Query Distillation: BERT-based Distillation for Ensemble Ranking

Wangshu Zhang[†] **Junhong Liu**[†] **Zujie Wen**[†] **Yafang Wang** [†*] **Gerard de Melo**[‡]

Ant Financial Services Group[†]
Hasso Plattner Institute, University of Potsdam[‡]

Abstract

Recent years have witnessed substantial progress in the development of neural ranking networks, but also an increasingly heavy computational burden due to growing numbers of parameters and the adoption of model ensembles. Knowledge Distillation (KD) is a common solution to balance the effectiveness and efficiency. However, it is not straightforward to apply KD to ranking problems. Ranking Distillation (RD) has been proposed to address this issue, but only shows effectiveness on recommendation tasks. We present a novel two-stage distillation method for ranking problems that allows a smaller student model to be trained while benefitting from the better performance of the teacher model, providing better control of the inference latency and computational burden. We design a novel BERT-based ranking model structure for list-wise ranking to serve as our student model. All ranking candidates are fed to the BERT model simultaneously, such that the self-attention mechanism can enable joint inference to rank the document list. Our experiments confirm the advantages of our method, not just with regard to the inference latency but also in terms of higher-quality rankings compared to the original teacher model.

1 Introduction

The information retrieval (IR) community has witnessed the flourishing development of neural ranking models in the past several years, examples including DRMM (Guo et al., 2016), DUET (Mitra et al., 2017), PACRR (Hui et al., 2017), and Co-PACRR (Hui et al., 2018). Recently, BERT (Devlin et al., 2018), the pre-trained deep bidirectional Transformer, has shown strong performance on a broad range of language processing tasks and has wide application in ranking tasks as well (MacAvaney et al., 2019; Nogueira and Cho, 2019; Nogueira et al., 2019; Qiao et al., 2019). To further boost the results, it is common to adopt model ensembles, as modern neural ranking models provide a wealth of options for sub-models. The scores of all such sub-models with regard to the relevance of a candidate document to a query are collected and fed into a LambdaMART (Burges, 2010) or XGBoost (Chen and Guestrin, 2016) model to obtain the final ensemble relevance score. However, the computational burden is extremely heavy when drawing on such ensembles, and the prominence of increasingly deep and large neural networks such as BERT exacerbates this problem even more. Furthermore, it is rather inconvenient to update and maintain an ensemble of large models, which is concerning in real-world online deployments.

Knowledge Distillation (KD) is a common approach to balance effectiveness and efficiency (Ba and Caruana, 2014; Hinton et al., 2015). A well-trained large model serves as a teacher for a smaller student model that is trained not only based on the ground truth labeled data, but also using the label distribution emitted by the teacher, such that the student is ultimately able to replace the teacher. However, it is not straightforward to apply KD to ranking problems. First of all, in ranking, we focus on the relative order of documents rather than the label distribution in classification problems, which KD is designed for. On top of this, the total number of documents is often so large that we retrieve only a subset of relevant documents to reduce the size of the ranking list, and computing the overall distribution over all documents is impractical.

[*]Corresponding author, email: yafang.wyf@antfin.com

Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, pages 33–43
Barcelona, Spain (Online), December 12, 2020

Recently, Tang and Wang (2018) proposed Ranking Distillation (RD) to address these issues in the context of recommender systems. Based on the ground truth training data set, the teacher model makes predictions for additional unlabeled documents to obtain a top-k unlabeled document ranking. The student model is then trained to minimize not just the ranking loss on the training data set but also a distillation loss with the example top-k ranking of unlabeled documents generated by its teacher. This method is reminiscent of that of Urner et al. (2011) and has shown its effectiveness in recommender systems.

However, this approach is beset by several problems when considering standard query document ranking tasks. Unlike recommender systems, in the case of search engines, it is rare for the teacher to predict an unlabeled document as being a positive sample, since the set of unlabeled documents consists of all documents that are not in the recall set, and the vast majority of these documents bear little connection with the query. Also, the weight of unlabeled samples needs to be set empirically, making it unclear whether this method can be effective for query document ranking.

In this paper, we propose a substantially different distillation method for ranking tasks. Instead of adding additional unlabeled documents for all the queries in the training data, our *Query Distillation* approach incorporates additional queries. It uses the teacher to predict top-ranked documents for these queries, enabling the teacher to guide the student's training. Furthermore, we divide the student training into two phases: The student model is first trained using teacher-labeled ranking data as teacher guidance and subsequently is fine-tuned using the ground truth labeled training data so as to obtain higher-quality rankings. Through this two-stage fine-tuned training regimen, we hope to obtain a student model that benefits from the strong retrieval effectiveness of the richer model but reduces the inference latency and computational burden.

We adopt a BERT-based model as the student in light of its outstanding performance in ranking tasks (Nogueira and Cho, 2019; Nogueira et al., 2019; Qiao et al., 2019). We also propose a novel BERT QD-list model structure for ranking, which, contrary to the common practice of treating a query document pair (BERT QD-pair) as the input to BERT, jointly considers a query and the entire candidate document list. The self-attention mechanism in BERT allows the query to be evaluated with regard to all documents simultaneously. In addition, a custom attention mask is applied to the KQV self-attention layers to help boost the results. Our experiments show that a single student model outperforms the original large ensemble model through the two-stage teacher–student training and our QD-list model saves substantial inference time while obtaining better results compared to the BERT QD-pair model.

The contributions of our work are two-fold: (1) We introduce a novel two-stage fine-tuned teacher–student training method and obtain a single student model that benefits from the high quality of the ensemble model while reducing the inference latency and computational burden. (2) We propose a list-wise BERT-based ranking model, which through self-attention, allows the model to observe more information while ranking. The experiments confirm the advantages of our contributions.

2 Related Work

2.1 Neural Ranking

The advent of deep learning has brought invigorating new progress to the information retrieval community. Although ranking models have been studied extensively since the mid-2000s, the traditional learning-to-rank paradigm heavily relies on manual feature engineering (Liu and others, 2009; Li, 2011). Commercial web search engines are known to incorporate thousands of carefully designed features, and the feature engineering process is time-consuming, incomplete, and over-specified. In recent years, neural models have attracted attention in light of their ability to obviate the need for handcrafted features. Well-known neural ranking models include DSSM (Huang et al., 2013), DRMM (Guo et al., 2016), DUET (Mitra et al., 2017), PACRR (Hui et al., 2017), and Co-PACRR (Hui et al., 2018).

2.2 Pre-trained Lanuage Models

Recently, neural models pre-trained on language modeling tasks such as ELMo (Peters et al., 2017), Open-AI GPT (Radford et al., 2019), and BERT (Devlin et al., 2018) have achieved impressive results across wide swaths of the NLP landscape. Among these model variants, BERT, the pre-trained deep

bidirectional Transformer, has shown strong performance on search-related tasks, including retrieval-based question answering (Yang et al., 2019a), and numerous experiments confirm that BERT-based methods can outperform state-of-the-art ad-hoc ranking baselines (MacAvaney et al., 2019; Nogueira and Cho, 2019; Nogueira et al., 2019; Qiao et al., 2019).

2.3 Knowledge Distilling

To address the computational overhead of large models, techniques such as the Knowledge Distillation (KD) framework have been proposed (Ba and Caruana, 2014; Hinton et al., 2015). These have shown remarkable potential in accelerating the inference time and improving the performance. Well-trained wide and deep networks are recruited as teachers, and the target student model is supervised not solely by the ground truth, but also by signals from the teacher model. A common approach towards mimicking teacher behaviour is to train the student model to additionally produce a softmax distribution matching that of the teacher model as closely as possible (Hinton et al., 2015). Another way of using teacher guidance for students is to directly mimic the hidden layers of the teacher model (Romero et al., 2014). Learning from multiple neural networks has as well been studied. Distilling an ensemble of neural networks was first introduced by Buciluǎ et al. (2006), where large amounts of pseudo-data are created by a teacher and serve to train a student to approximate the function learned by the teacher model. Learning from multiple teachers also leads to a better student, as shown by You et al. (2017). In their work, multiple teachers are combined via a voting strategy, and the student is required to mimic both the internal layers and the outputs of multiple teachers.

It is not straightforward to apply knowledge distillation to ranking problems, which has only recently been approached in the Ranking Distillation method (Tang and Wang, 2018). By introducing additional teacher-predicted unlabeled documents as teacher guidance, RD shows its effectiveness in recommendation problems, but as mentioned in the introduction, there still remain problems in applying this method to query-based retrieval of documents. Distilled Sentence Embedding (DSE), introduced by Barkan et al. (2020), is a method for sentence embedding distillation that has been shown effective on the GLUE benchmark, but encoding query and documents independently disregards the interaction between query and documents, which is important for ranking problems. In this paper, we will discuss our student training method for ranking problems and propose a novel ranking student model, showing the merit of these ideas.

3 Methodology

3.1 Model Ensemble Ranking

An information retrieval system usually proceeds in two stages. In the first *recall* stage, many potentially relevant documents are collected from a large document index using a simple relevance score such as BM25. These document candidates are then re-ranked according to their predicted relevance to the query in the second stage. Note that, throughout this paper, the term *document* can generally refer to any unit of text being retrieved, such as a passage, sentence, etc.

We focus on the second stage, which has a major contribution to the final result quality. Model ensembling is a common practice to boost the ranking quality and is widely used in state-of-the-art retrieval systems. Specifically, for a given query Q, there are m recalled results $t_0, t_1, \ldots, t_{m-1}, t_i \in T$, $i \in [0, m)$, where T is the set of all available documents. For one such document t_i, n models are separately invoked to score the query–document relevance and all of these scores are combined into a score vector $[s_0^i, s_1^i, \ldots, s_{n-1}^i]$. This feature vector provides the union of all the model scores and is fed into a ranking model such as LambdaMART (Burges, 2010) or XGBoost (Chen and Guestrin, 2016) to obtain the final ensemble score.

The advantages of such model ensembling are straightforward. By drawing on the aggregate ranking abilities of all models, the quality of the ensemble model is normally better than that of any individual model. However, needing multiple models to be deployed online simultaneously may decrease the stability and maintainability of the system.

3.2 Ranking Model Distillation

One method of simplifying a large model while maintaining the result quality is Knowledge Distillation (KD), where a well-trained large model serves as the teacher and a simpler student model is trained to not only predict the ground truth training data, but also mimic the label distribution generated by the teacher. However, it is not straightforward to apply KD to ranking problems. Recently, Tang and Wang (2018) proposed Ranking Distillation (RD) as a means of applying KD to recommendation ranking tasks. The teacher makes predictions on additional unlabeled documents and the top-k ranked list is obtained to train the student based on a distillation loss. However, this method is designed for recommender systems and there remain issues in applying it to query document ranking problems, as discussed in Section 1.

We propose Query Distillation as a novel two-stage distillation method for ranking. First, a large ensemble ranking model is trained on the training data to obtain a high-quality model that can serve as the teacher. Then, instead of adding additional unlabeled documents, we incorporate additional queries and use the recall method as well as the ranking teacher model to produce ranked lists for them. In production systems, additional queries can easily be sampled from query logs. Alternatively, one may also sample keywords and key phrases automatically identified in the document collection.

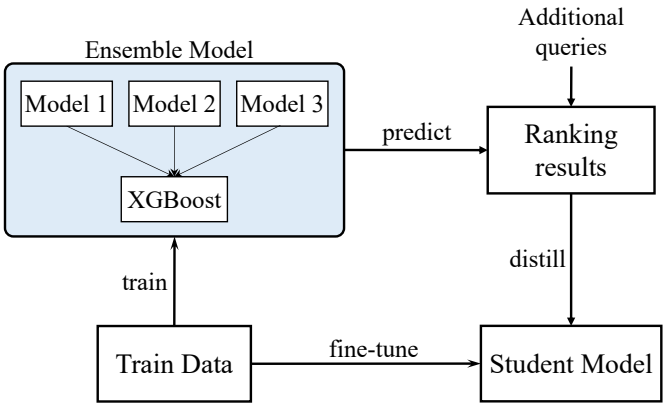

Figure 1: Flowchart of teacher–student training process for an ensemble ranking teacher model

The subsequent student training is guided by the ranking results provisioned by the teacher as well as the ground truth training data. The overall procedure is illustrated in Figure 1. The two stages of student training are as follows:

- *Teacher Guidance.* The student model is first trained based on ranking data generated by the teacher model using supplementary queries.
- *Ground Truth Training.* In the second stage, higher-quality ground truth labeled data is used to fine-tune the student model to attain a better performance.

By applying the two-stage teacher-student training method, we can obtain a student model that benefits from the retrieval quality of the ensemble, while reducing the inference latency and computational cost.

3.3 Student Model

We adopt BERT as our base student model. The standard practice to invoke BERT in ranking tasks is to form pairs of each query and candidate document token sequence. Previous work shows the effectiveness of this structure (Nogueira and Cho, 2019; Nogueira et al., 2019). We refer to this structure as query document pair BERT (BERT QD-pair). In this paper, we advance a new query document list BERT approach (BERT QD-list) that excels both in retrieval quality and inference efficiency. The two model structures are contrasted in Figure 2. We are given a query as a sequence of tokens $Q = (q_0, q_1, \ldots, q_{n-1})$ and top-k document candidates $T_0, T_1, \ldots, T_{k-1}$, where the token sequence of each document is $T_i = (t_0^i, t_1^i, \ldots, t_{l_i}^i)$, and l_i is the sequence length of T_i. Instead of pairing each query and document token sequence, BERT QD-list squeezes the entire document list into a single input sequence. All document tokens are tied together as the second sequence input of the BERT model, and the query tokens as the first sequence input are expected to reveal the most relevant documents among all candidates. We place a marker token [RANK] before each document to represent the document token sequence. Ultimately, all pertinent ranking information between the query and document resides in the marker token.

The advantages of BERT QD-list are two-fold. First of all, it makes the online model prediction more efficient. Furthermore, compared to the input of BERT QD-pair, where the query is only associated with one single document at a time, BERT QD-list enables list-wise joint ranking among all documents.

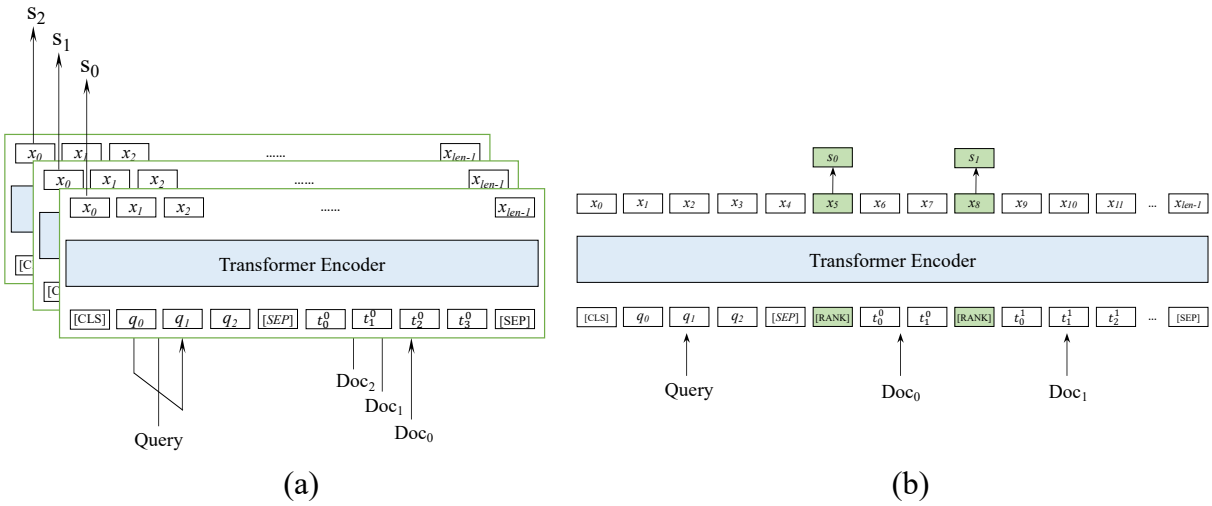

Figure 2: Structure of our student model. Figure (a) shows vanilla query document pair BERT (BERT QD-pair) and Figure (b) illustrates our query document list BERT (BERT QD-list) variant.

3.4 Loss Function

We add an output layer to the final Transformer output to transform every output token to a scalar value, in which the value of the positions of [RANK] tokens are gathered as the output score of every corresponding document. A hinge loss is adopted for our loss function:

$$L = \sum_i \Delta\text{nDCG}_i \ \max(0, \lambda - (\hat{s} - s_i)) \tag{1}$$

Here, $\hat{s}$ is the score of the correct document, and s_i denotes the other document scores. ΔnDCG stands for the absolute difference in nDCG score (Burges, 2010) between the correct document and the current document. λ is a constant set to be 0.1. Note that if there is only one correct document, softmax cross-entropy loss is also an option.

3.5 Multi-Head Self-Attention

Self-attention (Devlin et al., 2018) has proven capable of capturing long-distance dependency information between sentences and attending to evidence information in many tasks such as machine translation, reading comprehension, and text classification. We also adopt self-attention to help capture the relationship between the query and the candidate documents. We represent the L layers of the Transformer as $\mathbf{H}^l = \text{Transformer}_l(\mathbf{H}^{l-1}), l \in [1, L]$, where $\mathbf{H}^l = [h_0^l, h_1^l, \ldots, h_{n-1}^l]$, n being the sequence length. For the l-th BERT layer, the output of a self-attention head is

$$\mathbf{Q} = \mathbf{H}^{l-1}\mathbf{W}_l^Q, \quad \mathbf{K} = \mathbf{H}^{l-1}\mathbf{W}_l^K, \quad \mathbf{V} = \mathbf{H}^{l-1}\mathbf{W}_l^V \tag{2}$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \tag{3}$$

$$\mathbf{A}_l = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T} + \mathbf{M}}{\sqrt{d_k}}\right)\mathbf{V}_l \tag{4}$$

One of the most important challenges of our multiple document ranking problem is how to better make sense of the relationship between the query and the document tokens. The vanilla Transformer works to some extent, but since each pair of tokens has an attention dependency, noise may be introduced between candidate ranking documents. To address this issue, we propose several attention mask patterns applied to the KQV self-attention layer of the BERT model. We wish to grant query tokens access to affect all

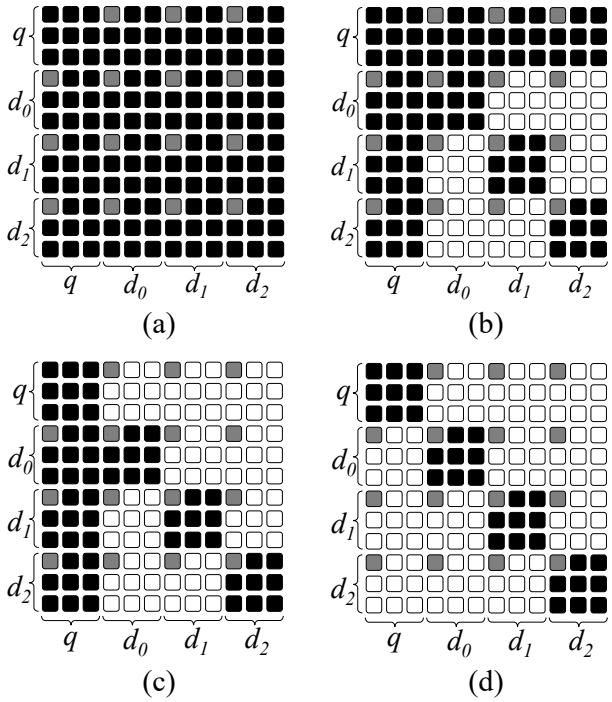

Figure 3: Different patterns of attention masks. q represents the query tokens, while d_k stands for tokens of a document k. For simplicity, only three ranking candidates are illustrated. A dark cell at row i and column j indicates that the i-th token is allowed to attend to the j-th token. A light cell indicates that the self-attention is inhibited between the corresponding pair of i and j. Cells shaded in gray denote the attention between different [RANK] marker tokens.

ranking documents, while the ties between candidate document token are restricted to prevent introducing noise. Only the special [RANK] tokens should attend to each other so as to enable the communication and comparison between candidate documents. We design our own mask matrix to control how tokens can attend to each other, as is shown in Figure 3. We can distinguish four kinds of mask patterns.

1. *No mask (Figure 3(a)).* The numbers in the attention matrix are all 1, and hence all tokens can attend to each other.
2. *Mutual document mask (Figure 3(b)).* Document tokens are inhibited from attending to tokens from other documents, which guarantees that noise from other document tokens cannot creep in, while the [RANK] tokens, which represent the overall document relevance, can have access to each other.
3. *Document–query mask (Figure 3(c)).* Compared to the second pattern, the output document tokens cannot affect any of the query tokens, while the query tokens can have attention bridges to documents, which may make the output document tokens cleaner.
4. *Segment-level self-attention (Figure 3(d)).* The query and each individual document possess only local self-attention, with the exception of the special [RANK] marker token, which can access other [RANK] tokens as well as query tokens.

4 Experiments

4.1 Experimental Settings

4.1.1 Dataset

We conduct experiments on two datasets. The first one is a retrieval based question answer dataset called Alipay Question Answers (Alipay QA), with queries collected from an online customer service with Mandarin Chinese data. For a given query, the system retrieves the most relevant answer from a document database with around $8,000$ documents, which cover the majority of user queries. The collected dataset consists of $38,017$ queries, each with 10 recall document candidates, within which the correct answer is labeled manually. It should be noted that there may be more than one suitable answer for a given query. In addition, another $500,000$ queries and recall document candidates are also collected for distillation usage.

The second dataset is MS MARCO (Microsoft Machine Reading Comprehension) by Nguyen et al. (2016), a large-scale English language dataset focused on machine reading comprehension, question answering, and passage ranking. The passage ranking sub-task consists of a collection of given queries q, each with $1,000$ relevant document passages selected by BM25, among which the most suitable passages that can answer the query have been marked manually. A large number of $532,761$ labeled queries and passages are available as training data. To fit our ranking distilling paradigm, we extract $50,000$ queries for training and test usage respectively, and for each query we keep the correct passage and randomly select 4 other recalled passages as recall candidates, which makes 5 recall documents. In addition, the remainder of around $400,000$ queries and recall documents is used as distillation data.

4.1.2 Baselines and Parameters Setting

An ensemble model is first trained for all datasets. For simplicity, only a BERT-based single model is adopted and we use query document pair BERT and query document list BERT as base single models. We trained two instances for both model types with different parameters so that in total four models are built for the ensemble. For the Alipay QA dataset, the best single model top-1 accuracy is 84.57%, and XGBoost is used for model ensembling, which attains a top-1 accuracy of 86.00%. Similarly, we also build an XGBoost ensemble model for the MS MARCO dataset. We obtain a best single model top-1 accuracy of 91.80%, versus 92.60% for the XGBoost model ensemble. All single and ensemble model baseline results are given in Table 1.

The ensemble model is then invoked as a teacher to label the additional queries and recalled documents. Student models are trained with our two-stage teacher–student training regimen, and we conduct distillation experiments on both the QD-pair and QD-list variants. The results for different model complexities are also collected by gradually reducing the number of hidden layers of the model.

	Alipay $\mathrm{QA}_{10\,\mathrm{recall}}$		$\mathrm{MS\,MARCO}_{5\,\mathrm{recall}}$	
Model	ACC@1	MRR@5	ACC@1	MRR@5
Best Single	84.57	91.14	91.80	95.59
Ensemble	86.00	91.95	92.60	95.92
BERT RD	83.87	90.61	92.00	95.66
BERT QD-pair	86.27	92.18	92.82	96.11
BERT QD-list	**86.83**	**92.49**	**92.91**	**96.17**

Table 1: ACC@1 and MRR@5 of student models compared to the best single model and the original ensemble model on both Alipay QA and MS MARCO datasets (%).

To draw a fair comparison among all experiments, we adopt the same BERT configuration, initialized with the parameters provided by the pre-trained BERT-base model (Devlin et al., 2018). The Chinese BERT base model is used for Alipay QA, while the English version is used for MS MARCO. The maximum sequence length of BERT QD-pair and BERT QD-list on Alipay QA is set to 64 and 228, respectively, and on MS MARCO it is set to 256 and 512, respectively. We apply Adam optimization (Kingma and Ba, 2014) with a learning rate of 4×10^{-5}, and adopt a dropout probability of 0.1. We consider the mutual document masking from Section 3.5 our default masking procedure, except where indicated otherwise.

4.1.3 Evaluation Measure

For both datasets, we consider a) whether the top-1 ranked document can answer the user's query, as well as b) the overall ranking quality. We thus adopt top-1 accuracy (ACC@1) and top-5 Mean Reciprocal Rank (MRR@5) (Burges, 2010) to evaluate the retrieval quality. Additionally, the inference time is also taken into consideration so as to evaluate the model's time efficiency.

4.2 Experimental Results

4.2.1 Ensemble Ranking Model Distillation

The main experimental results are given in Table 1. We first focus on the 12-layer model distilling method. BERT RD stands for the Ranking Distillation method by Tang and Wang (2018) using a base BERT model. BERT QD-pair and BERT QD-list represent our two-stage distillation of the query document pair BERT model and query document list BERT model, respectively.

#Layers	Model	Alipay $QA_{10\,recall}$			$MS\,MARCO_{5\,recall}$		
		ACC@1	MRR@5	Speed	ACC@1	MRR@5	Speed
12	BERT QD-pair	86.27	92.18	2.6	92.82	96.11	1.5
	BERT QD-list	**86.83**	**92.49**	6.6 (×**2.5**)	**92.91**	**96.17**	2.8 (×**1.9**)
9	BERT QD-pair	86.26	92.21	3.5	92.66	96.01	1.8
	BERT QD-list	**86.56**	**92.38**	8.8 (×**2.5**)	**92.90**	**96.15**	3.7 (×**2.1**)
6	BERT QD-pair	86.24	92.22	5.6	**92.62**	**96.01**	2.6
	BERT QD-list	**86.43**	**92.28**	13.0 (×**2.3**)	92.37	95.87	5.5 (×**2.1**)
3	BERT QD-pair	85.24	91.62	10.5	**91.77**	**95.48**	5.3
	BERT QD-list	**85.27**	**91.63**	25.7 (×**2.4**)	90.91	95.00	11.7 (×**2.2**)

Table 2: ACC@1 and MRR@5 of student model with various numbers of hidden layers (%). The inference time is as well reported in the Speed column, which represents the inference throughput of model (#samples/sec.).

BERT RD shows very limited effectiveness on our query document ranking tasks, since the additional distillation loss is inaccurate for our datasets, as discussed in Section 1. In contrast, both BERT QD-pair and BERT QD-list perform better than the large ensemble teacher model and far better than the original single models, which demonstrates the effectiveness of introducing our two-stage fine-tuned distilling method. While both distilled models outperform the original ensemble model, BERT QD-list obtains better results than the BERT QD pair variant. We conjecture that providing the document list to the model for a single joint prediction helps query tokens exchange information between all documents simultaneously during the KQV attention process. It is more difficult for the BERT QD-pair model to compare between documents and choose the best one because query tokens can only be cross-referenced with tokens from a single document at a time. In addition, the larger the number of candidate documents, the more benefit it appears we can obtain from BERT QD-list, since a larger number of documents can be compared. The experiments show that BERT QD-list outperforms QD-pair by 0.56% (ACC@1) and 0.31% (MRR@5) in absolute percentage points on Alipay QA, which has 10 document candidates, while the respective gains are 0.09% and 0.06% on MS MARCO, which has 5 recall documents.

4.2.2 Model Complexity

To get better inference time efficiency while retaining most of the retrieval quality, we can further shrink our distill model to a smaller size. For simplicity, model complexity is controlled by altering the number of hidden layers of BERT. We conduct experiments reducing the 12 hidden layers to 9, 6, and 3, respectively, as also reported in Table 2. The experiments show that the retrieval quality of both models drops as the number of hidden layers decreases, due to the reduction in model capacity. This reduction is non-linear: Taking BERT QD-list on Alipay QA as an example, ACC@1 decreases by 0.40% when the number of hidden layers drops from 12 to 6, but more drastically falls by 1.19% with a further reduction from 6 to 3. The overall results suggest that 6 hidden layers strikes a good balance between retrieval quality and model complexity. Note that at 3 hidden layers, BERT QD-pair shows little difference with BERT QD-list on Alipay QA and even outperforms the latter on MS MARCO, which suggests that the BERT list model requires a greater model capacity to handle the intricate relationships and comparisons between the query and the various documents. The simpler BERT QD-pair model thus constitutes an alternative in resource-constrained circumstances.

4.2.3 Inference Time Efficiency

In many situations, it is not enough for a model to be highly accurate. It also has to meet stringent time and space requirements. To assess this, we provide one single query document list sample at a time to obtain a prediction and measure how many samples the model can handle per second. All results are collected using PyTorch version 1.0.1 with Python 3.6.8 on a server equipped with Intel Xeon E5-2682 v4 @ 2.50GHz CPU, and the prediction service can access only up to 8 CPU cores. The time efficiency results are shown in the Speed column of Table 2. We collect data of the average number of samples the

model can process per second for various numbers of hidden layers and also report the multiplier in time efficiency of the BERT QD-list model compared to the BERT QD-pair model.

We consistently observe an inference time speed-up by using BERT QD-list. On Alipay QA, this amounts to a roughly 2.4 times faster inference of BERT QD-list compared to BERT QD-pair, and it is around 2.1 times faster on MS MARCO. The computational time efficiency of BERT QD-list comes from the fact that all input documents are compressed into a single long token sequence, while the BERT QD-pair model handles a batch of shorter token sequences at a time. The larger the number of candidate documents, the bigger the benefit in time efficiency, as confirmed by the result that the time efficiency advantage is greater on the 10 recall documents for Alipay QA in comparison to the MS MARCO dataset, which has 5 recall documents. In addition, although the throughput of the model increases as the number of hidden layers is reduced, the advantage of BERT QD-list over BERT QD-pair on both datasets remains reasonably consistent.

4.2.4 Mask Patterns

We further assess further variants of self-attention masking, as described in Section 3.5, on both datasets using BERT QD-list with 12 hidden layers. The results are compiled in Table 3. On Alipay QA, we observe that the Mutual-Doc and Doc-Query mask patterns, as illustrated in Figure 3(b) and (c), prevail over models without any mask applied. On MS MARCO, the advantage is weaker, since we have much fewer documents. Segment-level masking hampers the model performance dramatically on both datasets, thus establishing the importance of self-attention to bridge query tokens and document tokens.

	Alipay $\text{QA}_{10\,\text{recall}}$		$\text{MS MARCO}_{5\,\text{recall}}$	
Mask Type	ACC@1	MRR@5	ACC@1	MRR@5
No Mask	86.38	92.25	92.90	96.14
Mutual-Doc	**86.83**	**92.50**	**92.91**	**96.15**
Doc-Query	86.54	92.37	92.86	96.14
Segment	86.29	92.16	91.91	95.58

Table 3: ACC@1 and MRR@5 obtained for various mask patterns with the BERT QD-list model on both datasets (%). The different mask patterns are described in Figure 3.

4.2.5 Sequence Length Limitation

We finally evaluate the influence of the length of input sequence. Taking our 12-layer BERT QD-list model as an example, Table 1 shows that on the Alipay QA dataset the throughput of our model is 6.6 when the sequence length is 228, while on the MS MARCO dataset the throughput is 2.8 with a sequence length of 512, which means that a 2.25 times larger sequence length results in a 2.36 times slower model inference time. Furthermore, the largest position embedded in the pre-trained BERT model is 512, so we would need to train our own BERT position embedding if sequence lengths exceed 512. With larger inference time and sequence length limitation, there appear to be obstacles when applying our model to scenarios involving a large recall set and long input sequence lengths. In practice, a sophisticated ranking system often has multiple ranking processes, which are piped together to gradually obtain fewer but more accurate results. Our list model can be a good choice at the end of the ranking pipeline, since it delivers strong results but is limited by the input sequence length. Another option is to incorporate small changes to the network architecture for much greater scalability to long inputs (Beltagy et al., 2020).

5 Conclusion

In this paper, we introduce Query Distillation as a two-stage fine-tuned distillation training process for large ensemble ranking models. Furthermore, we propose a novel list-wise BERT model structure for ranking tasks (BERT QD-list), which is used as our student model. The experiments confirm the advantages of our query document list model not just with regard to the inference latency but also with regard to the retrieval quality over regular query document pair BERT (BERT QD-pair) as well as the original teacher models. In the future, we will try to apply our ranking distillation method on further tasks, such as answer selection in machine reading. Additionally, BERT variants that can handle larger sequence lengths such as XLNet (Yang et al., 2019b) will be evaluated to process even more documents at a time.

References

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? In *NIPS*, pages 2654–2662.

Oren Barkan, Noam Razin, Itzik Malkiel, Ori Katz, Avi Caciularu, and Noam Koenigstein. 2020. Scalable attentive sentence pair modeling via distilled sentence embedding. In *AAAI*, pages 3235–3242.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *SIGKDD*, pages 535–541. ACM.

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

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *SIGKDD*, pages 785–794. ACM.

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*.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, pages 2333–2338. ACM.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. A position-aware deep model for relevance matching in information retrieval. In *Proceedings of EMNLP 2017*, pages 1049–1058. ACL.

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

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

Hang Li. 2011. Learning to rank for information retrieval and natural language processing. *Synthesis Lectures on Human Language Technologies*, 4(1):1–113.

Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. Cedr: Contextualized embeddings for document ranking. In *SIGIR*, pages 1101–1104. ACM.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*, pages 1291–1299. International World Wide Web Conferences Steering Committee.

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.

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*.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.

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

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Jiaxi Tang and Ke Wang. 2018. Ranking distillation: Learning compact ranking models with high performance for recommender system. In *SIGKDD*, pages 2289–2298. ACM.

Ruth Urner, Shai Shalev-Shwartz, and Shai Ben-David. 2011. Access to unlabeled data can speed up prediction time. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 641–648.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Shan You, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Learning from multiple teacher networks. In *SIGKDD*, pages 1285–1294. ACM.