewave	62,429
	#votebluenomatterwho	44,145
	#makeamericagreatagain	37,913
	#democraticnationalconvention	33,109

Table 1: Keywords and hashtags used to collect tweets. They are organized by three categories: candidates, political parties/leaning, and election.

from Poli-Full (Poli-Test). The remaining tweets were used for training our language model. We refer to the sample of Poli-Full without Poli-Test as Poli-Large. We note that there is no overlap between Poli-Test and Poli-Large. Finally, we have a set of 10,000 unique, randomly selected non-political tweets (NonPoli-Test) from the Decahose API that do not contain election and politics-related hashtags and keywords.

3.2. Pre-training Settings

Because the Twitter content is limited to 280 characters, we reduce the sequence length to 128 tokens for a larger batch size of 8 per GPU. A constant learning rate of *2e-5* is used as recommended by the official BERT website.[6] We use the AdamW as opposed to the ordinary Adam optimizer as it is recommended for weight decay optimization (Loshchilov and Hutter, 2019). For the rest of the parameters, we use the default parameter settings for RoBERTa[7] with some modifications as suggested in the pre-training strategy for a BERT-like model recently proposed by Izsak et al. (2021). We perform a pre-training process in the distributed environment using PyTorch v1.8.1 and TensorFlow 2.2 on 2 GPUs (16GB of RAM) for more than 2M steps. Completing this took more than 1,000 hours (or approximately 42 days).

[6] https://github.com/google-research/bert

[7] Please refer to the original paper for more detail (Liu et al., 2019).

4. Model Evaluation

We evaluate our models using perplexity (classic NLP metric), masked token prediction (classic NLP task), and stance detection (important political classification task).

4.1. Perplexity of Language Model

While there are many different metrics for evaluating language models, one common technique is perplexity (PPL) (Huyen, 2019). Intuitively, it measures how well a probabilistic model predicts a sample. It is defined as the exponential average negative log-likelihood of a sequence. Formally, the perplexity score of a tokenized sequence $X = \{x_1, x_2, ..., x_t\}$ is defined as:

$$\text{PPL}(X) = \exp\left(-\frac{1}{t}\sum_{i=1}^{t}\log p_\theta(x_i|x_{<i})\right) \quad (1)$$

Here, t is the number of tokens of the sequence X and $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the i^{th} token conditioned on the preceding tokens as determined by a language model. A smaller perplexity score indicates that there are fewer choices, and therefore, a better model.

We compute the perplexity of the 10,000 tweets in Poli-Test and the 10,000 tweets in NonPoli-Test data sets. Table 2 shows the average perplexity scores of all the sequences of each data set. The table also shows that our model, PoliBERTweet, outperforms the other baseline models with the perplexity score of 4.4846 and 13.1037 on the political and non-political data sets, respectively. For the political data set, PoliBERTweet's perplexity score is almost half that of the next best score. We see that RoBERTa has a worse perplexity score than PoliBERTweet, and hypothesize that this occurs because RoBERTa was trained on news, books, and web data, while the other models were trained on Twitter data. Interestingly, BERTweet performs much better than TweetEval in terms of perplexity for both data sets. We believe this occurs because BERTweet was trained on a larger data set and uses different preprocessing methods than TweetEval.

Model	Political	Non-political
RoBERTa	25.1158	37.8536
TweetEval	18.4893	24.4660
BERTweet	8.0463	13.5791
PoliBERTweet	**4.4846**	**13.1037**

Table 2: Perplexity scores of language models for political (Poli-Test) and non-political (NonPoli-Test) data sets. Lower scores are better. The best scores are bolded.

4.2. Masked Token Prediction

We now compare the language model on a masked token prediction task. For this task, we randomly replace a token from each tweet with a special token <mask>. The goal is to predict the masked token given the modified tweet. Once again, we use the Poli-Test and NonPoli-Test data sets for our evaluation. We generate three augmented tweets per tweet with different masked tokens to understand the precision of each of the models. This produces 30,000 tweets in the result set for each language model.

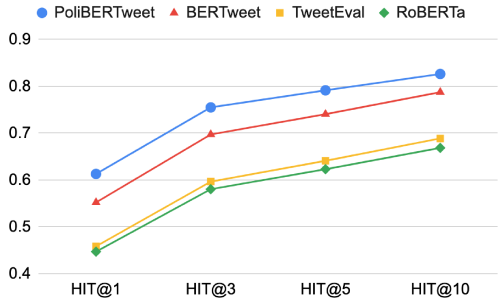
We design our evaluation as a ranking task, where for each tweet t_i, the model ranks the list of potential tokens, L_i, in decreasing order of relevancy. We evaluate the models using the *Hits@k* metric, a standard metric in information retrieval for ranking search results. It calculates the rate of correctly predicted tokens appearing in the top k of the sorted prediction list L_i. The *Hits@k* scores are calculated as shown in Equation 2, where N is the number of tweets and $\mathbb{I}[L_i \leq k]$ is 1 if the correct predicted token appears in the top k of L_i, otherwise the value is 0.

$$\text{score}_k = \frac{1}{N}\sum_{i}^{N}\mathbb{I}[L_i \leq k] \quad (2)$$

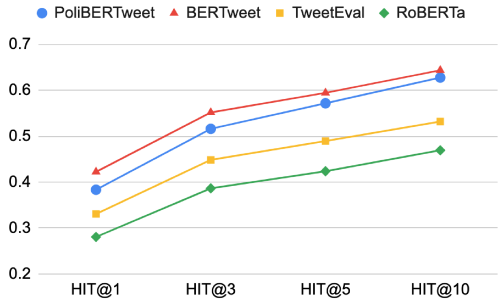
We consider the following values for $k : \{1, 3, 5, 10\}$. The results are shown in Figure 1. The x-axis shows the different k values and the y-axis shows score_k. We see that PoliBERTweet outperforms the other models on the political data by 4 to 17%. The second-best model is BERTweet. We hypothesize that because BERTweet is trained on a large Twitter corpus that contains tweets from two US elections (2012 and 2016), it has a moderate level of political semantic understanding. TweetEval and RoBERTa perform much worse than the best two, with TweetEval performing a little better. On the non-political data, BERTweet performs best with our model performing comparably, followed by TweetEval and RoBERTa. This is not surprising since BERTweet is generated using a broader, more general set of tweets.

4.3. Stance Detection

We now evaluate our model on a stance prediction task. While sentiment analysis aims to predict the overall tone of a post, this task aims to determine whether the author of the post has a favorable, negative or neutral position towards a specific target. For example, the statement *"Trump is doing a poor job handling the pandemic"* is a negative stance statement toward Trump, while *Trump is a strong leader* is a positive stance statement toward Trump. We use a stance-labeled data set that focuses on analyzing Twitter post content about the US 2020 election (Kawintiranon and Singh, 2021). The data set contains two sets of tweets and stance labels, one set focused on stance about Donald Trump and the other focused on stance about Joe Biden. These tweets were manually labeled by three



(a) Political Tweets



(b) Non-political Tweets

Figure 1: The performance of the language models on the masked token prediction task.

	Split	Total	Support	Oppose	Neutral
Biden	Train	875	266	279	330
	Test	375	112	106	157
Trump	Train	875	243	347	285
	Test	375	98	152	125

Table 3: Stance distribution for labeled Biden and Trump tweets.

MTurk annotators (see the original paper by Kawintiranon and Singh (2021) for more details). In total, there are 1250 stance labeled tweets for each candidate. We setup our experiment similar to the prior work, using 875 tweets as the training set and 375 tweets as the test set. The distribution of class labels is shown in Table 3. For this experiment, we not only compare the use of PoliBERTweet against the other language models, but we also fine-tune the other language models we compare to with a subsample of Poli-Large that contains 5 million tweets. We refer to this subsample as Poli-Medium. The reason we create these new models is two-fold. First, it allows us to determine if training on a significantly larger data set is beneficial for the stance detection tasks. Second, it gives us the opportunity to see if any general model will benefit from some additional domain-specific pre-training for domain specific stance detection. In other words, is it sufficient to just pre-train existing models on a more modest size domain-specific data set, or does a domain-specific model trained on significantly more domain-specific data perform better on this downstream task.

We employ the `transformer` library (Wolf et al., 2020) to fine-tune the classification layers of all models using the stance data sets. The output from the language model is input into a feed-forward neural network for the classification task. The architecture of the neural model is shown in Figure 2. A sentence is tokenized by the tokenizer of the specific language model into a list of tokens and used as input for the classification layers. After the stack of transformers, we use a dropout rate of 0.1 in order to avoid over-fitting. We then feed the output vectors into a feed-forward neural network with softmax.

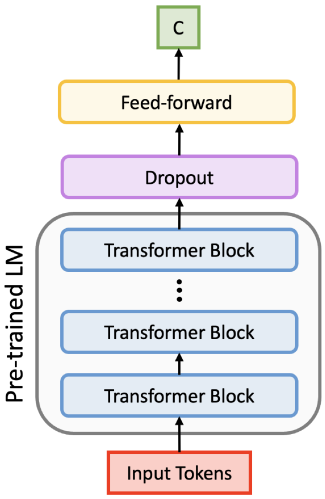


Figure 2: The structure of the neural network model for the stance classification task.

The classifier is a neural network model as shown in Equation 3, where y represents the output vector from the classifier, W_i is a weight vector at layer i randomly initialized, x represents a contextual representation vector from a language model after the dropout layer, and b_i is a bias vector at layer i where $i \in \{1, 2\}$. The weights of the classifier are updated using the cross-entropy loss function as shown in Equation 4. The class label C is calculated using the softmax function as shown in Equation 5. We use it to normalize the values of the output vector y from the classifier in order to obtain a probability score for each class.

$$y = W_2(\tanh(W_1 x^T + b_1)) + b_2 \tag{3}$$

$$Loss(y, class) = -y[class] + \log\left(\sum_j \exp\left(y[j]\right)\right) \tag{4}$$

$$C = \underset{j}{\operatorname{argmax}}\left(\operatorname{softmax}(y)\right) \tag{5}$$

We set the number of epochs to 30, use the AdamW optimizer (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with learning rates of {1e-3, 1e-4, 1e-5, 1e-6} and a batch size of 32. We fine-tune each model three times with three different seeds per task with early stopping. In each run, we split the full training set with a ratio of 90/10 to generate the training and validation sets for the model training. We select the highest performing learning rate on the validation set and use the corresponding model to evaluate the test set. We use macro-averaged F1 to evaluate the models since that was the metric used in the original paper (Kawintiranon and Singh, 2021). The average scores over three runs are presented in Figure 3.

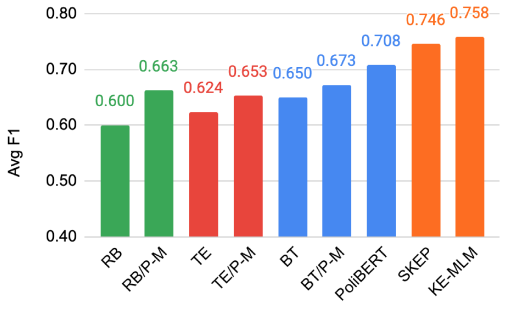
The results show that irrespective of the original model, pre-training on political tweets (Poli-Medium) leads to better performance on both the Biden and Trump data sets with an average F1 score that is 2 to 6% higher for the Biden data set and up to 3% higher for the Trump data set. We also find that our model trained on the full-size political data (PoliBERTweet) outperforms the other pre-trained baselines with an average F1 score that is 3 to 10% higher and 2 to 7% higher for Biden and Trump data sets, respectively. We hypothesize that the reason the F1 scores are all higher on the Trump data set compared to the Biden one is because there was more conversation on Twitter about Trump than Biden in our data set as well as in the data sets we are comparing to.

Our model also outperforms the state-of-the-art models, KE-MLM (Kawintiranon and Singh, 2021) and SKEP (Tian et al., 2020), on the Trump data set by 6 to 8%. On the Biden data set, KE-MLM performs better than our model by 5%. We would expect both KE-MLM and SKEP to perform better on both data sets since they are specifically designed for the stance detection task. We hypothesize that the KE-MLM could even improve their results if the model was trained using PoliBERTweet instead of the vanilla BERT.

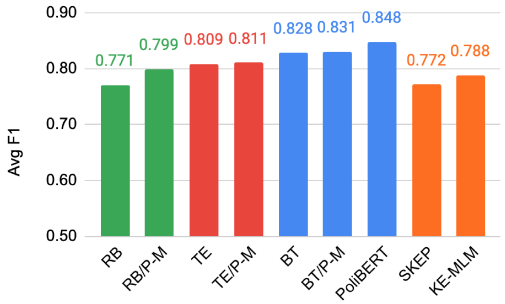
5. Conclusion

Language models play a crucial role in Twitter content analysis. There are many pre-trained language models built using Twitter data, including TweetEval and BERTweet. However, because they are general-purpose, they lack a strong semantic understanding of domain specific knowledge. In this paper, we investigate whether or not a domain specific language model for US politics is useful for classic NLP tasks. We selected politics because of the continued interdisciplinary focus on the 2020 US Presidential election. Some NLP tasks that may benefit from domain-specific knowledge include sentiment, stance detection, and disinformation detection.

Towards that end, we present PoliBERTweet, a pre-trained language model trained on a large corpus of political tweets for over 1000 hours using the starting weights from BERTweet. We evaluate PoliBERTweet and other state-of-the-art models on different NLP metrics and tasks. We find that PoliBERTweet performs better on all the tasks when focused on the political domain, and performs comparable to BERTweet on more



(a) Biden



(b) Trump

Figure 3: The performance of fine-tuned models on stance detection for both US Presidential candidates. The prefixes are model abbreviations described in Section 2 including RoBERTa (RB), TweetEval (TE), BERTweet (BT), our PoliBERTweet (PoliBERT), sentiment knowledge enhanced pre-training (SKEP) and knowledge enhanced masked language modeling (KE-MLM). P-M indicates Poli-Medium.

general domain tasks. This indicates that our model is not only valuable for politics-related domains but also general semantic understanding on Twitter. We make PoliBERTweet and other pre-trained models used for our stance detection evaluation publicly available, enabling researchers to use our model for conducting NLP research when using Twitter data for tasks involving the analysis of election tweets.

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.

6. References

ALDayel, A. and Magdy, W. (2021). Stance detection on social media: State of the art and trends. *Information Processing & Management*, 58(4):102597.

Barbieri, F., Camacho-Collados, J., Anke, L. E., and Neves, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*.

Bird, S. and Loper, E. (2004). NLTK: The natural language toolkit. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Interactive Poster and Demonstration Sessions (ACL)*.

Bode, L., Budak, C., Ladd, J. M., Newport, F., Pasek, J., Singh, L. O., Soroka, S. N., and Traugott, M. W. (2020). *Words that matter: How the news and social media shaped the 2016 Presidential campaign*. Brookings Institution Press.

Bozarth, L. and Budak, C. (2020). Toward a better performance evaluation framework for fake news classification. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Budak, C., Goel, S., and Rao, J. M. (2016). Fair and balanced? quantifying media bias through crowdsourced content analysis. *Public Opinion Quarterly*, 80(S1):250–271.

Budak, C., Kawintiranon, K., Singh, L., and Soroka, S. (2020). Real-time analysis shows that the first debate shifted attitudes among twitter users towards biden and the second solidified them. *USApp–American Politics and Policy Blog*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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: Human Language Technologies (NAACL)*.

Giachanou, A. and Crestani, F. (2016). Like it or not: A survey of twitter sentiment analysis methods. *ACM Computing Surveys (CSUR)*, 49(2):1–41.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Hardalov, M., Arora, A., Nakov, P., and Augenstein, I. (2021). A survey on stance detection for mis- and disinformation identification. *arXiv preprint arXiv:2103.00242*.

Hossain, T., Logan IV, R. L., Ugarte, A., Matsubara, Y., Young, S., and Singh, S. (2020). COVIDLies: Detecting COVID-19 misinformation on social media. In *Proceedings of the Workshop on NLP for COVID-19 (Part 2) at EMNLP*.

Huyen, C. (2019). Evaluation metrics for language modeling. *The Gradient*.

Izsak, P., Berchansky, M., and Levy, O. (2021). How to train BERT with an academic budget. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jiang, S., Metzger, M., Flanagin, A., and Wilson, C. (2020a). Modeling and measuring expressed (dis) belief in (mis) information. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*.

Jiang, Z.-H., Yu, W., Zhou, D., Chen, Y., Feng, J., and Yan, S. (2020b). ConvBERT: Improving BERT with span-based dynamic convolution. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., and Levy, O. (2019). SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics (TACL)*, 8:64–77.

Kawintiranon, K. and Singh, L. (2021). Knowledge enhanced masked language model for stance detection. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Ke, P., Ji, H., Liu, S., Zhu, X., and Huang, M. (2020). SentiLARE: Sentiment-aware language representation learning with linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Nguyen, D. Q., Vu, T., and Nguyen, A. T. (2020). BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*.

Oshikawa, R., Qian, J., and Wang, W. Y. (2020). A survey on natural language processing for fake news detection. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Pasek, J., McClain, C. A., Newport, F., and Marken, S. (2020a). Who's tweeting about the president? what big survey data can tell us about digital traces? *Social Science Computer Review*, 38(5):633–650.

Pasek, J., Singh, L. O., Wei, Y., Soroka, S. N., Ladd, J. M., Traugott, M. W., Budak, C., Bode, L., and Newport, F. (2020b). Attention to campaign events: Do twitter and self-report metrics tell the same story? *Big Data Meets Survey Science: A Collection of Innovative Methods*, pages 193–216.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, pages 1–26.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI Blog*.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of the Workshop on Energy Efficient Machine Learning and Cognitive Computing at NeurIPS*.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shaar, S., Babulkov, N., Da San Martino, G., and Nakov, P. (2020). That is a known lie: Detecting previously fact-checked claims. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., and Liu, H. (2020). FakeNewsNet: A data repository with news content, social context and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188.

Tian, H., Gao, C., Xiao, X., Liu, H., He, B., Wu, H., Wang, H., and Wu, F. (2020). SKEP: Sentiment knowledge enhanced pre-training for sentiment analysis. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Twitter. (2021). Decahose API. https:

//developer.twitter.com/en/docs/twitter-api/enterprise/decahose-api/overview/decahose. [Online; Retrieved 10/22/2021].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Wang, H., Can, D., Kazemzadeh, A., Bar, F., and Narayanan, S. (2012). A system for real-time twitter sentiment analysis of 2012 US presidential election cycle. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*.

Zhou, J., Tian, J., Wang, R., Wu, Y., Xiao, W., and He, L. (2020). SentiX: A sentiment-aware pre-trained model for cross-domain sentiment analysis. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.