An Empirical Study of Uniform-Architecture Knowledge Distillation in Document Ranking

Xubo Qin
JD.com
China
qratosone@live.com

Xiyuan Liu
University of Colorado, Boulder
United States
heathclief.liu.physics@gmail.com

Xiongfeng Zheng
Platform and Content Group, Tencent
China
peacezheng@tencent.com

Jie Liu
Platform and Content Group, Tencent
China
jesangliu@tencent.com

Yutao Zhu
yutao.zhu@umontreal.ca
DIRO, Université de Montréal
Canada

ABSTRACT

Although BERT-based ranking models have been commonly used in commercial search engines, they are usually time-consuming for online ranking tasks. Knowledge distillation, which aims at learning a smaller model with comparable performance to a larger model, is a common strategy for reducing the online inference latency. In this paper, we investigate the effect of different loss functions for uniform-architecture distillation of BERT-based ranking models. Here "uniform-architecture" denotes that both teacher and student models are in cross-encoder architecture, while the student models include small-scaled pre-trained language models. Our experimental results reveal that the optimal distillation configuration for ranking tasks is much different than general natural language processing tasks. Specifically, when the student models are in cross-encoder architecture, a pairwise loss of hard labels is critical for training student models, whereas the distillation objectives of intermediate Transformer layers may hurt performance. These findings emphasize the necessity of carefully designing a distillation strategy (for cross-encoder student models) tailored for document ranking with pairwise training samples.

1 INTRODUCTION

Recent years have witnessed great progress of applying deep learning methods to information retrieval tasks [19]. In particular, on document ranking, pre-trained language models (PLM), such as BERT [4], have achieved state-of-the art performance. However, because these pre-trained models often have a large number of parameters, they incur an inevitable computational cost and latency during the inference stage [6]. This problem will be even severe when deploying pre-trained models in latency-sensitive online ranking tasks. To tackle this problem, numerous PLM-based knowledge distillation (KD) methods [14] have been widely studied. The principle of knowledge distillation can be summarized as follows: first, learn a *teacher* model using the labels in training data, and then learn a *student* model using both the training data and the teacher model. Specifically, when training student models, the ground-truth labels from the data are used as *hard* labels, while the output logits of the teacher model are used as *soft* labels. In comparison to directly optimizing the student model with only hard labels, adding soft labels can assist the student model in learning to simulate the behavior of the teacher model, hence improving performance. It is critical to select an appropriate loss function (also known as *distillation objective* [7]) for learning soft labels.

Most of existing approaches [7, 15–17, 22] for language model distillation aim at improving the performance of student model in general natural language understanding (NLU) tasks [21]. Specially for document retrieval and ranking tasks, some latest approaches are focusing on cross-architecture distillation approaches [5, 10] using cross-encoder [12] teachers and ColBERT [8] or dual-encoder [2] based students. For those cross-architecture approaches, both teachers and students are using pre-trained models in the same scale (*i.e.* ERNIE2.0 [18] with 12-layer or 24-layer Transformer [20] Encoders). Since the amount of training samples for document retrieval and ranking tasks is usually very large, it may cost thousands of GPU hours to train the large-scaled teachers and students, which is usually not affordable for some commercial search engines. As a result, it's still necessary to explore the uniform-architecture distillation approaches with small-scaled cross-encoder students.

In this paper, we conduct an empirical study to investigate the effectiveness of different distillation objectives in document ranking tasks. We focus on analyzing the distillation effect **given a cross-encoder student model with a small-scaled PLM of 4 Transformer layers**. For uniform-architecture distillation in ranking tasks, one of the latest approaches is Simplified TinyBERT [3] (henceforth denoted as STinyBERT) which proposes some simplifications on TinyBERT [7] in general NLU tasks. A major problem is that the knowledge distillation objectives in both TinyBERT and STinyBERT are based on pointwise training samples, which fit to classification problems. However, in document ranking, pairwise training is widely used and has demonstrated to be more effective [1, 9]. Therefore, our first research problem in this paper is how to combine pairwise ranking with knowledge distillation objectives. We propose a straightforward idea of integrating pairwise ranking into STinyBERT and design a pairwise distillation objective. Then, after analyzing the mechanism of pairwise ranking, we consider

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

that in the original STinyBERT, the distillation objectives based on intermediate Transformer layers are superfluous. Our further empirical study confirms our speculation. Based on the experimental results, we simplify the knowledge distillation objectives in STinyBERT, and finally propose a **PD-BERT** (PD stands for Pairwise Distillation) strategy for knowledge distillation document ranking.

We conduct experiments on the MS MARCO dataset. The experimental results show that our proposed PD-BERT with fewer distillation objectives outperforms existing methods. Our results clearly indicate that to achieve best performance of ranking models, some novel distillation objectives based on pairwise training samples should be carefully designed, and these objectives should be able to represent the relative relations of the pairwise samples.

2 RELATED WORKS

There are multiple architectures for BERT-based ranking models, including cross-encoder [12], dual encoders [2] and ColBERT [8]. Comparing with other architectures, cross-encoder can achieve the best performance, while its inference latency is the highest comparing with the other ones. This issue is critical for commercial search engines. To address this issue, researchers have proposed various knowledge distillation approaches aiming to train a distilled BERT model to boost up inference without losing much performance, and cross-encoder models are usually used as teacher models.

Typical distillation approaches [7, 15–17, 22] for PLMs are focusing on general natural language understanding (NLU) tasks [21]. For document retrieval and ranking tasks, the corresponding approaches can be divided into **cross-architecture** and **uniform-architecture** approaches. Given a cross-encoder teacher model with large-scaled PLM, the cross-architecture approaches such as Margin-MSE [5] or ERNIE-Search [10] are focusing on distilling a student model in ColBERT or dual encoders architectures with a PLM as large as the teacher (such as 12 Transformer layers). While for uniform-architecture approaches, the student model is a cross-encoder with a PLM in smaller scale. Table 1 shows the differences between cross and uniform architecture.

For ranking tasks, existing state-of-the-art approach of uniform-architecture distillation is Simplified TinyBERT (STinyBERT) [3] which is an extension of TinyBERT [7] for general NLU tasks. Inheriting the spirit of TinyBERT, STinyBERT employs three objectives to learn the weights of some intermediate Transformer layers, including the weights of embedding layers, hidden states, and attention matrices. And comparing with original TinyBERT, STinyBERT merges the two steps for task-specific distillation into one step, and adds a pointwise loss based on ground-truth labels. However, as the training data of ranking task are usually organized as pairwise training samples [9], the effect of existing distillation objectives with pairwise training samples is still unclear.

Different from existing studies, we propose our PD-BERT strategy to adapt the pointwise-based distillation approaches for pairwise samples. We further conduct an empirical study, based on which we propose a simplification on the distillation objectives and obtain better results.

Table 1: Comparison between Cross & Uniform Architecture approaches. "PLM Scale" denotes the scale of student models' PLMs. "Arc." and "Enc." are the short of "Architecture" and "Encoder".

Type	Student Arc.	PLM Scale
Uniform-Arc.	Cross-Enc. (Same as teacher)	Smaller than teacher
Cross-Arc.	ColBERT or Dual-Enc.	Same as teacher

3 PROPOSED METHOD

3.1 Background

Before diving into the detail of our method, we first briefly introduce some background knowledge about pointwise/pairwise ranking and the commonly used distillation objectives in STinyBERT.

3.1.1 Pointwise vs. Pairwise Ranking. Pointwise and pairwise approaches are two commonly used methods in learning-to-rank framework. **Pointwise methods** consider a single document at a time in the loss function. They usually train a classifier (with a cross-entropy loss) to measure whether a document is relevant to a query. Then, the final ranking is achieved by sorting the documents according to their predicted scores. In pointwise methods, the score for each document is independent of those for other documents. On the contrary, **pairwise methods** consider a pair of documents in loss functions. Their target is to learn the relative order of two documents under a same query. In practice, the pairwise methods usually perform better than pointwise methods because predicting relative order is closer to the nature of ranking [1, 9]. Specifically, for a query q, assuming there is a relevant document d^+ and an irrelevant one d^-, the pairwise ranking loss can be defined as:

$$\mathcal{L}_{\text{pair}} = \max(0, 1 - f(q, d^+) + f(q, d^-)), \tag{1}$$

where $f(q, d)$ is the ranking score between q and d. Similar to pointwise methods, the final ranking list can be obtained by sorting the documents according to their ranking scores.

3.1.2 Knowledge Distillation in STinyBERT. Simplified TinyBERT (`STinyBERT`) is designed for document retrieval, which applies some simplifications over `TinyBERT` [7]. It is distilled from BERT with the following five objectives:

$$\mathcal{L}_1 = \mathcal{L}_{\text{attn}} + \mathcal{L}_{\text{hidn}} + \mathcal{L}_{\text{emb}} + \mathcal{L}_{\text{hard}} + \mathcal{L}_{\text{logits}}. \tag{2}$$

The first three objectives are designed for learning the attention weights, intermediate hidden layers, and embedding layers from the teacher model, which can be defined as: $\mathcal{L}_i = \text{MSE}(L_i^T, L_i^S), i \in \{\text{attn}, \text{hidn}, \text{emb}\}$. L_i^T and L_i^S are attention/hidden/embedding matrices of the teacher and student model, respectively. By optimizing these objectives, the student model can make their parameter metrics closer to the teacher model, thus leading to comparable performance. The last two objectives are defined in standard knowledge distillation. $\mathcal{L}_{\text{hard}}$ is a cross-entropy loss computed on the student model's output logits $\mathbf{z}^S$ and the label, while $\mathcal{L}_{\text{logits}}$ is a soft cross-entropy loss measuring the discrepancy between $\mathbf{z}^S$ and the logits of teacher $\mathbf{z}^T$. To adapt for distillation, $\mathbf{z} \in \mathbb{R}^2$ as the ranking task is formulated as a binary classification task. The probability of being relevant (*i.e.*, $\mathbf{z}^S[0]$) is used as the ranking score $f(q, d)$.

More details can be obtained in the original paper of TinyBERT [7] and STinyBERT [3].

3.2 Pairwise Distillation Objective

In original STinyBERT, $\mathcal{L}_{\text{hard}}$ is defined as a cross-entropy loss, which can be treated as a pointwise method. As introduced in Section 3.1.1, researchers have demonstrated that pairwise approaches can perform better than pointwise approaches in practice. Therefore, our first research question is: *how can we combine pairwise ranking with knowledge distillation objectives?*

A straightforward idea is replacing $\mathcal{L}_{\text{hard}}$ in Equation 2 with $\mathcal{L}_{\text{pair}}$ defined in Equation 1. However, different from pointwise ranking, pairwise ranking involves two documents (*i.e.*, d^+ and d^-). To consider such a document pair in knowledge distillation, we propose a pairwise distillation objective as:

$$\mathcal{L}_2 = \mathcal{L}_{\text{pair}} + \mathcal{L}^+ + \mathcal{L}^-, \tag{3}$$

$$\mathcal{L}^+ = \mathcal{L}^+_{\text{attn}} + \mathcal{L}^+_{\text{hidn}} + \mathcal{L}^+_{\text{emb}} + \mathcal{L}^+_{\text{logits}}, \tag{4}$$

$$\mathcal{L}^- = \mathcal{L}^-_{\text{attn}} + \mathcal{L}^-_{\text{hidn}} + \mathcal{L}^-_{\text{emb}} + \mathcal{L}^-_{\text{logits}}, \tag{5}$$

where $\mathcal{L}^+$ and $\mathcal{L}^-$ are computed on the positive pair (q, d^+) and the negative pair (q, d^-), respectively. STinyBERT distilled with $\mathcal{L}_2$ is denoted as "STinyBERT+Pairwise".

In Equation 3, we can see there are nine objectives should be computed in total, which is very complex. So, our next research question is: *are all these objectives necessary?*

Original STinyBERT is designed for pointwise ranking. The distillation objectives for different layers or attentions can help the student model obtain parameters similar to the teacher model, so that the student model can output a similar ranking score. Intuitively, this constraint may not be necessary for our pairwise distillation objective, because our model is trained to learn a relative order between two documents rather than compute a specific ranking score for a single document. To validate our assumption, we conduct an empirical study and find that the distillation objectives for intermediate parameters (*i.e.*, $\mathcal{L}^+_i$ and $\mathcal{L}^+_i, i \in \{\text{attn}, \text{emb}, \text{hidn}\}$) even hurt the model's performance (experimental results are given in Section 4.3.1). As a result, we propose a simplification on our pairwise distillation objective as:

$$\mathcal{L}_3 = \mathcal{L}_{\text{pair}} + \mathcal{L}^+_{\text{logits}} + \mathcal{L}^-_{\text{logits}}. \tag{6}$$

STinyBERT distilled with our proposed $\mathcal{L}_3$ is denoted as "PD-BERT".

4 EXPERIMENTS

4.1 Datasets and Implementation Details

Following previous work [3], we conduct experiments on a subset of the MSMARCO [11] passage ranking dataset, where only the first 3.2M samples are used for training. The queries and passages are truncated into the length of 32 and 120. For evaluation, we use the whole development set provided by MSMARCO official. It includes about 7k queries and each query contains 1k passages for re-ranking. We use MRR@10 as the evaluation metric, which is suggested by the official evaluation.

Table 2: Performance of different approaches.

Model	Type	MRR@10
BERT-base (Teacher)	Fine-tuned	**0.359**
TinyBERT-L4	Fine-tuned	0.324
STinyBERT	Distilled	0.330
STinyBERT+Pairwise	Distilled	0.334
Margin-MSE	Distilled	0.335
PD-BERT	Distilled	**0.340**

We use PyTorch [13] and HuggingFace's Transformers [23] to implement the models. For the teacher model, we fine-tune a pre-trained BERT-base [4] model for one epoch with a learning rate of 3e-6 and batch size of 32. This model has 12 Transformer layers and the hidden sizes of all layers are 768. For all student models, we use the general-distilled TinyBERT checkpoint to initialize the model.[1] This model is denoted as TinyBERT-L4, which consists of four Transformer layers with the hidden size of 312. The temperature for $\mathcal{L}_{\text{logits}}$ is fixed as 1. Other hyper-parameters and distillation settings are the same as those in original STinyBERT [3].

4.2 Baseline

We compare our proposed PD-BERT with three groups of methods:

(1) **Fine-tune methods.** We fine-tune BERT-base and TinyBERT-L4 on the dataset and report their performance. This BERT-base model is used as the teacher model in the following distillation approaches.

(2) **Distillation methods.** To validate the effectiveness of our proposed pairwise distillation objectives, we report the performance of original STinyBERT and STinyBERT+Pairwise. Both of these two approaches includes the transformer-based distillation objectives $\mathcal{L}_{\text{attn}}$, $\mathcal{L}_{\text{hidn}}$, $\mathcal{L}_{\text{emb}}$. We further report the performance of PD-BERT, which all those transformer-based objectives are ignored as Equation 6 describes.

(3) **Variants of STinyBERT.** To investigate the influence of different knowledge distillation objectives, we remove some objectives from Equation 3 and test the performance of these variants.

4.3 Results and Discussion

4.3.1 Overall Results. Table 2 shows the experimental results of different fine-tune methods and knowledge distillation methods. We can see that our proposed PD-BERT outperforms all other baselines, which clearly demonstrates its superiority. Furthermore, we have the following observations:

(1) All the distillation approaches can outperform TinyBERT-L4 with fine-tuning, minimizing the gap between the student model and the teacher model (BERT-base). Besides, our PD-BERT has achieved the best performance among all those approaches.

(2) Compared with STinyBERT, STinyBERT+Pairwise achieves 0.004 absolute improvement in terms of MRR@10. The difference of these two models is the ranking loss. So, the improvement validates the effectiveness of our proposed pairwise distillation objective.

[1] https://huggingface.co/huawei-noah/TinyBERT_General_4L_312D. Notice that TinyBERT provides checkpoints for both general and task-specific distillation, we use the general one to initialize the student model.

Table 3: Results of ablation study. $\mathcal{L}_{\text{pair}}$ is used in all variants, thus being omitted. All distillation objectives are computed on both positive and negative samples, so the mark +/− is also omitted.

Model	Distillation Objectives	MRR@10
STinyBERT+Pairwise	$\mathcal{L}_{\text{logits}}$, $\mathcal{L}_{\text{emb}}$, $\mathcal{L}_{\text{hidn}}$, $\mathcal{L}_{\text{attn}}$	0.334
↪ *w/o* Intermediate	$\mathcal{L}_{\text{logits}}$, $\mathcal{L}_{\text{emb}}$	0.339
↪ *w/o* Embedding	$\mathcal{L}_{\text{logits}}$, $\mathcal{L}_{\text{attn}}$, $\mathcal{L}_{\text{hidn}}$	0.334
↪ *w/o* Logits	$\mathcal{L}_{\text{attn}}$, $\mathcal{L}_{\text{hidn}}$, $\mathcal{L}_{\text{emb}}$	0.319
PD-BERT	$\mathcal{L}_{\text{logits}}$	**0.340**

Table 4: Performance with different distillation settings. "Teacher Layer" indicates which Transformer layer of the teacher model is used for distillation.

Model	Teacher Layer	MRR@10
STinyBERT+Pairwise	3,6,9,12	0.334
↪ Last 4 layers	9,10,11,12	0.333
↪ Last 1 layer	12	0.339
PD-BERT	-	**0.340**

This result is consistent with the findings of previous work that pairwise ranking usually performs better than pointwise ranking [9].

(3) Intriguingly, compared with STinyBERT+Pairwise, our PD-BERT is trained with less distillation objectives but achieves better performance, indicating that the Transformer-based distillation objectives do harm to the performance when pairwise training samples are used. This confirms our assumption that our pairwise distillation method can loose the constraint on learning intermediate parameters. With more flexible parameter tuning, the student model can adapt better to downstream tasks when pairwise training samples are given.

4.3.2 Ablation Studies. Previous experimental results imply that some knowledge distillation objectives are not suitable for document ranking tasks, especially for pairwise ranking. Therefore, we conduct an ablation study to investigate the influence of different distillation objectives in STinyBERT+Pairwise. The experimental results are shown in Table 3. We can see:

First, removing any distillation objectives for intermediate parameters leads to performance improvement. This confirms again our assumption that simulating parameters in intermediate Transformer layers is unnecessary when training with pairwise distillation objective. Second, compared with $\mathcal{L}_{\text{emb}}$, adding $\mathcal{L}_{\text{attn}}$ and $\mathcal{L}_{\text{hidn}}$ has negative effect on model's performance. Since the number of parameters in intermediate layers is much more than that in embedding layer, we speculate that learning too much parameters may disturb the pairwise distillation. Finally, when removing $\mathcal{L}_{\text{logits}}$, the model's performance drops significantly. This demonstrates the importance of learning output logits from the teacher model.

4.3.3 Effect of Different Distillation Settings. Previous studies [24] have analyzed the effect of different layers of BERT in document ranking and reported that BERT relies heavily on the interactions of the last several layers for predicting the relevance between query and document. Similarly, in knowledge distillation, the selection of distillation layers in BERT may also influence the performance. Following previous work [7], we test several different distillation settings over BERT layers (they are also known as mapping functions). The results are shown in Table 4, and we have the following findings:

First, distilling with the last four layers cannot bring further improvement. The explanation for this could be that, while the last four layers are considered to be crucial for capturing matching signals between query and document, they relied on the bottom layers as well. Due to the limited number of layers of STinyBERT, they are unable to extract as much information as vanilla BERT. Second, when only distilling the last layer of BERT, STinyBERT+Pairwise perform slightly worse than our PD-BERT. This reflects that the distillation on intermediate layer of BERT is useless. Therefore, our simplification on distillation objectives is straightforward yet effective.

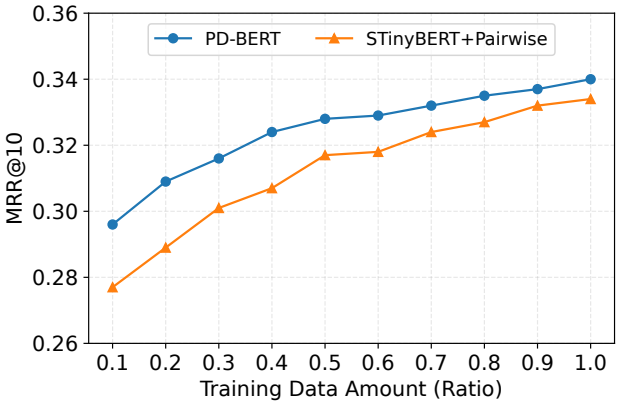

Figure 1: Performance comparison with different amounts of training data.

4.3.4 Influence of Training Data Amount. Knowledge distillation is usually influenced by the number of training data, to investigate such impact, we test the performance of PD-BERT and PTinyBERT+Pairwise trained with only a proportion of data (from 0.1 to 1.0). The result is shown in Figure 1.

It is evident to see that, the performance of both PD-BERT and STinyBERT+Pairwise is increasing with more training data used. This is natural as knowledge distillation benefits from sufficient data. Moreover, we can also observe that the performance difference between STinyBERT+Pairwise and PD-BERT becomes less when providing more training data. A possible explanation is, in pairwise training many candidate documents are repeatedly used in multiple document pairs. In Section 3.2, we described that the distillation objectives cannot measure the relative orders between those document pairs. As a result, most of those documents in training set will be redundant for the distillation objectives, which may lead to over-fitting. Compared with the distillation objective on output logits, those aiming at learning intermediate parameters are easier to over-fit and require more unique training documents for tuning.

5 CONCLUSION

In this paper we investigate the effect of those widely-used objectives of uniform-architecture distillation in ranking tasks. For the student models in cross-encoder architecture, we demonstrate that existing Transformer-based objectives in TinyBERT are not optimal with pairwise training samples. The distillation objectives of intermediate Transformer layers will even do harm to the student model's performance. This may be because the existing distillation objectives are based on pointwise training samples, and the Transformer-based objectives are easier to overfit since there are not enough unique pointwise samples in pairwise training. To fit the ranking task, some kinds of novel distillation objectives should be carefully designed to measure the relative orders in pairwise samples.

REFERENCES

[1] Christopher JC Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. *Learning* 11, 23-581 (2010), 81.

[2] Yinqiong Cai, Yixing Fan, Jiafeng Guo, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2021. Semantic Models for the First-stage Retrieval: A Comprehensive Review. *CoRR* abs/2103.04831 (2021). arXiv:2103.04831 https://arxiv.org/abs/2103.04831

[3] Xuanang Chen, Ben He, Kai Hui, Le Sun, and Yingfei Sun. 2021. Simplified TinyBERT: Knowledge Distillation for Document Retrieval. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12657)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 241–248. https://doi.org/10.1007/978-3-030-72240-1_21

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[5] Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. 2020. Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation. arXiv:2010.02666 [cs.IR]

[6] Sebastian Hofstätter and Allan Hanbury. 2019. Let's measure run time! Extending the IR replicability infrastructure to include performance aspects. arXiv:1907.04614 [cs.IR]

[7] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4163–4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

[8] Omar Khattab and Matei Zaharia. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3397271.3401075

[9] Tie-Yan Liu. 2010. Learning to rank for information retrieval. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy (Eds.). ACM, 904. https://doi.org/10.1145/1835449.1835676

[10] Yuxiang Lu, Yiding Liu, Jiaxiang Liu, Yunsheng Shi, Zhengjie Huang, Shikun Feng Yu Sun, Hao Tian, Hua Wu, Shuaiqiang Wang, Dawei Yin, and Haifeng Wang. 2022. ERNIE-Search: Bridging Cross-Encoder with Dual-Encoder via Self On-the-fly Distillation for Dense Passage Retrieval. https://doi.org/10.48550/ARXIV.2205.09153

[11] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*, Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne (Eds.). CEUR-WS.org. http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[12] Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. *CoRR* abs/1910.14424 (2019). arXiv:1910.14424 http://arxiv.org/abs/1910.14424

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 8024–8035. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[14] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. arXiv:1412.6550 [cs.LG]

[15] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108 (2019). arXiv:1910.01108 http://arxiv.org/abs/1910.01108

[16] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient Knowledge Distillation for BERT Model Compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 4322–4331. https://doi.org/10.18653/v1/D19-1441

[17] Siqi Sun, Zhe Gan, Yu Cheng, Yuwei Fang, Shuohang Wang, and Jingjing Liu. 2020. Contrastive Distillation on Intermediate Representations for Language Model Compression. *CoRR* abs/2009.14167 (2020). arXiv:2009.14167 https://arxiv.org/abs/2009.14167

[18] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019. ERNIE 2.0: A Continual Pre-training Framework for Language Understanding. https://doi.org/10.48550/ARXIV.1907.12412

[19] Dacheng Tao. 2020. *How Deep Learning Works for Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 5. https://doi.org/10.1145/3397271.3402429

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[21] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *CoRR* abs/1804.07461 (2018). arXiv:1804.07461 http://arxiv.org/abs/1804.07461

[22] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. arXiv:2002.10957 [cs.CL]

[23] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR* abs/1910.03771 (2019). arXiv:1910.03771 http://arxiv.org/abs/1910.03771

[24] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An Analysis of BERT in Document Ranking. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1941–1944. https://doi.org/10.1145/3397271.3401325