Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems Submitted 9/2021; published 11/2021

An explainability analysis of a sentiment prediction task using a transformer-based attention filter

Neşet Özkan Tan NTAN607@AUCKLANDUNI.AC.NZ
Joshua Bensemann JOSH.BENSEMANN@AUCKLAND.AC.NZ
Diana Benavides-Prado DBEN652@AUCKLANDUNI.AC.NZ
Yang Chen YANG.CHEN@AUCKLAND.AC.NZ
Mark Gahegan M.GAHEGAN@AUCKLAND.AC.NZ
Lia Lee JLEE794@AUCKLANDUNI.AC.NZ
Alex Yuxuan Peng YPEN260@AUCKLANDUNI.AC.NZ
Patricia Riddle PAT@CS.AUCKLAND.AC.NZ
Michael Witbrock M.WITBROCK@AUCKLAND.AC.NZ

Abstract

Transformer-based deep learning models have significantly improved the performance of machine learning in many natural language processing tasks in recent years. However, due to the computational complexity of the models' attention mechanisms, the input length they can work with is currently limited. Inspired by the human tendency to ignore many words during a reading comprehension task, we experiment with the effect of removing tokens from a sequence for sentiment analysis. In this preliminary study, we analyze a length-reducing system based on layer-level attention scores. The expectation is that this allows us to discover what percentage of input length is required to obtain a reasonable amount of accuracy in a sentiment analysis task. We show that the filtering system based on BERT allows us to reduce sequences lengths of up to 99% in a sentiment analysis task while still obtaining 70% accuracy.

1. Introduction

Long-Short-Term-Memory (LSTM) (Hochreiter & Schmidhuber, 1997) provides an architecturally integrated memory usage method for neural networks. A weakness of LSTM is memory capacity since each unit is designed to save learnable weights that represent a relationship with the other units (Ba et al., 2016). Large amounts of research in language modeling have been conducted so far to overcome this memory limitation problem. The majority of solutions in recent years have benefited from one well-known architecture called the transformer (Vaswani et al., 2017).

Transformer models are a type of Deep Neural Network (DNN) and are currently the state-of-the-art approach for many Natural Language Processing (NLP) tasks. These models usually take a sequence of text as their input. The input is pre-processed to become word tokens or sub-words (tokens), with each token corresponding to a multi-dimensional vector representation. Like other parameters in the models, these representations change during training time and are stable during inference for downstream tasks. The key novelty in transformer models is the self-attention mecha-

nisms contained within the model architecture. These mechanisms, known as attention heads, measure how important every word in the sequence is to the rest of the words via attention calculations. These calculations work by projecting token vectors into key, query and value vectors then taking the dot product of these projections. For more details about the exact description of self-attention and regularization steps, we refer readers to (Vaswani et al., 2017). Due to the high computational cost of the attention mechanism, transformer-based models usually limit the maximum length of the input sequence (typically to 512 tokens). If l is the sequence of length, then, we initially have a tensor of size $[l, d_e]$, where d_e is the dimension of the word embedding. By considering self-attention calculations among queries and keys, we encounter a computational cost of $\mathcal{O}(l^2 \times d_e)$ in order to obtain an $[l, l]$ tensor, which is called the *attention filter*. This process is repeated in each head across different layers of the models, and so becomes prohibitively expensive.

The leader boards for NLP tasks are dominated by transformer-based models, where many of these models use the famous Bidirectional Encoder Representations from Transformers model, also known as BERT (Devlin et al., 2018). Due to its popularity, BERT is the primary model evaluated in this paper. The BERT model in our experiments has 12 layers and 12 attention heads in each layer. As is true for all transformer-based large-scale language models, BERT also suffers from considerable memory and computational requirements which is summarized above for self-attention with only one head. This limitation inevitably restrict the maximum sequence lengths that the model can be applied to processing.

In general, humans can successfully perform any NLP tasks that a DNN can do. Additionally, humans often generalize better between tasks (although see Firestone, 2020 for a discussion on the fairness of these comparisons). In this paper, we were inspired by methods that humans use to perform these tasks to see if they can improve the performance of DNNs. The particular aspect we focus on is the skim-reading technique that humans use to absorb contexts from the long text (Fitzsimmons et al., 2014). During skim-reading, the reader does not attend to every word fully and learns to skip words of lower importance, with some estimates stating that about 40% of the words can be skipped without substantial loss of understanding (Hahn & Keller, 2018). Skim-reading strategies have been included within previous DNNs. For example, Yu et al. (2017) trained LSTM models to predict how many words can be skipped over within a sequence while performing a sentiment analysis task. Yu et al. (2017) found that the model could skip over several words at a time and still be as accurate or more accurate than the non-skipped models. Hahn & Keller (2018) showed that you could model the skipping processes using human eye movements and achieve the same result for sentiment analysis tasks.

The contribution of this study is our method for identifying word token importance within a sequence and its effect on performance:

- We show that BERT's first layer attention can be used as a filter that gives a remarkably effective sequence selection for the sentiment analysis task.
- We show that the distribution of the parts of speech chosen by the filter changes as the number of filtered tokens increases. We also show that adjectives are the most persistent parts of speech in the filtering progress for the sentiment analysis task.

In the next section, we present related work. We discuss our experiments and results in Section 3 and we conclude this paper in Section 4.

2. Related Work

The main bottleneck of the transformer's memory consumption is in the computation of the attention scores. The original transformer (Vaswani et al., 2017) allowed every token within a sequence to attend to every token of that sequence. This pattern has been referred to as full attention. The disadvantage of this pattern is that both memory and computation time increases quadratically as sequence length grows (Child et al., 2019). One strategy to mitigate this problem is to limit the attention of tokens to neighboring tokens in the sequence. This pattern is known as sparse attention and, although it was initially developed for image processing, it has been successfully applied to language models (Beltagy et al., 2020; Gupta & Berant, 2020; Zaheer et al., 2020). This sparse pattern can reduce computation growth from quadratic down to linear, providing considerable computation benefits at the expense of potentially losing important information from other sequence parts. Some work has looked at determining the optimal window size for sparse attention patterns (Sukhbaatar et al., 2019) and has suggested that only a few words tokens on either side of a sequence are required to achieve good performance at language modeling tasks.

DNNs based on the human tendency to skip words show that DNNs also learn to skip many tokens during sentiment analysis tasks Hahn & Keller (2018); Yu et al. (2017). However, the exclusion process in the models described above requires learning a separate model to predict which tokens are necessary and need to be attended to. However, transformers already have an inbuilt attention process that develops during primary task training. Therefore, we were interested in analyzing the self-attention matrix of a pre-trained model to see if that matrix can also be used to identify which words can be ignored. By learning which words can be ignored, it is possible to identify which words could be filtered out of a sequence, allowing sequences that are too long for input to be reduced to a manageable size.

Previous studies have explored the relationship between the words attended by humans during reading and words attended by DNNs during related NLP tasks. Sood et al. (2020) compared the attention patterns created by CNNs, LSTMs, and XLNet (Yang et al., 2019) during reading comprehension tasks to the eye movements of human participants who were performing the same tasks. Results indicated that human attention was more strongly correlated to the CNN and LSTM models than to XLNet. However, XLNet outperformed the other DNNs on the actual reading comprehension task.

The results of Sood et al. (2020) suggest that there is a weak relationship between human attention and transformer attention. However, this does not mean that transformer attention cannot be used to determine which words can be skipped during an NLP task performed by a transformer. In this paper, we have explored this idea by extracting the attention matrix produced by BERT (Devlin et al., 2018) to identify which tokens can be removed to shorten a sequence and observe how that affects model performance for the trained task.

There are reasons to believe that BERT can identify the importance of each word in a sequence. For example, Li et al. (2020) showed that BERT could be used to produce adversarial examples for

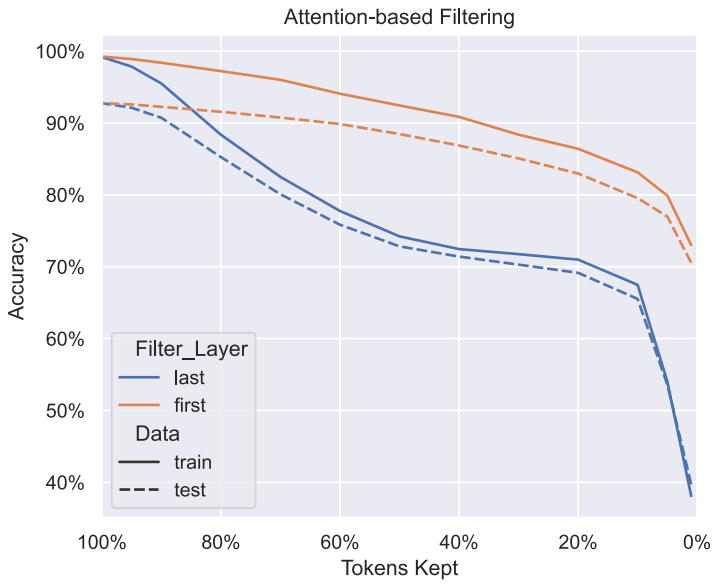

Figure 1. Results from then attention-based filtering using fine-tuned BERT model. Blue indicates filtering based on the first BERT encoder, Orange indicates filtering based on the last layer. Solid and dashed lines indicate performance on the training and testing datasets, respectively.

other BERT models. To do this, Li et al. used the BERT model to find words of high importance and then replace them so that the sentences maintained semantic consistency. Their results show that the adversarial examples cause large decreases in the model's performance.

On the other end of the scale, BERT has also been used to predict human attention via eye movement (Choudhary et al., 2021). To create these predictions, BERT embeddings were combined with other word features, such as syllables per word, and inputted into various machine learning methods such as LGBM and DNNs. While our goal is to simulate the human tendency to skip words while reading using BERT, it is interesting to note that it is also possible to predict aspects of human behavior using BERT.

3. Experiments and Results

The IMDB dataset (Maas et al., 2011) was used during these experiments. The dataset contains movie reviews collected from www.imdb.com along with a binary label indicating whether the review contained a positive or negative sentiment. There are 50000 examples within the dataset, where half were used for training and the other half for testing.

We first explored the removal of tokens from a sequence. Our interest was to see whether we could identify the essential tokens from a sequence by observing changes in accuracy after systematic removal of tokens based on the attention those tokens receive within a transformer encoder. We

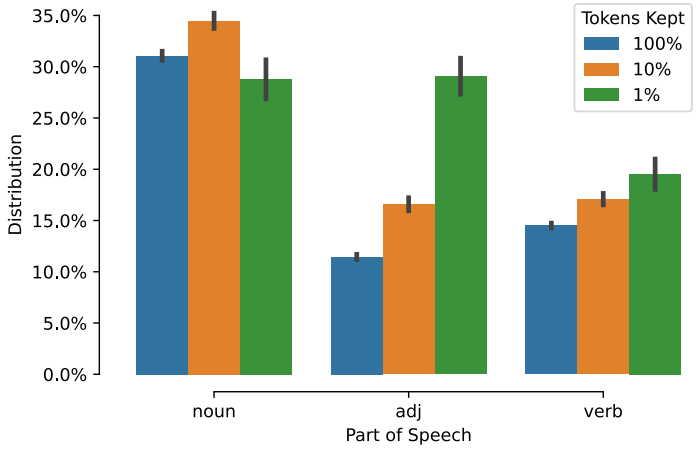

Figure 2. The distribution of different parts of speech present in the samples after filtering by the first layer.

conducted a test of attention-based filtering using a pre-trained BERT transformer (Devlin et al., 2018). The Bert-Base-Cased model was fine-tuned to predict the sentiments of reviews collected from IMDB. After training, each sample from the training and testing sets was inputted into our model.

The attention weights generated from each sample were extracted from either the first or last layers from BERT's encoder. The attention weights from the 12 attention heads were added together, resulting in a single attention matrix. The sum of each of the matrix's columns was then calculated, creating an attention score for each token.There are two special tokens in each sequence, namely CLS and SEP tokens to indicate start and end of sequence. The CLS and SEP tokens were removed, and the top X percentile of tokens was selected. The CLS and SEP tokens were appended on the start and end of the new sequence, respectively, and the sequence was input into the model to predict its sentiment.

Figure 1 shows the effects of filtering tokens based on the attention values from either the first or last layer from the fine-tuned BERT model. After fine-tuning, the model achieved 99.3% accuracy on the training set and 92.7% accuracy in the testing set. The effects of filtering depended on which of the layers was used to select the tokens. Filtering using the attention weights extracted from the first layer (blue line) was more effective than filtering by the last layer (orange line). Using the first layer allowed us to select the top 1% of the tokens (i.e., filter out the bottom 99%) and still achieve 73.0% accuracy on the training set and 70.6% accuracy on the testing set. Performing the same selection via the last layer resulted in below chance performance of less than 40% on both the training and testing datasets.

We were interested in which tokens were kept by the filtering process. We looked at how the distribution of noun, verb, and adjectives changed based on filtering. Parts of speech were identified using SpaCy (Honnibal et al., 2020) as a tokenizer. Figure 2 shows that while the proportion of

nouns and verbs are similar regardless of the amount of filtering, the proportion of adjectives dramatically increases. We also discovered that the increase in adjectives is due to a decrease in the parts of speech that are not nouns or verbs.

4. Conclusion

There have been several studies focused on computational reduction methods for long sequences in the transformer architecture. In this study, we have taken a simple approach. Rather than modifying the original architecture, we used the original model components to understand how much content may be needed for a sentiment analysis task based on attention scores using only the first and last layer. We also examined these selections from a part of speech (POS) perspective, which makes selection progress linguistically explainable.

Our experiments show that we sacrifice only about 2% of accuracy when we reduce the length of the sequences by half according to first layer attention scores. Interestingly, we find that we can ignore 99% of the tokens in the given sequence and still achieve 73.0% accuracy based on the first layer selection. In addition, we find that the adjectives are generally retained in the selection progress, which makes intuitive sense from a human skim-reading perspective. The selection of adjectives is somewhat remarkable when we eliminate 99% of the tokens.

As future work, we intend to extend our analysis by experimenting with different downstream tasks and models to benefit from reduced fine-tuning costs. Meanwhile, however, we hope that this preliminary study can offer a new approach for researchers studying the long-distance-dependency problems in transformers.

References

Ba, J., Hinton, G., Mnih, V., Leibo, J. Z., & Ionescu, C. (2016). Using fast weights to attend to the recent past.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Choudhary, S., Tandon, K., Agarwal, R., & Chatterjee, N. (2021). Mtl782_iitd at cmcl 2021 shared task: Prediction of eye-tracking features using bert embeddings and linguistic features. *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics* (pp. 114–119).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Firestone, C. (2020). Performance vs. competence in human–machine comparisons. *Proceedings of the National Academy of Sciences*, *117*, 26562–26571.

Fitzsimmons, G., Weal, M., & Drieghe, D. (2014). Skim reading: an adaptive strategy for reading on the web.

Gupta, A., & Berant, J. (2020). Gmat: Global memory augmentation for transformers. *arXiv preprint arXiv:2006.03274*.

Hahn, M., & Keller, F. (2018). Modeling task effects in human reading with neural attention. *arXiv preprint arXiv:1808.00054*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*, 1735–1780.

Honnibal, M., Montani, I., Van Landeghem, S., & Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python. From `https://doi.org/10.5281/zenodo.1212303`.

Li, L., Ma, R., Guo, Q., Xue, X., & Qiu, X. (2020). Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. *Learning Word Vectors for Sentiment Analysis*, (pp. 142–150).

Sood, E., Tannert, S., Frassinelli, D., Bulling, A., & Vu, N. T. (2020). Interpreting attention models with human visual attention in machine reading comprehension. *arXiv preprint arXiv:2010.06396*.

Sukhbaatar, S., Grave, E., Bojanowski, P., & Joulin, A. (2019). Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems* (pp. 5998–6008).

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

Yu, A. W., Lee, H., & Le, Q. V. (2017). Learning to skim text. *arXiv preprint arXiv:1704.06877*.

Zaheer, M., et al. (2020). Big bird: Transformers for longer sequences. *NeurIPS*.