Pretrained Language Models for Sequential Sentence Classification

Arman Cohan [1★] **Iz Beltagy** [1★] **Daniel King** [1] **Bhavana Dalvi** [1] **Daniel S. Weld** [1,2]

[1]Allen Institute for Artificial Intelligence, Seattle, WA

[2]Allen School of Computer Science & Engineering, University of Washington, Seattle, WA

{armanc,beltagy,daniel,bhavanad,danw}@allenai.org

Abstract

As a step toward better document-level understanding, we explore classification of a sequence of sentences into their corresponding categories, a task that requires understanding sentences in context of the document. Recent successful models for this task have used hierarchical models to contextualize sentence representations, and Conditional Random Fields (CRFs) to incorporate dependencies between subsequent labels. In this work, we show that pretrained language models, BERT (Devlin et al., 2018) in particular, can be used for this task to capture contextual dependencies without the need for hierarchical encoding nor a CRF. Specifically, we construct a joint sentence representation that allows BERT Transformer layers to directly utilize contextual information from all words in all sentences. Our approach achieves state-of-the-art results on four datasets, including a new dataset of structured scientific abstracts.

1 Introduction

Inspired by the importance of document-level natural language understanding, we explore classification of a sequence of sentences into their respective roles or functions. For example, one might classify sentences of scientific abstracts according to rhetorical roles (e.g., Introduction, Method, Result, Conclusion, etc). We refer to this task as Sequential Sentence Classification (SSC), because the meaning of a sentence in a document is often informed by context from neighboring sentences.

Recently, there have been a surge of new models for contextualized language representation, resulting in substantial improvements on many natural language processing tasks. These models use multiple layers of LSTMs (Hochreiter and Schmidhuber, 1997) or Transformers (Vaswani et al., 2017), and are pretrained on unsupervised text with language modeling objectives such as next word prediction (Peters et al., 2018; Radford et al., 2018) or masked token prediction (Devlin et al., 2018; Dong et al., 2019). BERT is among the most successful models for many token- and sentence-level tasks (Devlin et al., 2018; Liu et al., 2019). In addition to a masked token objective, BERT optimizes for next sentence prediction, allowing it to capture sentential context.

These objectives allow BERT to learn some document-level context through pretraining. In this work we explore the use of BERT for SSC. For this task, prior models are primarily based on hierarchical encoders over both words and sentences, often using a Conditional Random Field (CRF) (Lafferty et al., 2001) layer to capture document-level context (Cheng and Lapata, 2016; Jin and Szolovits, 2018; Chang et al., 2019). These models encode and contextualize sentences in two consecutive steps. In contrast, we propose an input representation which allows the Transformer layers in BERT to directly leverage contextualized representations of all words in all sentences, while still utilizing the pretrained weights from BERT. Specifically, we represent all the sentences in the document as one long sequence of words with special delimiter tokens in between them. We use the contextualized representations of the delimiter tokens to classify each sentence. The transformer layers allow the model to finetune the weights of these special tokens to encode contextual information necessary for correctly classifying sentences in context.

We apply our model to two instances of the SSC task in scientific text that can benefit from better contextualized representations of sentences: scientific abstract sentence classification and extractive summarization of scientific documents.

★Equal contribution.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3693–3699, Hong Kong, China, November 3–7, 2019.

Our contributions are as follows:

(i) We present a BERT-based approach for SSC that jointly encodes all sentences in the sequence, allowing the model to better utilize document-level context. *(ii)* We introduce and release CSABSTRUCT, a new dataset of manually annotated sentences from computer science abstracts. Unlike biomedical abstracts which are written with explicit structure, computer science abstracts are free-form and exhibit a variety of writing styles, making our dataset more challenging than existing datasets for this task. *(iii)* We achieve state-of-the-art (SOTA) results on multiple datasets of two SSC tasks: scientific abstract sentence classification and extractive summarization of scientific documents.[1]

2 Model

In Sequential Sentence Classification (SSC), the goal is to classify each sentence in a sequence of n sentences in a document. We propose an approach for SSC based on BERT to encode sentences in context. The BERT model architecture consists of multiple layers of Transformers and uses a specific input representation, with two special tokens, `[CLS]` and `[SEP]`, added at the beginning of the input sentence pair and between the sentences (or bag of sentences) respectively. The pretrained multi-layer TRANSFORMER architecture allows the BERT model to contextualize the input over the entire sequence, allowing it to capture necessary information for correct classification. To utilize this for the SSC task, we propose a special input representation without any additional complex architecture augmentation. Our approach allows the model to better incorporate context from all surrounding sentences.

Figure 1 gives an overview of our model. Given the sequence of sentences $\mathbf{S} = \langle \mathbf{S_1}, ..., \mathbf{S_n} \rangle$ we concatenate the first sentence with BERT's delimiter, `[SEP]`, and repeat this process for each sentence, forming a large sequence containing all tokens from all sentences. After inserting the standard `[CLS]` token at the beginning of this sequence, we feed it into BERT. Unlike BERT, which uses the `[CLS]` token for classification, we use the encodings of the `[SEP]` tokens to classify each sentence. We use a multi-layer feedforward network on top of the `[SEP]` representations of

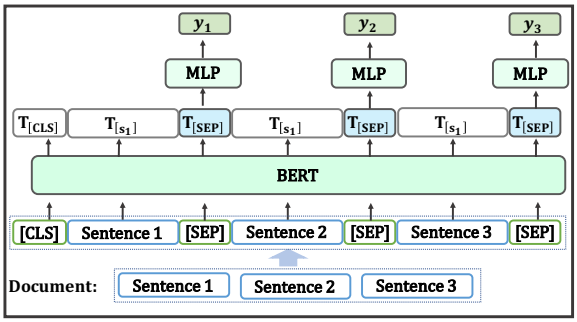


Figure 1: Overview of our model. Each `[SEP]` token is mapped to a contextualized representation of its sentence and then used to predict a label y_i for sentence$_i$.

each sentence to classify them to their corresponding categories.[2] Intuitively, through BERT's pretraining, the `[SEP]` tokens learn sentence structure and relations between continuous sentences (through the next sentence objective). The model is then finetuned on task-specific training data, where most of the model parameters are already pretrained using BERT and only a thin task-specific network on top is needed. During finetuning[3] the model learns appropriate weights for the `[SEP]` token to allow it to capture contextual information for classifying sentences in the sequence. This way of representing a sequence of sentences allows the self-attention layers of BERT to directly leverage contextual information from all words in all sentences, while still utilizing the pretrained weights from BERT. This is in contrast to existing hierarchical models which encode then contextualize sentences in two consecutive steps.[4]

Handling long sequences Released BERT pretrained weights support sequences of up to 512 wordpieces (Wu et al., 2016). This is limiting for our model on datasets where the length of each document is large, as we represent all sentences in one single sequence. However, the semantics of a sentence are usually more dependent on local context, rather than all sentences in a long docu-

[1]Code & data: https://github.com/allenai/sequential_sentence_classification

[2]It is also possible to add another special token (e.g., `[CLS]`) at the beginning of each sentence and perform classification on that token. Empirically, we found the approaches to perform similarly.

[3] Following terminology from Howard and Ruder (2018), "finetuning" refers to "training" a model that was previously pretrained. We use both terms interchangeably.

[4]It is possible to add a CRF layer or another contextualizing layer on top of `[SEP]` tokens in our model, but empirically, we did not find this addition to be helpful. One explanation is that the self-attention layers of our model are already capturing necessary contextual information from the document.

	PubMed	NICTA	CSAbstruct	CSPubSum
# docs	20K	1K	2.2K	21K
# sents	225K	21K	15K	601K

Table 1: Statistics of the evaluation datasets. The first three datasets are for the abstract sentence classification task and the last dataset is for summarization.

CSAbstruct characteristics		
Doc length (sentences)	avg : 6.7	std : 1.99
Sentence length (words)	avg : 21.8	std : 10.0
Label distribution	BACKGROUND	0.33
	METHOD	0.32
	RESULT	0.21
	OBJECTIVE	0.12
	OTHER	0.03

Table 2: Characteristics of our CSABSTRUCT dataset

ment. Therefore, we set a threshold on the number of sentences in each sequence. We recursively bisect the document until each split has less sentences than the specified threshold. At a limit of 10 sentences, only one division is needed to fit nearly all examples for the abstract sentence classification datasets. A limitation of this approach is that sentences on the edge of the splits could lose context from the previous(next) split. We leave this limitation to future work.

3 Tasks and Datasets

This section describes our tasks and datasets, and any model changes that are task-specific (see Table 1 for comparison of evaluation datasets).

3.1 Scientific abstract sentence classification

This task requires classifying sentences in scientific abstracts into their rhetorical roles (e.g., INTRODUCTION, METHOD, RESULTS, etc). We use the following three datasets in our experiments.

PUBMED-RCT (Dernoncourt and Lee, 2017) contains 20K biomedical abstracts from PubMed, with sentences classified as one of 5 categories {BACKGROUND, OBJECTIVE, METHOD, RESULT, CONCLUSION}. We use the preprocessed version of this dataset by Jin and Szolovits (2018).

CSABSTRUCT is a new dataset that we introduce. It has 2,189 manually annotated computer science abstracts with sentences annotated according to their rhetorical roles in the abstract, similar to the PUBMED-RCT categories. See §3.3 for details.

NICTA (Kim et al., 2011) contains 1,000 biomedical abstracts with sentences classified into PICO categories (Population, Intervention, Comparison, Outcome) (Richardson et al., 1995).

3.2 Extractive summarization of scientific documents

This task is to select a few text spans in a document that best summarize it. When the spans are sentences, this task can be viewed as SSC, classifying each sentence as a good summary sentence or not. Choosing the best summary sentences can benefit from context of surrounding sentences. We train on CSPUBSUMEXT (Collins et al., 2017), an extractive summarization dataset of 10k scientific papers, with sentences scored as good/bad summary sentences using ROUGE overlap scores with paper highlights. For evaluation, a separate test set, CSPUBSUM, of 150 publications and their paper highlights is used.[5]

A key difference between the training of our model and that of Collins et al. (2017) is that they use the ROUGE scores to label the top (bottom) 20 sentences as positive (negative), and the rest are neutral. However, we found it better to train our model to directly predict the ROUGE scores, and the loss function we used is Mean Square Error.

3.3 CSABSTRUCT construction details

CSABSTRUCT is a new dataset of annotated computer science abstracts with sentence labels according to their rhetorical roles. The key difference between this dataset and PUBMED-RCT is that PubMed abstracts are written according to a predefined structure, whereas computer science papers are free-form. Therefore, there is more variety in writing styles in CSABSTRUCT. CSABSTRUCT is collected from the Semantic Scholar corpus (Ammar et al., 2018). Each sentence is annotated by 5 workers on the Figure-eight platform,[6] with one of 5 categories {BACKGROUND, OBJECTIVE, METHOD, RESULT, OTHER}. Table 2 shows characteristics of the dataset. We use 8 abstracts (with 51 sentences) as test questions to train crowdworkers. Annotators whose accuracy is less than 75% are disqualified from doing the actual annotation job. The annotations are

[5] Dataset generated using author provided scripts: https://github.com/EdCo95/scientific-paper-summarisation

[6] http://figure-eight.com

Model	PUBMED	CSABST.	NICTA
Jin and Szolovits (2018)	92.6	81.3	84.7
BERT +Transformer	89.6	78.8	78.4
BERT +Transformer+CRF	92.1	78.5	79.1
Our model	**92.9**	**83.1**	**84.8**

Table 3: Abstract sentence classification (micro F1).

Model	ROUGE-L
SAF + F Ens (Collins et al., 2017)	0.313
BERT +Transformer	0.287
Our model	0.306
Our model + ABSTRACTROUGE	**0.314**

Table 4: Results on CSPUBSUM

aggregated using the agreement on a single sentence weighted by the accuracy of the annotator on the initial test questions. A confidence score is associated with each instance based on the annotator initial accuracy and agreement of all annotators on that instance. We then split the dataset 75%/15%/10% into train/dev/test partitions, such that the test set has the highest confidence scores. Agreement rate on a random subset of 200 sentences is 75%, which is quite high given the difficulty of the task. Compared with PUBMED-RCT, our dataset exhibits a wider variety of writing styles, since its abstracts are not written with an explicit structural template.

4 Experiments

Training and Implementation We implement our models using AllenNLP (Gardner et al., 2018). We use SCIBERT pretrained weights (Beltagy et al., 2019) in both our model and BERT-based baselines, because our datasets are from the scientific domain. As in prior work (Devlin et al., 2018; Howard and Ruder, 2018), for training, we use dropout of 0.1, the Adam (Kingma and Ba, 2015) optimizer for 2-5 epochs, and learning rates of $5e^6$, $1e^5$, $2e^5$, or $5e^5$. We use the largest batch size that fits in the memory of a Titan V GPU (between 1 to 4 depending on the dataset/model) and use gradient accumulation for effective batch size of 32. We report the average of results from 3 runs with different random seeds for the abstract sentence classification datasets to control potential non-determinism associated with deep neural models (Reimers and Gurevych, 2017). For summarization, we use the best model on the validation set. We choose hyperparameters based on the best performance on the validation set. We release our code and data to facilitate reproducibility.[7]

Baselines We compare our approach with two strong BERT-based baselines, finetuned for the task. The first baseline, BERT+Transformer, uses the `[CLS]` token to encode individual sentences as described in Devlin et al. (2018). We add an additional Transformer layer over the `[CLS]` vectors to contextualize the sentence representations over the entire sequence. The second baseline, BERT+Transformer+CRF, additionally adds a CRF layer. Both baselines split long lists of sentences into splits of length 30 using the method in §2 to fit into the GPU memory.

We also compare with existing SOTA models for each dataset. For the PUBMED-RCT and NICTA datasets, we report the results of Jin and Szolovits (2018), who use a hierarchical LSTM model augmented with attention and CRF. We also apply their model on our dataset, CSABSTRUCT, using the authors' original implementation.[8] For extractive summarization, we compare to Collins et al. (2017)'s model, `SAF+F Ens`, the model with highest reported results on this dataset. This model is an ensemble of an LSTM-based model augmented with global context and abstract similarity features, and a model trained on a set of hand-engineered features.

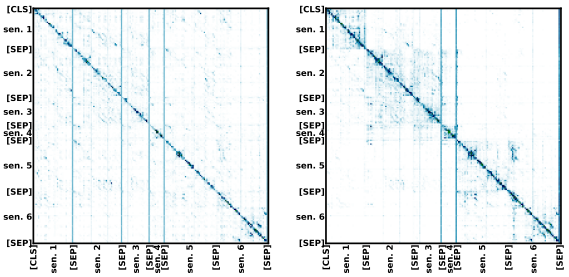
4.1 Results

Table 3 summarizes results for abstract sentence classification. Our approach achieves state-of-the-art results on all three datasets, outperforming Jin and Szolovits (2018). It also outperforms our BERT-based baselines. The performance gap between our baselines and our best model is large for small datasets (CSABSTRUCT, NICTA), and smaller for the large dataset (PUBMED-RCT). This suggests the importance of pretraining for small datasets.

Table 4 summarizes results on CSPUBSUM. Following Collins et al. (2017) we take the top 10 predicted sentences as the summary and use ROUGE-L scores for evaluation. It is clear that our approach outperforms BERT+TRANSFORMER. The BERT +TRANSFORMER+CRF baseline is not included

[7] https://github.com/allenai/sequential_sentence_classification

[8] https://github.com/jind11/HSLN-Joint-Sentence-Classification



(a) Before finetuning (b) After finetuning

Figure 2: Self-attention weights of the top 2 layers of BERT for one abstract. Cell value in row i, column j, is the maximum attention weight of token i attending to token j across all 12 Transformer attention heads.

here because, as mentioned in section 3, we train our model to predict ROUGE, not binary labels as in Collins et al. (2017). As in Collins et al. (2017), we found the ABSTRACT-ROUGE feature to be useful. Our model augmented with this feature slightly outperforms Collins et al. (2017)'s model, which is a relatively complex ensemble model and uses a number of carefully engineered features for the task. Our model is a single model with only one added feature.

Analysis To better understand the advantage of our joint sentence encoding relative to the BERT+Transformer baseline, we qualitatively analyze examples from CSABSTRUCT that our model gets right and the baseline gets wrong. We found that 34/134 of such examples require context to classify correctly.[9]

For example, sentences 2 and 3 from one abstract are as follows: "*We present an improved oracle for the arc-eager transition system, which provides a set of optimal transitions [...].*", "*In such cases, the oracle provides transitions that will lead to the best reachable tree [...].*". In isolation, the label for sentence 3 is ambiguous, but with context from the previous sentence, it clearly falls under the METHOD category.

Figure 2 shows BERT self-attention weights for the above-mentioned abstract before and after finetuning. Before (Figure 2a), attention weights don't exhibit a clear pattern. After (Figure 2b), we observe blocks along the matrix diagonal of sentences attending to themselves, except for the block encompassing sentences 2 and 3. The words in these two sentences attend to each other, enabling the encoding of sentence 3 to capture the information needed from sentence 2 to predict its label (see Appendix A for additional patterns).

[9]Of the 1349 examples in the test set, our model gets 134 correct that the BERT +Transformer baseline gets wrong, and the baseline gets 79 correct that our model gets wrong.

5 Related Work

Prior work on scientific Sequential Sentence Classification datasets (e.g. PUBMED-RCT and NICTA) use hierarchical sequence encoders (e.g. LSTMs) to encode each sentence and contextualize the encodings, and apply CRF on top (Dernoncourt and Lee, 2017; Jin and Szolovits, 2018). Hierarchical models are also used for summarization (Cheng and Lapata, 2016; Nallapati et al., 2016; Narayan et al., 2018), usually trained in a seq2seq fashion (Sutskever et al., 2014) and evaluated on newswire data such as the CNN/Daily mail benchmark (Hermann et al., 2015). Prior work proposed generating summaries of scientific text by leveraging citations (Cohan and Goharian, 2015) and highlights (Collins et al., 2017). The highlights-based summarization dataset introduced by Collins et al. (2017) is among the largest extractive scientific summarization datasets. Prior work focuses on specific architectures designed for each of the tasks described in §3, giving them more power to model each task directly. Our approach is more general, uses minimal architecture augmentation, leverages language model pretraining, and can handle a variety of SSC tasks.

6 Conclusion and Future Work

We demonstrated how we can leverage pretrained language models, in particular BERT, for SSC without additional complex architectures. We showed that jointly encoding sentences in a sequence results in improvements across multiple datasets and tasks in the scientific domain. For future work, we would like to explore methods for better encoding long sequences using pretrained language models.

Acknowledgments

We would like to thank Matthew Peters, Waleed Ammar and Hanna Hajishirzi for helpful discussions, Madeleine van Zuylen for help in crowdsourcing and data analysis, and the three anonymous reviewers for their comments and suggestions. Computations on beaker.org were supported in part by credits from Google Cloud. Other support includes ONR grant N00014-18-1-2193 and the WRF/Cable Professorship,

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler C. Murray, Hsu-Han Ooi, Matthew E. Peters, Joanna Power, Sam Skjonsberg, Lucy Lu Wang, Christopher Wilhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT))*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ming-Wei Chang, Kristina Toutanova, Kenton Lee, and Jacob Devlin. 2019. Language model pre-training for hierarchical document representations. *CoRR*, abs/1901.09128.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *CoRR*, abs/1603.07252.

Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article's discourse structure. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Ed Collins, Isabelle Augenstein, and Sebastian Riedel. 2017. A supervised approach to extractive summarisation of scientific papers. In *CoNLL*.

Franck Dernoncourt and Ji Young Lee. 2017. Pubmed 200k rct: a dataset for sequential sentence classification in medical abstracts. In *IJCNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *ArXiv*, abs/1905.03197.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *ArXiv*, abs/1803.07640.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*.

Di Jin and Peter Szolovits. 2018. Hierarchical neural networks for sequential sentence classification in medical scientific abstracts. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Su Kim, David Martínez, Lawrence Cavedon, and Lars Yencken. 2011. Automatic classification of sentences to support evidence based medicine. In *BMC Bioinformatics*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

John D. Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

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*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2016. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke S. Zettlemoyer. 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-HLT)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Preprint*.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Empirical Methods in Natural Language Processing (EMNLP)*.

William S. Richardson, Mark C. Wilson, James A. Nishikawa, and Robert S. Hayward. 1995. The well-built clinical question: a key to evidence-based decisions. *ACP journal club*, 123 3:A12–3.

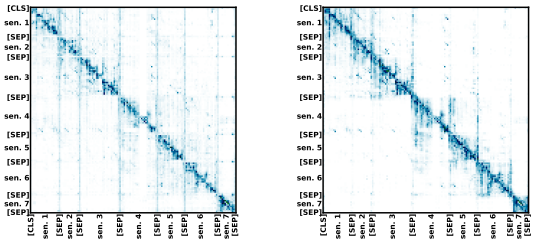
Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

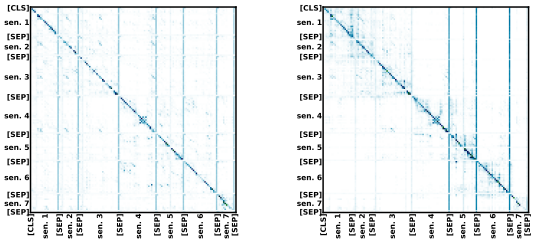
A Additional analysis

Figures 3 and 4 show attention weights of BERT before and after finetuning. We observe that before finetuning, the attention patterns on `[SEP]` tokens and periods is almost identical between sentences. However, after finetuning, the model attends to sentences differently, likely based on their different role in the sentence that requires different contextual information.



(a) Before finetuning (b) After finetuning

Figure 3: Visualization of attention weights for layer 8 of BERT before and after finetuning.



(a) Before finetuning (b) After finetuning

Figure 4: Visualization of attention weights in final layer (layer 12) of BERT before and after finetuning.