

HIERARCHICAL TRANSFORMERS FOR LONG DOCUMENT CLASSIFICATION

Raghavendra Pappagari¹, Piotr Żelasko², Jesús Villalba¹, Yishay Carmiel², and Najim Dehak¹

¹Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD

²Avaya Conversational Intelligence

{rpappag1, jvillal17, ndehak3}@jhu.edu
petezor@gmail.com, ycarmiel@avaya.com

ABSTRACT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a recently introduced language representation model based upon the transfer learning paradigm. We extend its fine-tuning procedure to address one of its major limitations - applicability to inputs longer than a few hundred words, such as transcripts of human call conversations. Our method is conceptually simple. We segment the input into smaller chunks and feed each of them into the base model. Then, we propagate each output through a single recurrent layer, or another transformer, followed by a softmax activation. We obtain the final classification decision after the last segment has been consumed. We show that both BERT extensions are quick to fine-tune and converge after as little as 1 epoch of training on a small, domain-specific data set. We successfully apply them in three different tasks involving customer call satisfaction prediction and topic classification, and obtain a significant improvement over the baseline models in two of them.

Index Terms— Transformer, BERT, Recurrent Neural Networks, Topic Identification, Customer Satisfaction Prediction

1. INTRODUCTION

Bidirectional Encoder Representations from Transformers (BERT) is a novel Transformer [1] model, which recently achieved state-of-the-art performance in several language understanding tasks, such as question answering, natural language inference, semantic similarity, sentiment analysis, and others [2]. While well-suited to dealing with relatively short sequences, Transformers suffer from a major issue that hinders their applicability in classification of long sequences, i.e. they are able to consume only a limited context of symbols as their input [3].

There are several natural language (NLP) processing tasks that involve such long sequences. Of particular interest are topic identification of spoken conversations [4, 5, 6] and call center customer satisfaction prediction [7, 8, 9, 10]. Call center conversations, while usually quite short and to the point,

often involve agents trying to solve very complex issues that the customers experience, resulting in some calls taking even an hour or more. For speech analytics purposes, these calls are typically transcribed using an automatic speech recognition (ASR) system, and processed in textual representations further down the NLP pipeline. These transcripts sometimes exceed the length of 5000 words. Furthermore, temporal information might play an important role in tasks like CSAT. For example, a customer may be angry at the beginning of the call, but after her issue is resolved, she would be very satisfied with the way it was handled. Therefore, simple bag of words models, or any model that does not include temporal dependencies between the inputs, may not be well-suited to handle this category of tasks. This motivates us to employ model such as BERT in this task.

In this paper, we propose a method that builds upon BERT's architecture. We split the input text sequence into shorter segments in order to obtain a representation for each of them using BERT. Then, we use either a recurrent LSTM [11] network, or another Transformer, to perform the actual classification. We call these techniques **Recurrence over BERT (RoBERT)** and **Transformer over BERT (ToBERT)**. Given that these models introduce a hierarchy of representations (segment-wise and document-wise), we refer to them as Hierarchical Transformers. To the best of our knowledge, no attempt has been done before to use the Transformer architecture for classification of such long sequences.

Our novel contributions are:

- Two extensions - RoBERT and ToBERT - to the BERT model, which enable its application in classification of long texts by performing segmentation and using another layer on top of the segment representations.
- State-of-the-art results on the Fisher topic classification task.
- Significant improvement on the CSAT prediction task over the MS-CNN model.

2. RELATED WORK

Several dimensionality reduction algorithms such as RBM, autoencoders, subspace multinomial models (SMM) are used to obtain a low dimensional representation of documents from a simple BOW representation and then classify it using a simple linear classifiers [12, 13, 14, 5]. In [15] hierarchical attention networks are used for document classification. They evaluate their model on several datasets with average number of words around 150. Character-level CNN are explored in [16] but it is prohibitive for very long documents. In [17], dataset collected from arXiv papers is used for classification. For classification, they sample random blocks of words and use them together for classification instead of using full document which may work well as arXiv papers are usually coherent and well written on a well defined topic. Their method may not work well on spoken conversations as random block of words usually do not represent topic of full conversation.

Several researchers addressed the problem of predicting customer satisfaction [7, 8, 9, 10]. In most of these works, logistic regression, SVM, CNN are applied on different kinds of representations.

In [18], authors use BERT for document classification but the average document length is less than BERT maximum length 512. TransformerXL [3] is an extension to the Transformer architecture that allows it to better deal with long inputs for the language modelling task. It relies on the autoregressive property of the model, which is not the case in our tasks.

3. METHOD

3.1. BERT

Because our work builds heavily upon BERT, we provide a brief summary of its features. BERT is built upon the Transformer architecture [1], which uses self-attention, feed-forward layers, residual connections and layer normalization as the main building blocks. It has two pre-training objectives:

- Masked language modelling - some of the words in a sentence are being masked and the model has to predict them based on the context (note the difference from the typical autoregressive language model training objective);
- Next sentence prediction - given two input sequences, decide whether the second one is the next sentence or not.

BERT has been shown to beat the state-of-the-art performance on 11 tasks with no modifications to the model architecture, besides adding a task-specific output layer [2]. We follow same procedure suggested in [2] for our tasks. Fig. 1 shows the BERT model for classification. We obtain two

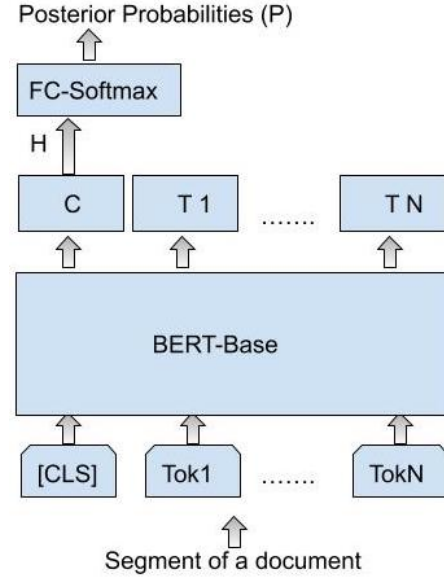


Fig. 1: BERT model for classification. H denotes BERT segment representations from last transformer block, P denotes segment posterior probabilities. Figure inspired from [2]

kinds of representation from BERT: pooled output from last transformer block, denoted by H, and posterior probabilities, denoted by P. There are two variants of BERT - BERT-Base and BERT-Large. In this work we are using BERT-Base for faster training and experimentation, however, our methods are applicable to BERT-Large as well. BERT-Base and BERT-Large are different in model parameters such as number of transformer blocks, number of self-attention heads. Total number of parameters in BERT-Base are 110M and 340M in BERT-Large.

BERT suffers from major limitations in terms of handling long sequences. Firstly, the self-attention layer has a quadratic complexity $O(n^2)$ in terms of the sequence length n [1]. Secondly, BERT uses a learned positional embeddings scheme [2], which means that it won't likely be able to generalize to positions beyond those seen in the training data.

To investigate the effect of fine-tuning BERT on task performance, we use either the pre-trained BERT weights¹, or the weights from a BERT fine-tuned on the task-specific dataset on a segment-level (i.e. we preserve the original label but fine-tune on each segment separately instead of on the whole text sequence). We compare these results to using the fine-tuned segment-level BERT predictions directly as inputs to the next layer.

3.2. Recurrence over BERT

Given that BERT is limited to a particular input length, we split the input sequence into segments of a fixed size with

¹ Available at <https://github.com/huggingface/pytorch-pretrained-BERT>

overlap. For each of these segments, we obtain H or P from BERT model. We then stack these segment-level representations into a sequence, which serves as input to a small (100-dimensional) LSTM layer. Its output serves as a *document embedding*. Finally, we use two fully connected layers with ReLU (30-dimensional) and softmax (the same dimensionality as the number of classes) activations to obtain the final predictions.

With this approach, we overcome BERT’s computational complexity, reducing it to $O(n/k * k^2) = O(nk)$ for RoBERT, with k denoting the segment size (the LSTM component has negligible linear complexity $O(k)$). The positional embeddings are also no longer an issue.

3.3. Transformer over BERT

Given that Transformers’ edge over recurrent networks is their ability to effectively capture long distance relationships between words in a sequence [1], we experiment with replacing the LSTM recurrent layer in favor of a small Transformer model (2 layers of transformer building block containing self-attention, fully connected, etc.). To investigate if preserving the information about the input sequence order is important, we also build a variant of ToBERT which learns positional embeddings at the segment-level representations (but is limited to sequences of length seen during the training).

ToBERT’s computational complexity $O(\frac{n^2}{k^2})$ is asymptotically inferior to RoBERT, as the top-level Transformer model again suffers from quadratic complexity in the number of segments. However, in practice this number is much smaller than the input sequence length ($\frac{n}{k} \ll n$), so we haven’t observed performance or memory issues with our datasets.

4. EXPERIMENTS

We evaluated our models on 3 different datasets:

- CSAT dataset for CSAT prediction, consisting of spoken transcripts (automatic via ASR).
- 20 newsgroups for topic identification task, consisting of written text;
- Fisher Phase 1 corpus for topic identification task, consisting of spoken transcripts (manual);

4.1. CSAT

CSAT dataset consists of US English telephone speech from call centers. For each call in this dataset, customers participated in that call gave a rating on his experience with agent. Originally, this dataset has labels rated on a scale 1-9 with 9 being extremely satisfied and 1 being extremely dissatisfied. Fig. 2 shows the histogram of ratings for our dataset. As the

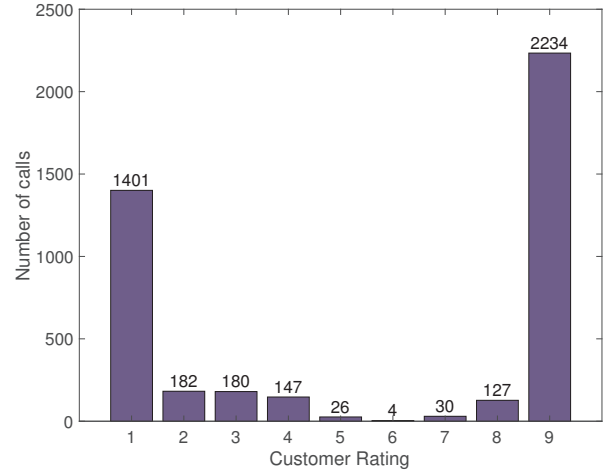


Fig. 2: Histogram of customer ratings. Rating 9 corresponds to extremely satisfied and 1 to extremely dissatisfied

distribution is skewed towards extremes, we choose to do binary classification with ratings above 4.5 as satisfied and below 4.5 as dissatisfied. Quantization of ratings also helped us to create a balanced dataset. This dataset contains 4331 calls and we split them into 3 sets for our experiments: 2866 calls for training, 362 calls for validation and, finally, 1103 calls for testing.

We obtained the transcripts by employing an ASR system. The ASR system uses TDNN-LSTM acoustic model trained on Fisher and Switchboard datasets with lattice-free maximum mutual information criterion [19]. The word error rates using four-gram language models were 9.2% and 17.3% respectively on Switchboard and CallHome portions of Eval2000 dataset².

4.2. 20 newsgroups

20 newsgroups³ data set is one of the frequently used datasets in the text processing community for text classification and text clustering. This data set contains approximately 20,000 English documents from 20 topics to be identified, with 11314 documents for training and 7532 for testing. In this work, we used only 90% of documents for training and the remaining 10% for validation. For fair comparison with other publications, we used 53160 words vocabulary set available in the datasets website.

4.3. Fisher

Fisher Phase 1 US English corpus is often used for automatic speech recognition in speech community. In this work, we used it for topic identification as in [4]. The documents are 10-minute long telephone conversations between two people

²<https://catalog.ldc.upenn.edu/LDC2002T43>

³<http://qwone.com/~jason/20Newsgroups/>

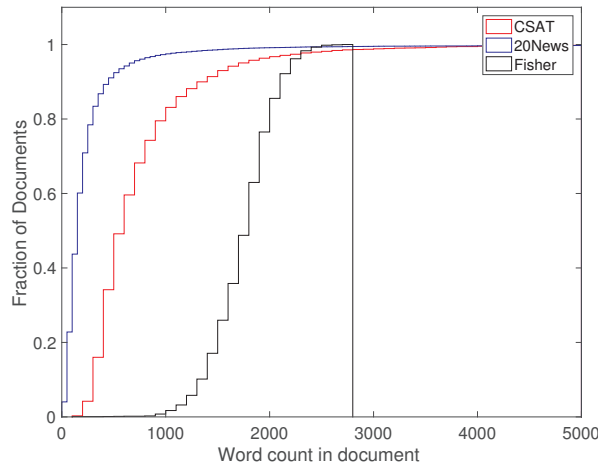


Fig. 3: Cumulative distribution of document lengths.

Dataset	C	N	AW	L
CSAT	2	4331	787	10503
20 newsgroups	20	18846	266	10334
Fisher	40	2746	1788	2713

Table 1: Dataset statistics. C indicates number of Classes, N the Number of documents, AW the Average number of Words per document and L the Longest document length.

discussing a given topic. We used same training and test splits as [4] in which 1374 and 1372 documents are used for training and testing respectively. For validation of our model, we used 10% of training dataset and the remaining 90% was used for actual model training. The number of topics in this data set is 40.

4.4. Dataset Statistics

Table 1 shows statistics of our datasets. It can be observed that average length of Fisher is much higher than 20 newsgroups and CSAT. Cumulative distribution of document lengths for each dataset is shown in Fig. 3. It can be observed that almost all of the documents in Fisher dataset have length more than 1000 words. For CSAT, more than 50% of the documents have length greater than 500 and for 20newsgroups only 10% of the documents have length greater than 500. Note that, for CSAT and 20newsgroups, there are few documents with length more than 5000.

4.5. Architecture and Training Details

In this work, we split document into segments of 200 tokens with a shift of 50 tokens to extract features from BERT model. For RoBERT, LSTM model is trained to minimize cross-entropy loss with *Adam* optimizer [20]. The initial learning rate is set to 0.001 and is reduced by a factor of

	RoBERT	ToBERT
CSAT	71.16	74.77
20newsgroups	60.75	65.04
Fisher	38.04	80.68

Table 2: Results using segment representations (H) from a pre-trained BERT (without fine-tuning).

	RoBERT	ToBERT
CSAT	83.65	83.48
20newsgroups	84.71	85.52
Fisher	82.28	95.48

Table 3: Results using segment representations (H) from a fine-tuned BERT.

0.95 if validation loss does not decrease for 3-epochs. For ToBERT, the Transformer is trained with the default BERT version of *Adam* optimizer [2] with an initial learning rate of $5e-5$. We report accuracy in all of our experiments. We chose a model with the best validation accuracy to calculate accuracy on the test set. To accommodate for non-determinism of some TensorFlow⁴ GPU operations, we report accuracy averaged over 5 runs.

5. RESULTS

Table 2 presents results using pre-trained BERT features. We extracted features from the pooled output of final transformer block as these were shown to be working well for most of the tasks [2]. The features extracted from a pre-trained BERT model without any fine-tuning lead to a sub-par performance. However, We also notice that ToBERT model exploited the pre-trained BERT features better than RoBERT. It also converged faster than RoBERT. Table 3 shows results using features extracted after fine-tuning BERT model with our datasets. Significant improvements can be observed compared to using pre-trained BERT features. Also, it can be noticed that ToBERT outperforms RoBERT on Fisher and 20newsgroups dataset by 13.63% and 0.81% respectively. On CSAT, ToBERT performs slightly worse than RoBERT but it is not statistically significant as this dataset is small.

Table 4 presents results using fine-tuned BERT predictions instead of the pooled output from final transformer block. For each document, having obtained segment-wise predictions we can obtain final prediction for the whole document in three ways:

- Compute the average of all segment-wise predictions and find the most probable class;

⁴We used TensorFlow version 1.14.0.

	Most frequent	Average	RoBERT	ToBERT
CSAT	81.03	82.84	83.54	81.48
20newsgroups	84.78	84.51	84.07	85.47
Fisher	88.70	88.48	91.18	94.16

Table 4: Comparison of models using fine-tuned BERT segment-level predictions (P) instead of segment representations (H).

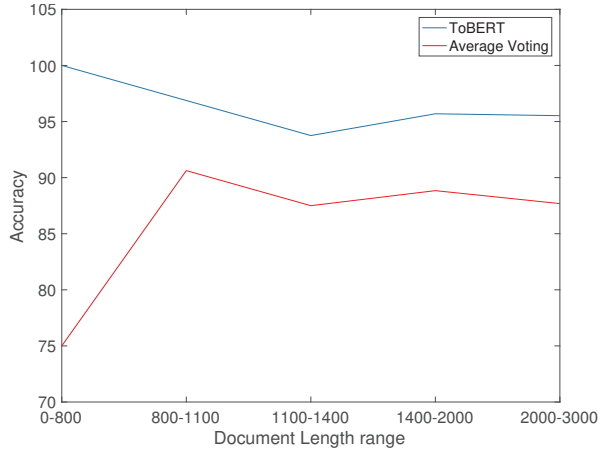


Fig. 4: Comparison of average voting and ToBERT for various document length ranges for Fisher dataset.

- Find the most frequently predicted class;
- Train a classification model.

It can be observed from Table 4 that a simple averaging operation or taking most frequent predicted class works competitively for CSAT and 20newsgroups but not for the Fisher dataset. We believe the improvements from using RoBERT or ToBERT, compared to simple averaging or most frequent operations, are proportional to the fraction of long documents in the dataset. CSAT and 20newsgroups have (on average) significantly shorter documents than Fisher, as seen in Fig. 3. Also, significant improvements for Fisher could be because of less confident predictions from BERT model as this dataset has 40 classes. Fig. 4 presents the comparison of average voting and ToBERT for various document length ranges for Fisher dataset. We used fine-tuned BERT segment-level predictions (P) for this analysis. It can be observed that ToBERT outperforms average voting in every interval. To the best of our knowledge, this is a state-of-the-art result reported on the Fisher dataset.

Table 5 presents the effect of position embeddings on the model performance. It can be observed that position embeddings did not significantly affect the model performance for Fisher and 20newsgroups, but they helped slightly in CSAT prediction (an absolute improvement of 0.64% F1-score). We

	Position embeddings	
	No	Yes
CSAT	82.84	83.48
20newsgroups	85.51	85.52
Fisher	95.84	95.48

Table 5: The effect of including positional embeddings in ToBERT model. Fine-tuned BERT segment representations were used for these results.

dataset	Model	Accuracy
CSAT	MS-CNN	79.53
	ToBERT	83.48
	RoBERT	83.65
20 newsgroups	SCDV [21]	84.6
	MS-CNN [6]	86.12
	ToBERT	85.52
	RoBERT	84.71
Fisher	SVM MCE [4]	91.9
	MS-CNN [6]	92.93
	ToBERT	95.48
	RoBERT	91.18

Table 6: Comparison of our results with previous works.

think that this is explained by the fact that Fisher and 20newsgroups are topic identification tasks, and the topic does not change much throughout these documents. However, CSAT may vary during the call, and in some cases a naive assumption that the sequential nature of the transcripts is irrelevant may lead to wrong conclusions.

Table 6 compares our results with previous works. It can be seen that our model ToBERT outperforms CNN based experiments by significant margin on CSAT and Fisher datasets. For CSAT dataset, we used multi-scale CNN (MS-CNN) as the baseline, given its strong results on Fisher and 20newsgroups. The setup was replicated from [6] for comparison. We also see that our result on 20 newsgroups is 0.6% worse than the state-of-the-art.

6. CONCLUSIONS

In this paper, we presented two methods for long documents using BERT model: RoBERT and ToBERT. We evaluated our experiments on two classification tasks - customer satisfaction prediction and topic identification - using 3 datasets: CSAT, 20newsgroups and Fisher. We observed that ToBERT outperforms RoBERT on pre-trained BERT features and fine-tuned

BERT features for all our tasks. Also, we noticed that fine-tuned BERT performs better than pre-trained BERT. We have shown that both RoBERT and ToBERT improved the simple baselines of taking an average (or the most frequent) of segment-wise predictions for long documents to obtain final prediction. Position embeddings did not significantly affect our models performance, but slightly improved the accuracy on the CSAT task. We obtained the best results on Fisher dataset and good improvements for CSAT task compared to the CNN baseline. It is interesting to note that the longer the average input in a given task, the bigger improvement we observe w.r.t. the baseline for that task. Our results confirm that both RoBERT and ToBERT can be used for long sequences with competitive performance and quick fine-tuning procedure. For future work, we shall focus on training models on long documents directly (i.e. in an end-to-end manner).

7. REFERENCES

- [1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [3] Zihang Dai, Zhilin Yang, Yiming Yang, William W Cohen, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” *arXiv preprint arXiv:1901.02860*, 2019.
- [4] Timothy J Hazen, “Mce training techniques for topic identification of spoken audio documents,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2451–2460, 2011.
- [5] Santosh Kesiraju, Lukás Burget, Igor Szöke, and Jan Cernocký, “Learning document representations using subspace multinomial model,” in *INTERSPEECH*, 2016, pp. 700–704.
- [6] Raghavendra Pappagari, Jesús Villalba, and Najim Dehak, “Joint verification-identification in end-to-end multi-scale cnn framework for topic identification,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6199–6203.
- [7] Shammur Absar Chowdhury, Evgeny A Stepanov, Giuseppe Riccardi, et al., “Predicting user satisfaction from turn-taking in spoken conversations,” in *INTER-SPEECH*, 2016, pp. 2910–2914.
- [8] Jordi Luque, Carlos Segura, Ariadna Sánchez, Martí Umbert, and Luis Angel Galindo, “The role of linguistic and prosodic cues on the prediction of self-reported satisfaction in contact centre phone calls,” *Proc. Inter-speech 2017*, pp. 2346–2350, 2017.
- [9] Youngja Park and Stephen C Gates, “Towards real-time measurement of customer satisfaction using automatically generated call transcripts,” in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 1387–1396.
- [10] Stefan Meinzer, Ulf Jensen, Alexander Thamm, Joachim Hornegger, and Björn M Eskofier, “Can machine learning techniques predict customer dissatisfaction? a feasibility study for the automotive industry,” *Artificial Intelligence Research*, vol. 6, no. 1, pp. 80, 2016.
- [11] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [12] Geoffrey E Hinton and Ruslan R Salakhutdinov, “Replicated softmax: an undirected topic model,” in *Advances in neural information processing systems*, 2009, pp. 1607–1614.
- [13] Hugo Larochelle and Yoshua Bengio, “Classification using discriminative restricted boltzmann machines,” in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 536–543.
- [14] Yu Chen and Mohammed J Zaki, “Kate: K-competitive autoencoder for text,” *arXiv preprint arXiv:1705.02033*, 2017.
- [15] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, “Hierarchical attention networks for document classification,” in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.
- [16] Xiang Zhang, Junbo Zhao, and Yann LeCun, “Character-level convolutional networks for text classification,” in *Advances in neural information processing systems*, 2015, pp. 649–657.
- [17] Liu Liu, Kaile Liu, Zhenghai Cong, Jiali Zhao, Yefei Ji, and Jun He, “Long length document classification by local convolutional feature aggregation,” *Algorithms*, vol. 11, no. 8, pp. 109, 2018.
- [18] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin, “Docbert: Bert for document classification,” *arXiv preprint arXiv:1904.08398*, 2019.

- [19] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur, “Purely sequence-trained neural networks for asr based on lattice-free mmi.,” in *Interspeech*, 2016, pp. 2751–2755.
- [20] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [21] Dheeraj Mekala, Vivek Gupta, Bhargavi Paranjape, and Harish Karnick, “Scdv: Sparse composite document vectors using soft clustering over distributional representations,” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 670–680.