Early Exiting BERT for Efficient Document Ranking

Ji Xin,[1,2] **Rodrigo Nogueira,**[1] **Yaoliang Yu,**[1,2] and **Jimmy Lin**[1,2]

[1] David R. Cheriton School of Computer Science, University of Waterloo
[2] Vector Institute for Artificial Intelligence

Abstract

Pre-trained language models such as BERT have shown their effectiveness in various tasks. Despite their power, they are known to be computationally intensive, which hinders real-world applications. In this paper, we introduce early exiting BERT for document ranking. With a slight modification, BERT becomes a model with multiple output paths, and each inference sample can exit early from these paths. In this way, computation can be effectively allocated among samples, and overall system latency is significantly reduced while the original quality is maintained. Our experiments on two document ranking datasets demonstrate up to 2.5× inference speedup with minimal quality degradation. The source code of our implementation can be found at https://github.com/castorini/earlyexiting-monobert.

1 Introduction

Large scale pre-trained language models such as ELMo (Peters et al., 2018), GPT (Radford et al., 2019), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ALBERT (Lan et al., 2019) have brought impressive improvements to natural language processing (NLP) and information retrieval (IR) applications. However, these large-scale models bring to our community not only exciting results, but also concerns about intensive computation demands and high inference latency, especially in real-world deployments.

In this paper, we study how to accelerate inference of BERT-based IR models. We follow the framework of MonoBERT (Nogueira and Cho, 2019), which performs binary classification on query–document pairs into relevant/non-relevant. To accelerate inference for BERT, we employ the idea of *early exiting* as in DeeBERT (Xin et al., 2020). In DeeBERT, extra classification layers

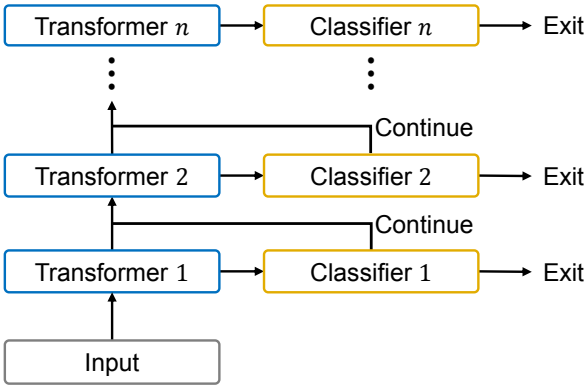

Figure 1: Overview of early exiting BERT for document ranking. Blue blocks are transformer layers and orange blocks are classifiers.

are attached to transformer layers of a pre-trained BERT model (Figure 1). The model is then fine-tuned on the downstream training dataset. At inference time, a sample is sequentially processed by transformer layers and classifiers. If a classifier is confident of its prediction, it returns the result and inference ends early; otherwise, the next transformer layer proceeds with the computation. Different from DeeBERT, which treats all classes equally, we use *asymmetric* early exiting for document ranking: the exiting threshold for positive predictions is higher than for negative ones, since the two classes in document ranking are intrinsically different, and it is natural to allocate more computational resources for positive samples.

We conduct experiments on $\text{BERT}_{\text{BASE}}$ with two document ranking datasets, MS MARCO passage (Bajaj et al., 2016) and ASNQ (Garg et al., 2019). We compare against Cascade Transformer (Soldaini and Moschitti, 2020), a recently proposed technique to accelerate inference in BERT-based document ranking. Results show that our method can reduce inference latency by up to 2.5× with minimal effectiveness degradation.

Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, pages 83–88
Online, November 20, 2020.

2 Related Work

Neural document ranking models have brought significant improvements to IR tasks. Throughout this paper, we refer to candidates to be retrieved as *documents*, although they may actually be passages, answers to a question, etc. Representative neural ranking models include DRMM (Guo et al., 2016), DUET (Mitra et al., 2017), KNRM (Xiong et al., 2017), Co-PACRR (Hui et al., 2018) — just to name a few. In recent years, large-scale pre-trained language models, especially those based on the transformer architecture (Vaswani et al., 2017), have been applied to IR tasks and have pushed the state of the art even further (Nogueira and Cho, 2019; Dai and Callan, 2019; Yilmaz et al., 2019; Li et al., 2020).

The idea of early exiting for neural networks originates from BranchyNet (Teerapittayanon et al., 2017), and is also applied to NLP tasks in several papers (Xin et al., 2020; Schwartz et al., 2020; Liu et al., 2020; Zhou et al., 2020). Our work differs from them by using an early exiting strategy that specializes for document ranking. Another related work that focuses on retrieval is Cascade Transformer (Soldaini and Moschitti, 2020), where a fixed proportion of samples are dropped after each layer. In contrast, our work drops samples based on their scores, and empirically we are able to achieve higher inference speedups.

3 Early Exiting for Document Ranking

The task of concern is *document re-ranking*, i.e., to rank among a small candidate document set, which is generated by a "bag of words" IR technique such as BM25. We assume in this paper that the candidate set is provided as input.

3.1 MonoBERT

We first briefly describe MonoBERT (Nogueira and Cho, 2019), the neural ranking model on which our early exiting model is built.

The input to MonoBERT is a query–document pair, which is organized as one input sequence in the following format:

[CLS] Q [SEP] D [SEP]

Here, Q and D are the query and the document, and [CLS] and [SEP] are special tokens for marking the beginning of input and separating the query and document sequences. Details can be found in the BERT paper (Devlin et al., 2019).

The task of MonoBERT is binary classification: it produces a probability distribution over two classes, relevant and non-relevant.

MonoBERT is initialized with a pre-trained BERT model (or other models with a similar architecture such as RoBERTa). A classifier, which is typically a single-layer fully-connected network, is attached to the last transformer layer of the BERT model; concretely, the classifier takes as input the last layer hidden state corresponding to the [CLS] token, and outputs the binary prediction. For fine-tuning, the model is updated with binary label supervision. For inference, a query–document pair's relevance score is the predicted probability of the document being relevant, and this score is used for subsequent re-ranking of the candidates.

3.2 Fine-Tuning Early Exiting MonoBERT

Our model, early exiting MonoBERT, is a multi-output variant of BERT which enables early exiting. Similar to MonoBERT, we start with a pre-trained $\text{BERT}_{\text{BASE}}$ model with n transformer layers and attach n classifiers to it (Figure 1).

Our fine-tuning method is different from DeeBERT (Xin et al., 2020), where a two-stage fine-tuning method is employed. Instead, we fine-tune the model by simply minimizing the sum of loss functions of all classifiers. The rationale is that with abundant training data, as in our case, this simple fine-tuning method yields comparable or even better results and is also faster. The loss function of the i^{th} classifier is

$$L_i(x, y; \theta) = H(y, f_i(x; \theta)), \tag{1}$$

where x is the input query–document pair, y the binary label of whether the pair is relevant, θ the collection of all parameters, H the cross-entropy loss function, and f_i the binary probability distribution returned by the i^{th} classifier. The network is fine-tuned with the following objective:

$$\min_{\theta} \sum_{(x,y)\in\mathcal{D}} \sum_{i} L_i(x, y; \theta), \tag{2}$$

where $\mathcal{D}$ is the fine-tuning dataset.

3.3 Asymmetric Early Exiting

After the multi-output model is fine-tuned, it is used for inference with early exiting. When an inference sample is fed into the model, it is processed sequentially by each transformer layer and classifier. If the i^{th} layer classifier is confident of

Algorithm 1: Asymmetric Early Exiting

for $i = 1$ to n **do**
  $prob_i = f_i(x;\theta)$
  if $prob_i^{\text{pos}} > \tau_p$ **or** $prob_i^{\text{neg}} > \tau_n$ **then**
    return $prob_i^{\text{pos}}$
  end if
end for
return $prob_n^{\text{pos}}$

the prediction $f_i(x;\theta)$, early exiting is performed and subsequent transformer layers are skipped. We define the *confidence* of $f_i(x;\theta)$ as the higher probability of the two classes. Finally, documents are ranked with respect to their predicted probability of the positive class (relevant).

In previous early exiting BERT for NLP papers, *symmetric* early exiting is used, i.e., if the model's confidence exceeds a threshold, the sample exits. The early exiting algorithm is therefore symmetric with respect to all classes. In the case of early exiting for document ranking, however, there are two fundamental differences from NLP applications. Firstly, the two classes (relevant and non-relevant) are clearly not symmetric: we only care about relevant documents, and the more relevant they are, the more computation resources should be allocated to them. Secondly, for positive samples, confidence is not only the criterion for early exiting, but also the score for subsequent re-ranking.

To bridge the differences mentioned above, we propose to use *asymmetric* early exiting for document ranking. Concretely, we define two thresholds for confidence, τ_p and τ_n, for positive (relevant) and negative (non-relevant) predictions, respectively. We will show by experiments that we should choose a higher positive confidence threshold than the negative one, i.e., if a document is likely to be non-relevant, then we can stop its inference earlier, but if it is expected to be relevant, we should be prudent and use more layers to obtain accurate scores. Details are shown in Algorithm 1.

4 Experimental Setup

We apply early exiting on $\text{BERT}_{\text{BASE}}$ and conduct experiments on two datasets for document ranking, MS MARCO passage (Bajaj et al., 2016) and ASNQ (Garg et al., 2019).

Model and Implementation. We start from a pre-trained $\text{BERT}_{\text{BASE}}$ model. The implementation is adapted from the HuggingFace Transformers Library (Wolf et al., 2019). We fine-tune the model on 4 NVIDIA Tesla V100 GPUs, with a batch size of 60. For other hyperparameters such as learning rate and maximum sequence length, we follow MonoBERT (Nogueira and Cho, 2019) for MS MARCO passage and Cascade Transformer (Soldaini and Moschitti, 2020) for ASNQ.

Dataset Details. MS MARCO passage provides a small version of the training set,[1] from which we build our training set by selecting tuples with *unique* pairs of query–relevant document, yielding a dataset of 832k query–document pairs, half relevant and half non-relevant. We fine-tune the model for 4 epochs. Its development set has 6.9k queries, and for each query there are 1k candidate documents, among which there is approximately 1 relevant document.

ASNQ's training set[2] has 20M query–document pairs. Similar to MS MARCO passage, we select only unique pairs of query–relevant document, and then complement the dataset with the same amount of query–non-relevant document pairs to yield a training set of 114k pairs. We fine-tune for 2 epochs. Its development set has 1.3k queries, and each query has, on average, 400 candidate documents and 3 relevant ones.

5 Experimental Results

We show the trade-offs between model quality and computation of early exiting in Tables 1 and 2 for MS MARCO passage and ANSQ, respectively. Different trade-offs are achieved by setting various negative confidence threshold τ_n, while τ_p is always set to 1; we will provide detailed analyses of these thresholds later. Inference efficiency is quantified by the average exit layer of inference samples; this metric is, according to our experiments, proportional to actual wall-clock runtime, while being invariant across multiple runs.

We can see that in both datasets, early exiting is able to accelerate inference by $\sim$2.5$\times$ while maintaining the original model effectiveness. It is worth noting that in Cascade Transformer (CT) (Soldaini and Moschitti, 2020), only a part of the development set is used for evaluation, and therefore the scores are not directly comparable. However, in

[1] https://msmarco.blob.core.windows.net/msmarcoranking/triples.train.small.tar.gz

[2] https://github.com/alexa/wqa_tanda

Method	MRR @10	τ_n	Speedup
MB	0.347		1.0×
eeMB	0.343	1.00	1.0×
	0.343 (−0%)	0.95	2.6×
	0.340 (−1%)	0.90	2.9×
	0.336 (−2%)	0.85	3.2×
	0.327 (−5%)	0.80	3.5×
	0.312 (−9%)	0.75	3.9×
	0.290 (−15%)	0.70	4.3×

Table 1: MS MARCO passage development set results. MB: MonoBERT; eeMB: early exiting MonoBERT.

Method	nDCG @10	MRR	τ_n	Speedup
CT	0.661	0.654		1.0×
	0.653	0.653		1.6×
	0.650	0.648		1.8×
	0.650	0.645		2.0×
eeMB	0.650	0.633	1.00	1.0×
	0.650	0.633	0.99	2.5×
	0.648	0.632	0.95	3.2×
	0.646	0.632	0.90	3.6×
	0.638	0.627	0.80	4.1×

Table 2: ASNQ development set results. CT: Cascade Transformer; eeMB: early exiting MonoBERT. Absolute values of scores of CT and eeMB are not directly comparable due to dataset differences.

terms of *relative* performance, our model appears to achieve a bit higher inference speedup with a comparable score degradation.

We also compare confidence-based early exiting trade-offs with *layer-wise scores* of the model in Figures 2 and 3. The layer-wise score of the i^{th} layer is obtained by forcing all inference samples to exit through the classifier at the i^{th} layer. It provide a series of baselines: if we want to save 50% inference computation for a 12-layer model, a straightforward way is to use the 6$^{\text{th}}$ layer's classifier for all samples. The first two layers are omitted from the layer-wise score curves since their scores are too low to be useful. The figures show that the early exiting idea significantly outperforms the naïve baselines.

To analyze the effect of different confidence thresholds, we plot in Figure 4 the comparison of confidence thresholds τ_p and τ_n using the MS

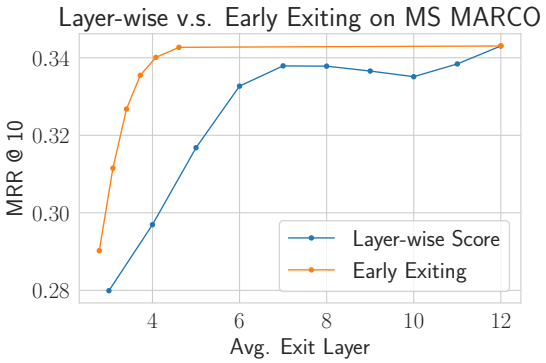

Figure 2: Comparison between layer-wise scores and early exiting trade-offs, on the MS MARCO passage development set.

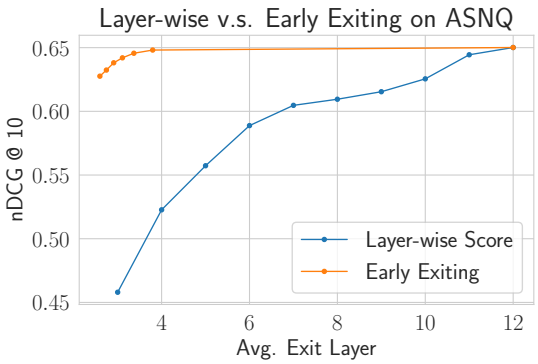

Figure 3: Comparison between layer-wise scores and early exiting trade-offs, on the ASNQ development set.

MARCO passage dataset. Results from ASNQ are very similar and therefore omitted. Each curve corresponds to one τ_p value, and points within a curve are plotted by choosing different τ_n values.[3]

We notice that the trade-off performance is monotonic with respect to the positive confidence threshold: the higher τ_p is, the better the trade-offs are. We speculate the reason is that while smaller τ_p improves efficiency by allowing more samples to exit earlier, the quality of predictions from earlier layers of relevant samples degrades drastically. Considering the fact that relevant samples constitute only a tiny fraction of all candidates in both datasets, setting $\tau_p = 1$, i.e., using as many transformer layers as we can on positive samples, is the optimal choice. It is worth noting that for other datasets with higher relevant candidate proportions, the optimal τ_p may be smaller than 1.

Within one curve (a fixed τ_p), τ_n controls the

[3]Points on each curve, from left to right, correspond to τ_n values in Table 1, from bottom to top.

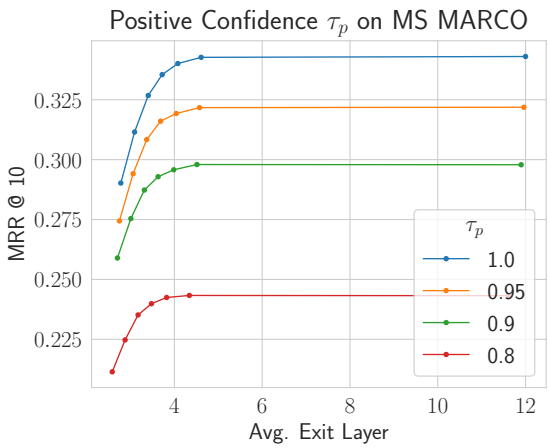

Figure 4: Comparison between different values of τ_p (curves) and τ_n (points on a curve).

trade-offs between efficiency and quality: lower τ_n allows more negative samples to exit earlier, thus improving efficiency with relatively small quality degradation. Such asymmetry demonstrates the necessity of using two confidence thresholds instead of one: while we can safely perform early exiting on negative samples and save computation, we should allocate far more resources to positive samples for the most accurate predictions.

6 Conclusion

We propose asymmetric early exiting BERT for document ranking, an effective method to improve model efficiency in IR tasks. Computation resources are allocated to samples according to their needs. Experiments show that our method is able to achieve different quality–latency trade-offs by setting different thresholds, and also improves model efficiency over baselines.

Acknowledgments

This research was supported in part by the Canada First Research Excellence Fund and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In *Proceedings of SIGIR*, pages 985–988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*, pages 4171–4186.

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2019. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. *arXiv preprint arXiv:1911.04118*.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of CIKM*, pages 55–64.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard De Melo. 2018. Co-PACRR: A context-aware neural IR model for ad-hoc retrieval. In *Proceedings of WSDM*, pages 279–287.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage representation aggregation for document reranking. *arXiv preprint arXiv:2008.09093*.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of ACL*, pages 6035–6044.

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 arXiv:1907.11692*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of WWW*, pages 1291–1299.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*, pages 2227–2237.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020. The right tool for the job: Matching model and instance complexities. In *Proceedings of ACL*, pages 6640–6651.

Luca Soldaini and Alessandro Moschitti. 2020. The Cascade Transformer: An application for efficient answer sentence selection. In *Proceedings of ACL*, pages 5697–5708.

Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2017. BranchyNet: Fast inference via early exiting from deep neural networks. *arXiv preprint arXiv:1709.01686*.

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 *Proceedings of NeurIPS*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of ACL*, pages 2246–2251.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of SIGIR*, pages 55–64.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. In *Proceedings of EMNLP*, pages 3481–3487.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. BERT loses patience: Fast and robust inference with early exit. *arXiv preprint arXiv:2006.04152*.