PILE: Pairwise Iterative Logits Ensemble for Multi-Teacher Labeled Distillation

Lianshang Cai[*] **Linhao Zhang**[*] **Dehong Ma**[†] **Jun Fan**
Daiting Shi, Yi Wu, Zhicong Cheng, Simiu Gu, Dawei Yin[†]
Baidu Inc., Beijing, China
{cailianshang,zhanglinhao,madehong,fanjun}@baidu.com
{shidaiting01,wuyi01,chengzhicong01,gusimiu}@baidu.com
yindawei@acm.org

Abstract

Pre-trained language models have become a crucial part of ranking systems and achieved very impressive effects recently. To maintain high performance while keeping efficient computations, knowledge distillation is widely used. In this paper, we focus on two key questions in knowledge distillation for ranking models: 1) how to ensemble knowledge from multi-teacher; 2) how to utilize the label information of data in the distillation process. We propose a unified algorithm called Pairwise Iterative Logits Ensemble (PILE) to tackle these two questions simultaneously. PILE ensembles multi-teacher logits supervised by label information in an iterative way and achieved competitive performance in both offline and online experiments. The proposed method has been deployed in a real-world commercial search system.

1 Introduction

Search engines have been an infrastructure in the Information Age to satisfy people's needs for querying about information. In modern search engines, multi-stage pipelines are usually employed where *ranking* is usually known as the very final stage. It takes as input the shortlisted candidates of relevant documents (i.e. web pages) retrieved from previous stages, and concentrates on sorting (Lin et al., 2021) based on the degree of match between the latent semantics of documents and the search intent of the user's query.

With the flourishing of pre-trained language models (Devlin et al., 2018; Sun et al., 2019b; Lan et al., 2019; He et al., 2020), BERT-based models have achieved state-of-the-art performance in a broad range of downstream tasks and text ranking is no exception. Pre-trained rankers based on BERT show impressive performance in ranking tasks (Nogueira and Cho, 2019; Nogueira et al., 2019; Zou et al., 2021).

Despite the state-of-the-art performance pre-trained models yield in laboratories, it is hardly possible to apply them directly in real-world search engines. Their large numbers of parameters go against computational efficiency while the online environment is strictly restricted in resources. Therefore, before deploying a pre-trained model online, one necessary procedure is to reduce computational costs.

Knowledge distillation is one of the most commonly used methods to reduce the model size (Cristian et al., 2006; Hinton et al., 2015a) and accelerate the computation process. In a standard workflow of distillation, a large model (i.e. teacher) is pre-trained and finetuned well in advance, and a small model (i.e. student) imitates the teacher model's behaviors. The knowledge learned by the teacher model is then transferred to the student model.

One of the risk factors hindering the improvement of the student model is that the knowledge acquired by a single teacher may be insufficient and biased. A straightforward solution is using an ensemble of multiple teachers for knowledge transfer. The ensemble process takes the predictions of multiple teachers into account and provides comprehensive guidance that helps to improve student performance (You et al., 2017). However, these teachers sometimes conflict with each other, and the heuristic of treating them equally by taking the mean of their predictions ignores the fact that they vary in confidence and correctness, thus often leading to suboptimal performance (Du et al., 2020). Besides, the valuable label information is ignored. Hence, how to ensemble knowledge from multi-teacher and how to utilize the label information are two key questions during the distillation process.

In this work, we introduce a unified algorithm called **Pairwise Iterative Logits Ensemble (PILE)** to tackle these two questions simultane-

[*] Equal contribution.
[†] Corresponding authors.

Proceedings of EMNLP 2022 Industry Track, pages 597–605
December 9–11, 2020. ©2022 Association for Computational Linguistics

ously. The key idea of PILE is to assign a higher weight to teachers that produce more consistent soft targets with the golden labels. The resulting soft targets not only retain the generalization information transferred from the teacher models but also are integrated with the label information annotated by human experts.

We conduct both offline and online experiments and the results validate the effectiveness of PILE. The main contributions of this paper can be summarized as follows:

- We propose PILE, a specially designed ensemble algorithm to tackle multi-teacher distillation and labeled distillation. To the best of our knowledge, PILE is the first work that addresses these two key questions simultaneously in the ranking scenario.
- We conduct extensive offline and online experiments to demonstrate the effectiveness of our proposed method. The results show that PILE effectively boosts a real-world search engine's performance.

2 Related Work

Text Ranking The goal of text ranking is to generate an ordered list of texts in response to a query. Conventional learning-to-rank (LTR) techniques (Li, 2014) are widely used for text ranking, which plays an important role in a wide range of applications, like search engines and recommender systems. LTR techniques can be roughly categorized into three types: pointwise approach (Cooper et al., 1992; Li et al., 2007), pairwise approach (Joachims, 2002; Zheng et al., 2007), and listwise approach (Cao et al., 2007; Burges, 2010). The former two are more widely used in practice as they are easier to optimize.

Recently, deep learning approaches have been widely adopted in ranking and BERT-based ranking models achieve state-of-the-art ranking effectiveness. For example, Nogueira and Cho (2019) use BERT-large (Devlin et al., 2018) as the backbone and feed the concatenation of query and passage text to estimate the relevant scores for passage re-ranking. Nogueira et al. (2019) formulate the ranking problem as a pointwise and pairwise classification problem and tackle them with two BERT models in a multi-stage ranking pipeline. Yilmaz et al. (2019) aggregate sentence-level information to estimate the relevance of the documents and transfer the learned model to capture cross-domain notions of relevance.

However, the performance improvement comes at the cost of efficiency, which limits their real-world application. There are several ways to maintain high performance while keeping efficient computations for BERT-based models, such as knowledge distillation (Hinton et al., 2015b), weight sharing (Lan et al., 2019), pruning (Pasandi et al., 2020; Xu et al., 2020), and quantization (Hubara et al., 2017; Jacob et al., 2018). In this paper, we focus on knowledge distillation which has proven a promising way to compress large models while maintaining performance.

Knowledge Distillation The idea of knowledge distillation was first introduced by Cristian et al. (2006) to train small and fast models to mimic cumbersome and complex models, without much loss in performance. Hinton et al. (2015a) developed this idea further by minimizing the difference between their soft target distribution. With the rise of the pre-training and fine-tuning paradigm, various work has later extended this idea to large-scale pre-trained models and shown impressive results on multiple NLP tasks (Wang et al., 2019; Rajpurkar et al., 2018; Lai et al., 2017) with a significant gain in training efficiency. Sanh et al. (2019) conducted knowledge transfer during the pre-training phase, also known as a task-agnostic way. Sun et al. (2019a) proposed an approach to transfer knowledge between intermediate layers in the fine-tuning stage. Jiao et al. (2020) additionally uses attention-based distillation and hidden states-based distillation for students to imitate teachers' behaviors in intermediate layers. Wang et al. (2020) introduced self-attention relation-based transfer and teacher assistants (Mirzadeh et al., 2020) to further improve the performance of students.

Ensemble Knowledge Distillation There is also some other work exploring the issues of multi-teacher distillation. For example, Du et al. (2020) adaptively ensemble knowledge distillation to find a better optimizing direction for the student network. Wu et al. (2021) designed a co-finetuning framework to jointly finetune multiple teachers for better collaborative knowledge distillation. Li et al. (2021) explored the influence of teacher model adoption which is promising for improving student performance. Different from the above work, we investigate the problem of ensemble knowledge distillation in ranking tasks and use the golden label to supervise the ensemble process.

3 Methodology

3.1 Ranking Task Definition

In a search system, the ranking task aims to measure the relative order of a set of documents $D_q = \{d_i\}_{i=1}^{N}$ given a query $q \in Q$, where Q is a set of user queries and $D_q \subset \mathbb{D}$ is a set of q-related documents retrieved from a large document corpus $\mathbb{D}$ (Liu et al., 2021). The ranking model determines the order of documents by computing the relevance score $f(q, d; \theta)$ of each query-document pair $\{(q, d_i)\}_{i=1}^{N}$, where f is a scoring function parameterized by θ representing the relevance of query q and document d .

As regards training procedure, the ranking model is learned by minimizing the empirical loss over the training data as

$$\mathcal{L} = \sum_{q \in Q} l(Y_{D_q}, F(q, D_q)),$$

where l is the loss function in learning to rank, e.g. pointwise loss, pairwise loss or listwise loss, and $F(q, D_q) = \{f(q, d_i)\}_{i=1}^{N}$ is a set of relevance scores, $Y_{D_q} = \{y_i\}_{i=1}^{N}$ is a set of labels. The label y_i is often assigned an integer range from 0 to 4, representing the relevance of the query-document pair (q, d_i).

3.2 Knowledge Distillation

Due to the resource constraint, the ranking model can not directly serve online in a real production environment and we use knowledge distillation (KD) to compress the model size. In a commonly used KD framework, a large teacher model T is pretrained or finetuned well in advance, and the knowledge of the teacher is transferred to a small student S by minimization of the difference between them, which can be formulated as:

$$\mathcal{L}_{KD} = \sum_{x \in \mathcal{D}} L(f^S(x), f^T(x)),$$

where $\mathcal{D}$ denotes the training dataset and x is the input sample, $f^S(\cdot)$ and $f^T(\cdot)$ represent behavior measurements of teacher and student models, and $L(\cdot)$ is a loss function to measure the difference between their behaviors. The behaviors are usually represented by soft target distributions of the last layer, hidden state distributions or other deep semantic features such as self-attention distributions and embedding layer outputs (Hinton et al., 2015a; Sun et al., 2020; Jiao et al., 2020; Wang et al., 2020, 2021). The methods that transfer the knowledge between the internal layers are limited in generality since the teachers and students are required to have the same model structure or align with each other in the number of layers or the size of hidden layers. Based on this consideration of generality, we perform knowledge distillation on the last layer only.

3.3 Pairwise Iterative Logits Ensemble

Knowledge distillation by one single teacher may bring some bias to the student model while simple yet common mitigation is distillation on the average of logits output by n multiple teachers. However, the teachers with diversity may conflict with each other since the biased teacher contributes equally as the unbiased teacher, which corrodes the confidence of distillation logits. Since the logits produced by teachers on different data vary in the degree of confidence, we conduct a dynamic weighting process for the ensemble. In the ranking task, the logit predicted by the teacher for a document represents a measurement of relevance with the query. The larger logits represent more correlation between the query and the document, and the correlation information is already annotated in its label. Thus, we consider utilizing the golden label to direct the assignment of weight. The procedures of the PILE algorithm are provided in Figure 1.

We start with initializing the ensembled distillation logit $e^{(0)}(q, d)$ for each query q and document d by way of averaging each teacher's outputs:

$$\begin{aligned} w_k(q, d) &= 1 \\ e^{(0)}(q, d) &= \frac{1}{Z(q, d)} \sum_k w_k(q, d) f_k(q, d) \\ Z(q, d) &= \sum_k w_k(q, d), \end{aligned}$$

where $f_k(q, d)$ represents the relevance score predicted by k-th teacher and $w_k(q, d)$ is its weight w.r.t query q and document d.

Then, we perform the iterations of the update procedure. At iteration t, we randomly choose a pair of docs (d_i, d_j) related to the same query q, and check whether the magnitude of their ensemble logits is consistent with their labels. More specifically, if label $y_i > y_j$ and ensemble logits $e^{(t)}(q, d_i) < e^{(t)}(q, d_j)$, we call this pair of docs in

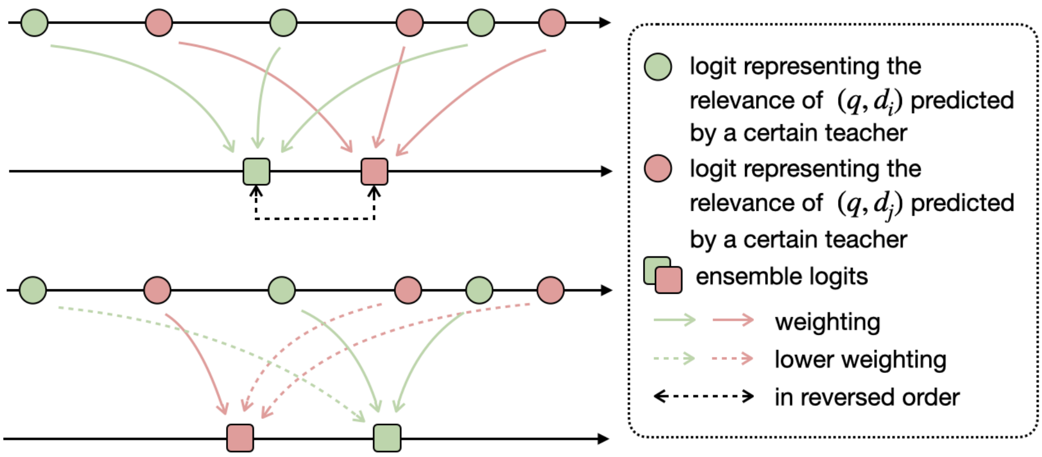
logit representing the relevance of (q, d_i) predicted by a certain teacher
logit representing the relevance of (q, d_j) predicted by a certain teacher
ensemble logits
weighting
lower weighting
in reversed order

Figure 1: Procedures of PILE: 1) start with initializing the ensemble distillation logits with equal weights and 2) update a pair of resulting logits in reversed order by reassigning the weights of teachers.

reversed order or a negative pair and consider the ensemble logits have been biased. We modify the biased ensemble logits by reassigning to zero the weight of teachers responsible for the reversed order error. The reassignment rule can be formulated as follows:

$$w_k(q, d_i) = \begin{cases} 0, & f_k(q, d_i) < e^{(t)}(q, d_i) \\ 1, & otherwise \end{cases}$$
$$w_k(q, d_j) = \begin{cases} 0, & f_k(q, d_j) > e^{(t)}(q, d_j) \\ 1, & otherwise \end{cases}$$

where we assume the docs pair (d_i, d_j) is in reversed order and label $y_i > y_j$ for ease of explanation. Then, we update the ensemble logits with an update rate λ:

$$\tilde{e}^{(t+1)}(q, d) = \frac{1}{Z(q, d)} \sum_i w_k(q, d) f_k(q, d)$$
$$Z(q, d) = \sum_k w_k(q, d)$$
$$e^{(t+1)}(q, d) = (1 - \lambda) e^{(t)}(q, d) + \lambda \tilde{e}^{(t+1)}(q, d)$$

We repeat the updating in an iterative process until the magnitude of the ensemble logits of each pair of docs is consistent with their labels or it reaches the maximum iteration number. We use the final ensemble logits to perform knowledge distillation for the student model.

4 Experiments

To investigate the effectiveness of our proposed method, we conduct offline experiments with baseline models and deploy our proposed model in the real-world production environment. In this section, we report the details of the experiment setups, datasets we used, evaluation metrics, the results of the experiments, and the case study.

Data	#Query	#Query-Doc Pair
log data	635,420,390	2,970,692,361
train data	432,410	8,794,863
test data	12,044	289,835

Table 1: Dataset statistics

4.1 Datasets

The datasets on which we pre-train, finetune, and evaluate our proposed method are collected from the Baidu search engine. For the pre-training stage, we collect a large-scale unlabeled dataset (log data) by means of the anonymous search log. The dataset contains $2,970,692,361$ query-document pairs. As regards finetuning stage, queries and documents are collected from the search pipelines and manually labeled on Baidu's crowdsourcing platform, where a group of hired annotators assigned an integer label range from 0 to 4 to each query-document pair, representing their relevance as {bad, fair, good, excellent, perfect}. We repeat the same process for the test set. The dataset information is summarized in Table 1.

4.2 Training details

We use a 12-layer Transformer (Vaswani et al., 2017) structure with 768 hidden sizes and 12 attention heads as the backbone of teacher models and a 6-layer Transformer structure with 768 hidden sizes and 12 attention heads as student models, where the parameters are randomly initialized, pre-trained and finetuned on the datasets described in section 4.1.

In order to obtain multiple teachers with similar performance and some differences for ensemble knowledge distillation, we use the same pre-trained checkpoint and finetune it on different samplings of the total train data. Specifically, the teacher T_1 is finetuned on the whole train data, and teacher T_2 and T_3 are finetuned on 80% of the whole train data. The intuition behind this is that we want all the teachers to perform relatively well but not behave as the same one otherwise the ensemble of three teachers may degenerate into a single model, which will compromise the benefits of the ensemble knowledge distillation. As a result, T_1 performs best on the test set and the other two teachers have a competitive performance as T_1. The results of three teachers on the test set are shown in Table 2.

In the training procedure, we use the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. For both 12-layer and 6-layer models, we set the learning rate as 2e-5, the batch size as 64, and the warm-up step as 1000. In the PILE procedure, the maximum number of iterations is set to $|D_q|^{\frac{3}{2}}$ where $|D_q|$ is the size of documents set D_q related to a given query q and the update rate λ is set to 0.9. In the knowledge distillation stage, we set the warm-up step as 1000, the learning rate as 2e-5 and the batch size as 1024.

4.3 Evaluation Metrics

The evaluation metrics we used in the experiments are as follows.

The **Positive-Negative Ratio (PNR)** measures the consistency between the golden labels and the scores output by models. For a given query q and a list of N associated documents ranked by model, the PNR can be calculated by this formulation:

$$PNR = \frac{\sum_{i,j\in[1,N]} I\{y_i > y_j\} I\{f(q,d_i) > f(q,d_j)\}}{\sum_{i,j\in[1,N]} I\{y_i > y_j\} I\{f(q,d_i) < f(q,d_j)\}},$$

where $I\{\cdot\}$ is the indicator function, taking the value 1 if the internal statement is true or 0 otherwise. For the test set that contains a good many queries, we average PNR values over all queries.

The **Discounted Cumulative Gain (DCG)** (Järvelin and Kekäläinen, 2000) is a widely used metric that evaluates the ranking result of search engines. More specifically, DCG is calculated as a weighted sum of the document's relevance degree G_i at each position i, where the weight is assigned according to the document's position in the ranking results:

$$DCG = \sum_i \frac{G_i}{log_2(i+1)}$$

The **Interleaving** (Chapelle et al., 2012; Chuklin et al., 2015) is extensively used for comparing a new system with the base system in industrial information systems evaluation. The results of two systems are interleaved and presented together to the end users, whose clicks would be credited to the system that provides the corresponding results. The gain of the new system A over the base system B can be denoted as Δ_{AB}:

$$\Delta_{AB} = 0.5 * \frac{wins(A) - wins(B)}{wins(A) + wins(B) + ties(A,B)},$$

where $wins(A)$(or $wins(B)$) counts the number of times when the results produced by system A (or B) are more preferred than the other system for a given query and $ties(A, B)$ counts the number of times when the two systems are tied.

We also conduct a comparison called **Good vs. Same vs. Bad (GSB)** between two systems by inviting professional annotators to estimate which system produced a greater ranking result for each given query (Zhao et al., 2011). The gain of a new system can be formulated as:

$$\Delta_{GSB} = \frac{\#Good - \#Bad}{\#Good + \#Bad + \#Same},$$

where $\#Good$ (or $\#Bad$) denotes the number of queries that the new (or base) system provides better ranking results and $\#Same$ for the number of results that are equal in quality.

4.4 Offline Experimental Results

We conduct several comparison experiments to verify our proposed method. The models in the offline comparison experiments include:

- **Base**: We use an ERNIE-based ranking model as our base model, which is finetuned with a pairwise loss using human-labeled query-document pairs without any guidelines from teachers;

- **single-KD**: In this setting we add knowledge distillation loss when training the base model using the teacher that performs best on the test set;

Method	PNR	Improvement
Teacher1	3.21	-
Teacher2	3.20	-
Teacher3	3.19	-
Base	3.11	-
+ single-KD	3.15	+1.29%
+ AE-KD	3.16	+1.61%
+ PILE-KD	**3.18**	+2.25%

Table 2: Offline comparison of the proposed methods.

- **AE-KD**: Instead of using the single teacher, this variant uses an ensemble of 3 teachers with averaged weight to perform knowledge distillation;
- **PILE-KD**: When performing knowledge distillation, PILE-KD uses human-annotated labels with the help of the PILE algorithm to conduct a dynamic weighting process for the ensemble of 3 teachers.

The results of each model are shown in Table 2 with the improvement compared to the base model. We also report the performance of the teachers used in knowledge distillation. As we expected, all the distilled models consistently outperform finetuned base model thanks to teacher models' guidance and regularization. And besides, using an ensemble of teachers gains further promotion than the single teacher distillation. After ensembling multiple teachers by averaging distillation logits, the PNR reaches 3.16, exceeding the base by 1.61%. This shows that the remission from biased distillation by the cooperation of multiple teachers improves students in semantic matching. Moreover, by applying the PILE algorithm, we can see that the student can beat the base model by a large margin w.r.t PNR, where the value is improved to 3.18 by 2.25% improvement. It shows the effectiveness of dynamic reduction of biased teachers' weight in the ensemble process.

4.5 Online Experimental Results

To investigate the effectiveness of our proposed method in the real production environment, we deploy the proposed model in Baidu Search, a widely used Chinese search engine, and conduct online experiments for comparison.

The results are presented in Table 3, which comprises the performance comparison regarding ΔDCG, ΔGSB, and Δ_{AB}. We consider the queries from the perspective of types and lengths in the search log. The tail queries and long queries are the queries whose search frequency is lower than 10 times per week or whose length is greater than 10 respectively. Since the heterogeneous search queries follow long-tail distributions, the tail queries make up a significant part of the queries in the search engine. As we can see the proposed method improves the performance of the online ranking system consistently. Particularly, we can observe that the gains of tail queries in the ΔDCG and ΔGSB for our method are 0.27% and 1.62% respectively. Compared with AE-KD, PILE-KD can enable students to retain the ability of teachers as much as possible, and the improvement on long-tail queries also confirms this.

	Random	Tail
ΔDCG	+0.10%	+0.27%
ΔGSB	+3.70%	+1.62%

	Query Type		Query Length	
	Random	Tail	short	long
Δ_{AB}	+0.022%	+0.029%	+0.01%	+0.039%

Table 3: Online comparison of the proposed methods.

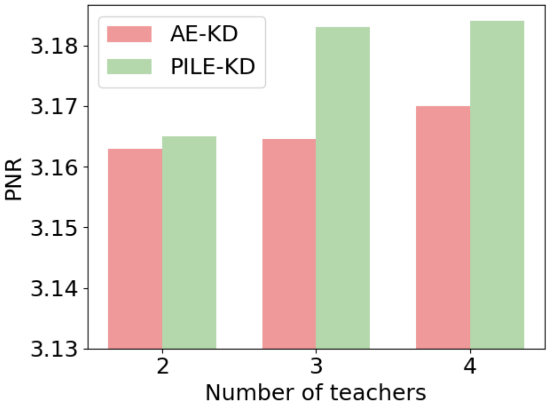

Figure 2: The effect of the number of teachers.

4.6 Ablation Studies

To illustrate the detailed effects of the proposed algorithm, we take a deep insight into the contributions of each setting.

Number of teachers. We first focus on the effect of the number of teachers and the results are shown in Figure2. As expected, the PNRs under both AE-KD and PILE-KD settings increase with the number of teachers and consistently outperform

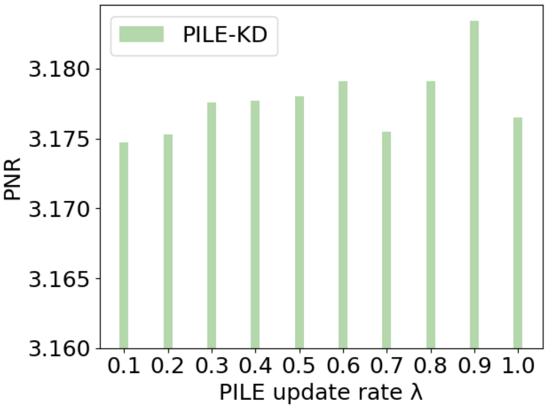

Figure 3: The effect of update rate λ in PILE.

the result of the single-KD model, showing the benefits of using multiple teachers for distillation. Besides, the performance under PILE-KD settings has always been better than it under AE-KD settings, showing the benefits of reducing noise and bias with the help of the label information annotated by human experts in the process of ensemble knowledge distillation.

PILE update rate. We further examine the effect of the PILE update rate λ. As shown in section 3.3, the update rate λ controls the smoothness of two successive iterations. As we can see in Figure 3, the PNR result increases with the λ ranging from 0.1 to 1.0, until reaching the apex at 0.9. The larger update rate λ makes the reliable teachers get more weight in a PILE iteration, resulting in more dependable soft targets. The results further prove that the distillation process can benefit from PILE iteration.

4.7 Case Study

To illustrate the effect of the PILE algorithm in the ensemble knowledge distillation concretely, we show a case that corrects the ensemble logits of two documents that are in reversed order.

As we can see in Table 4, two documents that are related to the same query are labeled by annotators as 0 and 3 respectively, representing their relevance with the query. The teachers' predictions of the relevance are listed in the table as $T_1 \sim T_3$. In the last two rows of the table, we show the ensemble results using averaged weight (AE) and the PILE method respectively.

To get the ensemble logits for knowledge distillation, AE takes the mean of teachers' predictions. However, influenced by individual teachers, the result ensemble logits of the two documents are contrary to their golden labels. In other words, the document with higher relevance is scored lower than the irrelevant one after the ensemble process, which will confuse the student in the knowledge transfer process. Benefiting from the PILE iteration, the teachers consistent with the golden label are assigned more weight. The resulting soft targets not only retain the knowledge that transfers from teachers but also are integrated with the label information annotated by human experts, which is more promising for knowledge distillation.

query	北京驾校教练一个月能挣多少钱？(How much does a Beijing driving school coach earn a month?)	
doc	北京私人教练一个月能挣多少钱？(How much does a Beijing personal trainer make a month?)	驾校教练工资一月多少？百度知道 (How much is the salary of a driving school coach per month? Baidu Knows)
label	0	3
T_1	0.0589	0.0271
T_2	0.1923	0.0331
T_3	0.1057	0.0983
AE	0.1190	0.0528
PILE	0.0590	0.0981

Table 4: The teachers' logits and ensemble logits for two documents.

5 Conclusion

In this work, we propose an easy-to-implement approach to multi-teacher distillation for large-scale ranking models. Our algorithm ensembles multi-teacher logits supervised by human-annotated labels in an iterative way. We conduct the offline experiments as well as deploy our methods in an online commercial search system which demonstrates its superiority.

Acknowledgements

The authors would like to thank the colleagues at Baidu Inc. for their constructive suggestions that helped improve the paper, and hope everything goes well with their work. The authors are also indebted to the anonymous reviewers for their valuable comments and suggestions on the paper.

References

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

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-scale validation and analysis of interleaved search evaluation. *ACM Trans. Inf. Syst.*, 30:6:1–6:41.

Aleksandr Chuklin, Anne Schuth, Ke Zhou, and Maarten De Rijke. 2015. A comparative analysis of interleaving methods for aggregated search. *ACM Trans. Inf. Syst.*, 33(2).

William S Cooper, Fredric C Gey, and Daniel P Dabney. 1992. Probabilistic retrieval based on staged logistic regression. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 198–210.

Bucila Cristian, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

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

Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. 2020. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *Advances in Neural Information Processing Systems*, 33:12345–12355.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015a. Distilling the knowledge in a neural network. *Computer Science*, 14(7):38–39.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015b. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2017. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713.

Kalervo Järvelin and Jaana Kekäläinen. 2000. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

X. Jiao, Y. Yin, L. Shang, X. Jiang, and Q. Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard H. Hovy. 2017. RACE: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

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*.

Hang Li. 2014. Learning to rank for information retrieval and natural language processing. *Synthesis lectures on human language technologies*, 7(3):1–121.

Lei Li, Yankai Lin, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. 2021. Dynamic knowledge distillation for pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ping Li, Qiang Wu, and Christopher Burges. 2007. Mcrank: Learning to rank using multiple classification and gradient boosting. *Advances in neural information processing systems*, 20.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2021. Pretrained transformers for text ranking: Bert and beyond. *Synthesis Lectures on Human Language Technologies*, 14(4):1–325.

Yiding Liu, Weixue Lu, Suqi Cheng, Daiting Shi, Shuaiqiang Wang, Zhicong Cheng, and Dawei Yin.

2021. Pre-trained language model for web-scale retrieval in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3365–3375.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198.

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

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*.

Morteza Mousa Pasandi, Mohsen Hajabdollahi, Nader Karimi, and Shadrokh Samavi. 2020. Modeling of pruning techniques for deep neural networks simplification. *arXiv preprint arXiv:2001.04062*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

S. Sun, Y. Cheng, Z. Gan, and J. Liu. 2019a. 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)*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019b. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

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. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. One teacher is enough? pre-trained language model distillation from multiple teachers. *arXiv preprint arXiv:2106.01023*.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing bert by progressive module replacing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zeynep Akkalyoncu Yilmaz, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Cross-domain modeling of sentence-level evidence for document retrieval. 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)*, pages 3490–3496.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Shiqi Zhao, Haifeng Wang, Chao Li, Ting Liu, and Yi Guan. 2011. Automatically generating questions from queries for community-based question answering. In *Proceedings of 5th international joint conference on natural language processing*.

Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 287–294.

Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4014–4022.