Team Howard Beale at SemEval-2019 Task 4: Hyperpartisan News Detection with BERT

Osman Mutlu[*]
Koç University
İstanbul, Sarıyer
omutlu@ku.edu.tr

Ozan Arkan Can[*]
Koç University
İstanbul, Sarıyer
ocan13@ku.edu.tr

Erenay Dayanık
University of Stuttgart
Stuttgart
erenaydayanik@gmail.com

Abstract

This paper describes our system for SemEval-2019 Task 4: Hyperpartisan News Detection (Kiesel et al., 2019). We use pretrained BERT (Devlin et al., 2018) architecture and investigate the effect of different fine tuning regimes on the final classification task. We show that additional pretraining on news domain improves the performance on the Hyperpartisan News Detection task. Our system[1] ranked 8th out of 42 teams with 78.3% accuracy on the held-out test dataset.

1 Introduction

With the rapid spread of the Internet and next-generation media development, people started to follow news through the Internet by abandoning de facto sources such as television and radio. Recent studies reveal that 43% of Americans report often getting news online (Shearer and Gottfried, 2017). In parallel with that, there also has been a massive improvement in the NLP research in news domain to keep the content true, fair and unbiased. SemEval-2019 Task 4: Hyperpartisan News Detection, is yet another attempt under this objective. Hyperpartisan is defined as being extremely biased in favor of a political party (Bastos and Mercea, 2017) and the aim of the shared task is to detect hyperpartisan argumentation in news text. Though it is an important task by itself, hyperpartisan argument detection is also considered as a very first step (or even replacement) of fake news detection, because it has been shown by (Potthast et al., 2018) that there is a high positive correlation between having a hyperpartisan argumentation and being fake for news items.

In this shared task, we seek to model this problem as a text classification task. In general, the task aims to label the text in the question with one or more classes or categories. The main question of text classification is how to mathematically represent the words/tokens such that they retain their original meaning in the context they appear. This question has been tried to be answered in many different ways so far. In earlier work, people mainly used the ”bag of words” approach in algorithms such as Naive Bayes, Decision Tree, and SVM. Then, (Mikolov et al., 2013) advanced the field further by introducing word embeddings, capturing a somewhat meaningful representation of words. However, recent studies (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018) showed that contextual word embeddings perform quite better than traditional word embeddings in many different NLP tasks as a result of their superior capacity of meaning representation. Among those, BERT attracts researchers most because of (i) its transformer based architecture enabling faster training and (ii) state of the art results in many different tasks.

Though it is quite new, BERT has been tried in many different domains than the one proposed in Devlin et al. (2018). However, almost all of these studies have two things in common: they don't start training BERT from scratch and the target domain contains very limited data (Zhu et al., 2018; Yang et al., 2019; Alberti et al., 2019). In this study, on the other hand, we address (1) the performance of BERT by comparing its domain specific pre-trained and fine-tuned performances, and (2) in the setting where the target domain has extensively more data. In the following sections, we first summarize the BERT architecture, then give details of shared task data set, and then describe experimental setups we used to train BERT model. In the results section, we compare the performance of BERT under different settings and share our submission results for the shared task.

[*]equal contribution
[1]https://github.com/ozanarkancan/hyperpartisan

2 Method

Transformer[2] (Vaswani et al., 2017) originally came out as a machine translation architecture and it uses the idea of self attention mechanism (Parikh et al., 2016; Lin et al., 2017). It has an encoder-decoder design and both parts use the same novel multi-head attention mechanism. The encoder part takes an input sentence and derives a representation from it using this attention mechanism. Afterwards, the decoder generates the target sentence by performing multi-headed attention over the encoder stack.

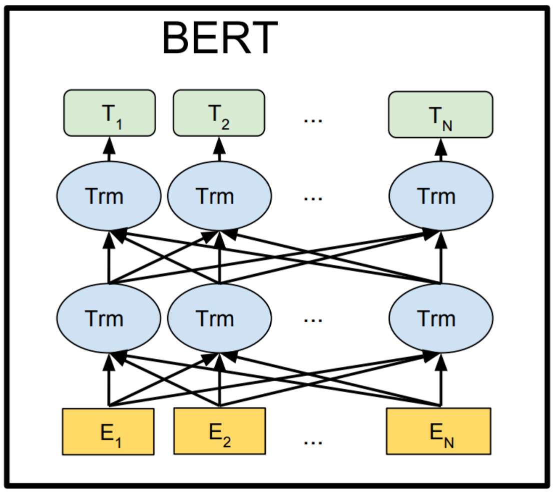

Figure 1: BERT Architecture (Devlin et al., 2018).

Figure 1 illustrates the architecture of the model. BERT learns bidirectional representations jointly on both left and right context of text making use of the encoder part of the Transformer. Devlin et al. (2018) introduced two unsupervised tasks to pretrain this architecture, Next Sentence Prediction and Masked Language Modeling. In Next Sentence Prediction task, the goal is to determine whether the sentence comes after the specified previous sentence or not. It takes two sentences as input, the latter being in its original form 50% of the time, while other times it can be any random sentence from the corpus. In Masked Language Modeling task, 15% of the words in the input sentences are masked and the model tries to predict these words. Training takes place with the combined loss of these two unsupervised tasks. Resulting representations can be further fine-tuned with a task specific layer on the top for a number of NLP tasks using appropriate supervised data.

[2]http://nlp.seas.harvard.edu/2018/04/03/attention.html

In this study, we use an open source PyTorch implementation[3] of BERT architecture. We make use of BERT-Base pretrained model provided by Devlin et al. (2018) in order to avoid pretraining from scratch. Similar to Devlin et al. (2018), we use the representation obtained from the last layer for the first token (i.e. ”[CLS]”) for the sentence representation and a softmax classifier on top of it for predicting hyperpartisanship.

3 Experiments

In this section, we first introduce data provided by the shared task and the data preprocessing step. Then, we give the details of our experiments and results with BERT under pretraining and fine-tuning settings.

3.1 Data

Task provides data that consist of 750.000 articles labelled portal-wise and 645 articles labelled manually, and they divide the former into 600.000 and 150.000 as train and development set. Portal-wise data is labelled as hyperpartisan or not, according to publishers known affinities provided by BuzzFeed journalists or MediaBiasFactCheck.com. In our experiments, we first shuffled and then split the portal-wise data into three: 705.000, 40.000, 5.000 articles for train, development and test respectively.

3.2 Preprocessing

For all our experiments we remove some unwanted text from the articles. We replaced HTML character reference for ampersand and numeric entity reference, and removed adjacent underscore characters which is possibly used as a replacement for classified information in data. We also removed lines, solely containing ”*” characters, used for separation of different news in the same article.

3.3 Input Representation

BERT restricts the input length to a maximum of 512 tokens. We select the first n tokens from the beginning of the article, because using the lead sentences of a news article has been found to be more effective for some NLP tasks (Wasson, 1998). We use the same tokenization method and embeddings as Devlin et al. (2018) to represent the words.

[3]https://github.com/huggingface/pytorch-pretrained-BERT

3.4 Fine-tuning Only

In order to show how BERT performs in news domain, our first attempt was to use the training data to only fine-tune the pretrained model for classification. We used BERT-Base which consists of 12 transformer blocks on top of each other applying 12 headed attention mechanism, hidden size of 768 and a total of 110 million parameters. We set 16 as our batch size and 2e-5 as our learning rate as recommended by Devlin et al. (2018) for fine-tuning on classification tasks.

Max Length	Dev		Test	
	Accuracy	*F1*	*Accuracy*	*F1*
128	84.99	84.91	84.40	84.36
256	88.91	88.89	**88.31**	88.31
512	**89.12**	89.09	88.15	88.14

Table 1: Classification results on our portal-wise data splits with fine-tuned BERT.

We performed experiments using 128, 256 and 512 as our maximum sequence lengths and found out that 256 gives us the best test results, as shown in Table 1. Although the results for experiments with maximum sequence lengths of 256 and 512 are relatively close to each other, we chose 256 for computational efficiency. From these results, we can argue that for news articles, the first 128 tokens do not carry enough information.

3.5 Pretraining + Fine-tuning

For the pretraining step, the data used by two unsupervised tasks need to be generated. For the Next Sentence Prediction task, originally, one would go over the articles sentence by sentence to generate pretraining data, but our data is not made of split sentences. To avoid using a tool for sentence splitting, as it would take too much time in large scale, for each document from the training data, we extract a chunk of text with a random length sampled from a uniform distribution defined as an interval between %15 and %85 of the maximum sequence length. The reason for this is to make the model more robust to non-sentential input and leave space for the second sentence. As the second sentence, 50% of the time, we select the chunk following the original one with a length that is complementing the first chunk's length up to maximum sequence length. Other times, when we need the next sentence to be random, we take a random chunk from other documents. We extract more than one sample from a single document, avoiding overlapping between chunks. For Masked LM task, we follow the same approach with Devlin et al. (2018).

Model	Combined Loss
BERT-Base	3.65
Our Version	**1.79**

Table 2: Results on the held-out dataset for pretraining tasks.

At the end of pretraining data generation process, we accumulated near 3.5 million samples, only running the process once on our train split, so without any duplication unlike Devlin et al. (2018) because of time restrictions. We also generated a small held-out dataset using our test split to use in evaluation. Starting from the pretrained model of BERT-Base instead of a cold start, we trained the model with a learning rate of 3e-5 and 256 as the maximum sequence length for 290k iterations. Table 2 presents the combined loss of two unsupervised tasks on the held-out data for original BERT-Base and further pretrained model with the generated data. Results show that pretraining BERT further with data from an unseen domain greatly increases its representational power.

Model	Dev		Test	
	Accuracy	*F1*	*Accuracy*	*F1*
Fine-Tuning Only	88.91	88.89	88.31	88.31
Pretraining +Fine-Tuning	**89.69**	89.67	**89.30**	89.29

Table 3: Comparison of fine-tuning only and pretraining + fine-tuning models.

After this step, we applied the same fine-tuning as previous section with the same parameters. Table 3 demonstrates that pretraining BERT with domain specific data using unsupervised tasks improves the performance of the model on the supervised classificiation task.

4 Shared Task Results

The evaluation of SemEval-2019 Task 4, Hyperpartisan News Detection task is done through the online platform of TIRA (**?**). It serves as a means of blind evaluation of the submitted model. Accuracy is used as the official evaluation metric and the deciding test set is an another manually labelled news articles set named "by-article-test-set" which was kept hidden from the participants.

Model	article-test				publisher-test			
	Accuracy	*Precision*	*Recall*	*F1*	*Accuracy*	*Precision*	*Recall*	*F1*
Fine-Tuning (publisher) + Fine-Tuning (article)	**78.3**	83.71	70.38	76.5	63.45	67.98	50.85	58.18
Pretraining (publisher) + Fine-Tuning (publisher) + Fine-Tuning (article)	73.4	66.81	92.99	77.76	64.15	60.64	80.6	69.21
Pretraining (publisher) +Fine-Tuning (publisher)	60.82	57.11	86.94	68.93	**67.25**	62.45	86.5	72.53

Table 4: Shared task results.

In our first attempt, we fine-tuned BERT with portal-wise train split using development set to get the best model. After this we further train it with 645 manually labeled data (i.e. ”by-article-train-set”), because it comes from the same sample as test data.

In our last attempt, we pretrained BERT with our portal-wise train split, and then fine-tune it as described before. Again, we further fine-tune our model with ”by-article-train-set” data. The results of our two attempts can be seen in Table 4. The third model in the table is to show the effect of the last fine-tuning step on ”by-article-train-set”.

Looking at the results of second and third models on ”by-article-test-set” shows us, although we fine-tune BERT with supervised data for the same classification task, fine-tuning on ”by-article-train-set” improves the results drastically. This may be rooted from the domain difference in between ”by-article-test-set” and portal-wise train data.

Although our experiments (Table 3) show us that pretraining BERT further with data from news domain has a positive effect on overall accuracy, we are not able to observe the similar effect on ”by-article-test-set”. The second model adapts to the publisher domain more than the first model does because of the extensive pretraining before fine-tuning. As the difference between publisher and article is highly notable from the findings before, overfitting to the publisher domain might end up hurting the generalization of the model. So, this would explain the unexpected drop of performance between the second model and the first model.

5 Conclusion

We presented a BERT baseline for the Hyperpartisan News Detection task. We demonstrated that pretraining BERT in an unseen domain improves the performance of the model on the domain specific supervised task. We also showed that the difference in news source affects the generalization. Our best performing system ranked 8th out of 42 teams with 78.3% accuracy on the held-out test dataset. From our findings, we believe that domain adaptation is important for the BERT architecture and we would like to investigate the effect of from scratch unsupervised pretraining on the supervised task as future work.

Acknowledgments

We would like to thank task organizers for their support. The study is funded by the European Research Council (ERC) Starting Grant 714868.

Howard Beale

Figure 2: Howard Beale delivering his ”I’m as mad as hell” speech.

Beale[4] is a news anchor who decides to commit suicide on live air. Instead, he gives his famous speech about modern American life and convinces American people to scream his words: ”I’m as mad as hell, and I’m not going to take this any more!”. But the media sees his breakdown as an opportunity for huge ratings. We believe that the speech is now more than ever relevant to our media. Choosing ”Howard Beale” as the team name is our scream from the windows of Academia.

[4]https://www.imdb.com/title/tt0074958/

References

Chris Alberti, Kenton Lee, and Michael Collins. 2019. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*. Version 1.

Marco T Bastos and Dan Mercea. 2017. The brexit botnet and user-generated hyperpartisan news. *Social Science Computer Review*, page 0894439317734157.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. Version 1.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 Task 4: Hyperpartisan News Detection. In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval 2019)*. Association for Computational Linguistics.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*. Version 1.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*. Version 3.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. In *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 231–240. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf*.

Elisa Shearer and Jeffrey Gottfried. 2017. News use across social media platforms 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Mark Wasson. 1998. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*, volume 2.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*. Version 1.

Chenguang Zhu, Michael Zeng, and Xuedong Huang. 2018. Sdnet: Contextualized attention-based deep network for conversational question answering. *arXiv preprint arXiv:1812.03593*. Version 5.