T^{2}Ranking: A Large-scale Chinese Benchmark for Passage Ranking

Xiaohui Xie
xiexiaohui@mail.tsinghua.edu.cn
DCST, Tsinghua University.
Zhongguancun Lab.
Beijing, China

Qian Dong
dq22@mails.tsinghua.edu.cn
DCST, Tsinghua University.
Zhongguancun Lab.
Beijing, China

Bingning Wang
bryantwwang@tencent.com
Tencent Inc.
Beijing, China

Feiyang Lv
feiyanglv@tencent.com
Tencent Inc.
Beijing, China

Ting Yao
tessieyao@tencent.com
Tencent Inc.
Beijing, China

Weinan Gan
carrygan@tencent.com
Tencent Inc.
Beijing, China

Zhijing Wu
wuzhijing.joyce@gmail.com
Beijing Institute of Technology
Beijing, China

Xiangsheng Li
lixsh6@gmail.com
Tencent Inc.
Beijing, China

Haitao Li
liht22@mails.tsinghua.edu.cn
DCST, Tsinghua University.
Zhongguancun Lab.
Beijing, China

Yiqun Liu
yiqunliu@tsinghua.edu.cn
DCST, Tsinghua University.
Zhongguancun Lab.
Beijing, China

Jin Ma
daniellwang@tencent.com
Tencent Inc.
Beijing, China

ABSTRACT

Passage ranking involves two stages: passage retrieval and passage re-ranking, which are important and challenging topics for both academics and industries in the area of Information Retrieval (IR). However, the commonly-used datasets for passage ranking usually focus on the English language. For non-English scenarios, such as Chinese, the existing datasets are limited in terms of data scale, fine-grained relevance annotation and false negative issues. To address this problem, we introduce T^{2}Ranking, a large-scale Chinese benchmark for passage ranking. T^{2}Ranking comprises more than 300K queries and over 2M unique passages from real-world search engines. Expert annotators are recruited to provide 4-level graded relevance scores (fine-grained) for query-passage pairs instead of binary relevance judgments (coarse-grained). To ease the false negative issues, more passages with higher diversities are considered when performing relevance annotations, especially in the test set, to ensure a more accurate evaluation. Apart from the textual query and passage data, other auxiliary resources are also provided, such as query types and XML files of documents which passages are generated from, to facilitate further studies. To evaluate the dataset, commonly used ranking models are implemented and tested on T^{2}Ranking as baselines. The experimental results show that T^{2}Ranking is challenging and there is still scope for improvement. The full data [1] and all codes are available at https://github.com/THUIR/T2Ranking/

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07.
https://doi.org/10.1145/3539618.3591874

CCS CONCEPTS

• **Information systems** → **Evaluation of retrieval results**.

KEYWORDS

Test collection, Passage retrieval, Passage re-ranking, Passage ranking, Search evaluation

ACM Reference Format:
Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, and Jin Ma. 2023. **T^{2}Ranking**: A Large-scale Chinese Benchmark for Passage Ranking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591874

1 INTRODUCTION

Passage ranking is a crucial component of information retrieval systems. The promising performance of passage ranking leads to satisfaction of search users and benefits multiple IR-related applications, e.g., question answering [1] and reading comprehension [17]. Typically, passage ranking encapsulates two coherent stages, i.e., passage retrieval and passage re-ranking. The goal of passage ranking is to compile a search result list ordered in terms of relevance to the query from a large passage collection. The first stage, passage retrieval, needs to recall relevant passages from a massive

[1] The dataset is licensed under the Apache License 2.0

passage corpus. Hence, efficiency should also be considered besides effectiveness [10]. The second stage, passage re-ranking, may employ models that focus more on effectiveness to re-rank passages retrieved in the first stage.

To support the passage ranking research, various benchmark datasets are constructed. Some of them support both the first-stage retrieval (FR) and second-stage re-ranking (SR) task, while others focus on the SR task. We present the summary of data statistics of some common datasets in Table 1. Commonly-used datasets focus on English scenarios. For example, Trec Complex Answer Retrieval (Car) [6], TriviaQA [11] and MS-MARCO [16]. Among them, MS-MARCO is a large-scale dataset with 8.8 million passages. The queries are question-based, and human-generated answers are provided by annotators. Subsequently, a binary relevance score can be obtained by determining whether the answer related to the query exists in the passage; that is, relevant (1) for passages containing the answer and non-relevant (0) for those that don't. Following the success of MS-MARCO, similar datasets have also been constructed in the non-English community, such as Chinese. For example, mMarco-Chinese [3] is the Chinese version of the original MS-MARCO with the help of machine translation. $\text{DuReader}_{\text{retrieval}}$ [20] adopts a similar paradigm that generates binary relevance judgments for query-passage pairs from human-generated answers. Multi-CPR [15] is a multi-domain Chinese dataset for passage retrieval, with three different domains and a certain amount of human-annotated query-passage pairs. Besides, Sogou-SRR [29], Sogou-QCL [30] and TianGong-PDR [27] are provided based on user logs from Sogou[2], a popular Chinese search engine.

Although existing datasets facilitate the development of passage ranking applications, there are several limitations that need to be addressed:

(1) The datasets are neither large-scale nor human-generated, especially in the Chinese community. Sogou-SRR and Tiangong-PDR involve a limited number of queries. Although mMarco-Chinese and Sogou-QCL are large-scale, the former is based on translation while the latter only embraces click labels. Recently, two passage-ranking datasets with considerable data scales are constructed, namely, $\text{DuReader}_{\text{retrieval}}$ and Multi-CPR.
(2) Fine-grained human annotations are limited. Most datasets apply binary relevance annotations. Since Roitero et al. [24] show the benefit of fine-grained relevance scales, recent work also investigates fine-grained relevance annotations beyond binary (coarse) paradigms. However, the number of fine-grained annotations is quite limited, for example, less than 100K in Sogou-SRR and TianGong-PDR.
(3) False negative problem harms the accuracy of evaluation. As Arabzadeh et al. [2] point out, the existing passage ranking datasets suffer from the false negative problem, i.e., relevant results are labeled as irrelevant. This problem exists mainly due to limited human annotations in the large-scale dataset, which will harm the accuracy of the evaluation. For example, for each query in Multi-CPR, only one passage will be marked as positive while others are regarded as negative. A recent dataset, i.e., $\text{DuReader}_{\text{retrieval}}$, attempts to ease this issue by asking annotators to manually check and relabel the passages in the top retrieved results pooled from multiple retrievers.

In order to ensure high-quality training and evaluation of passage ranking models, we construct and release a new Chinese dataset, named T^2Ranking, comprising of more than 307K question-based queries and over 2.3M passages extracted from 1.8M web documents. Specifically, we sample search queries from user logs of the Sogou search engine, a popular search system in China, and perform query preprocessing, such as filtering pornographic queries and non-interrogative queries, and removing similar queries, to obtain a clean and high-quality query set with 307K queries (50K for the test set). For each query, we extract the content of corresponding documents from different search engines and remove vertical results (e.g., image search results and video search results) and duplicate results for the following process. To ensure the semantic integrity of each passage, we train and use a passage segment model to access passages from each document, which gives us around 1.3 passages per document. We then use a passage clustering approach to discard highly similar passages and generate the query-passage pool. Moreover, we also record query types and other auxiliary resources of documents to facilitate extending studies (e.g., multimodal tasks and out-of-domain (OOD) tasks). For a given query and its corresponding passages, we hire expert annotators to provide 4-level relevance judgments of each query-passage pair and adopt an active learning-based data sampling to improve the efficiency and quality of annotation. All hired annotators are full-time staff engaged in annotation work.

We carry out comprehensive analyses and present comprehensive statistics of the proposed dataset. Additionally, we conduct comprehensive experiments to evaluate the performance of multiple passage retrieval models as well as passage re-ranking models, on T^2Ranking. The experimental results show that T^2Ranking is a highly challenging task and there is still potential for further performance improvement.

In summary, we make the following contributions:

- We build a large-scale Chinese dataset, named T^2Ranking for passage ranking (retrieval and re-ranking). T^2Ranking contains more than 300K queries and over 2M unique passages, and also comes with fine-grained relevance annotations, along with query types, document titles and XML files as multimodal information.
- We leverage multiple strategies to ensure the high quality of our dataset, such as using a passage segment model and a passage clustering model to enhance the semantic integrity and diversity of passages and employing active learning for annotation method to improve the efficiency and quality of data annotation.
- We conduct extensive experiments to evaluate the performance of existing passage retrieval and re-ranking models on T^2Ranking. Experimental results show room for further improvement which might be brought by more sophisticated models in the future.

2 RELATED WORK

There are several benchmark datasets developed for passage ranking. For datasets that have relevance annotations for all query-passage pairs, both passage retrieval and passage re-ranking tasks

[2]https://www.sogou.com/

Table 1: The data statistics of datasets commonly used in passage ranking. Qrys (Psgs): Queries (Passages). FR(SR): First (Second)-stage of passage ranking, i.e., passage Retrieval (Re-ranking).

Dataset	Lang	#Qrys	#Psgs	Qrys.source	Psgs.source	Annotation	Task
Trec Car [6]	EN	2M	30M	Wiki doc.	Wiki doc.	Binary	SR
TriviaQA [11]	EN	95K	650K	Trivia Web.	Wiki./Web doc.	Binary	FR, SR
MS-MARCO [16]	EN	516K	8.8M	User logs	Web doc.	Binary	FR, SR
Sogou-SRR [29]	CN	6K	63K	User logs	Web doc.	Fine-grained	SR
Sogou-QCL [30]	CN	537K	9M	User logs	Web doc.	Click labels	SR
TianGong-PDR [27]	CN	70	11K	User logs	News doc	Fine-grained	FR, SR
mMarco-Chinese [3]	CN	516K	8.8M	User logs	Web doc.	Binary	FR, SR
Multi-CPR [15]	CN	303K	3M	User logs	Result Title	Binary	FR, SR
DuReader$_{\text{retrieval}}$ [20]	CN	97K	8.9M	User logs	Web doc.	Binary	FR, SR
T^{2}Ranking(Ours)	CN	307K	2.3M	User logs	Web doc.	Fine-grained	FR, SR

can be tested. Other datasets, however, only focus on passage re-ranking tasks, providing relevance annotations only for query-passage pairs in which the passages have been extracted from the initial result lists recalled by the first-stage retrievers. We use FR to denote the first stage of passage ranking, i.e., passage retrieval and SR to denote the second stage of passage ranking, i.e., passage re-ranking as shown in Table 1.

Commonly used datasets for passage ranking are constructed for the English community. Trec Complex Answer Retrieval (CAR) [6] uses topics, outlines, and paragraphs extracted from Wikipedia. For the training set, a passage is considered relevant if it is found within the Wikipedia pages of the topic and non-relevant otherwise. The test set, comprised of 113 complex topics, has 50 passages per topic that are manually annotated. TriviaQA [11] gathers question-answer pairs from 14 trivia and quiz-league websites and passages from Wikipedia and web documents. MS-MARCO [16] is widely utilized due to its large scale. Unlike Trec Car and TriviaQA, queries in MS-MARCO are sourced from user-generated queries, which are question-based, from the Bing search engine[3]. The Passages are extracted from realistic web documents returned by the same search engine. Then human editors are recruited and instructed to create a natural language answer with the correct information extracted strictly from the passages provided given particular queries. The relevance levels of passages in both TriviaQA and MS-MARCO are determined in a binary fashion, based on whether or not the passages contain facets of the true answer to a given query.

For the Chinese community, there exist several datasets designed for training and evaluating passage ranking models. Drawing upon the Sogou search engine, three datasets have been established, namely Sogou-SRR [29], Sogou-QCL [30] and TianGong-PDR [27]. Sogou-SRR (Search Result Relevance) consists of 6K queries and corresponding top 10 search results. For each search result, the screenshot, title, snippet, HTML source code, parse tree, URL as well as a 4-grade relevance score and the result type are provided. Sogou-QCL is a large-scale dataset compromised of 537K queries and more than 9 million Chinese web pages. Rather than human-generated relevance judgments, relevance levels of query-result pairs are assessed based on click labels. Queries from Tiangong-PDR are collected from Sogou's search logs, while passages are obtained from Web pages data from the Sina news website[4]. Moreover, four-grade human-assessed relevance labels for each query-passage pair are available. Besides, mMarco-Chinese [3] is constructed via machine translation from MS-MARCO. However, these datasets are not large-scale and/or human-generated. Recently, Qiu et al. [20] propose a new dataset, named DuReader$_{\text{retrieval}}$, for benchmarking the passage retrieval models from Baidu search[5]. Similar to MS-MARCO, queries in DuReader$_{\text{retrieval}}$ are question-based, and human-generated answers are collected to access the relevance levels of passages. Long et al. [15] build Multi-CPR which is a multi-domain dataset for passage ranking. Queries and passages for Multi-CPR are gathered from three different vertical search systems: E-commerce, Entertainment Video, and Medical. Rather than being extracted from web documents, passages in Multi-CPR refer to titles of search results, such as product titles in E-commerce search, resulting in shorter passage lengths. Human annotators have been recruited to judge the relevance level (binary) of the query-passage pairs. For each query, the most semantically relevant passage is marked as positive, while the others are marked as negative.

3 TASK DEFINITION

In this section, we formally define the tasks in T^{2}Ranking. Our proposed dataset focuses on two stages of passage ranking, namely, passage retrieval and re-ranking. This aligns with the pipeline of modern information retrieval systems, which follows the retrieval-then-re-ranking paradigm.

The goal of passage retrieval is to retrieve candidate passages in response to a given query. Given a query q, a retrieval model is used to retrieve a candidate set of passages $\mathcal{K} = \{\mathbf{p}_j^{\mathbf{q}}\}_{j=1}^{K}$ from a large corpus $\mathcal{G} = \{\mathbf{p}_i\}_{i=1}^{G}$, where $K \ll G$. In particular, a passage consists of a sequence of words $\mathbf{p} = \{w_p\}_{p=1}^{|\mathbf{p}|}$, where $|\mathbf{p}|$ represents the length of passage $\mathbf{p}$. Similarly, a query is a sequence of words $\mathbf{q} = \{w_q\}_{q=1}^{|\mathbf{q}|}$. The main challenge in passage retrieval lies in efficiently retrieving the relevant passages for a query, given the vast number of passages in the corpus. Following retrieval, re-ranking is proposed to derive

[3]https://www.bing.com

[4]https://www.sina.com.cn/

[5]https://www.baidu.com/

a permutation over $\mathcal{K}$, such that the more relevant passages are ranked higher in the list. In contrast to the retrieval task, the re-ranking task demands that models have a strong capability for relevance modeling, which is capable of capturing subtle semantic differences between relevant passages in the candidate set $\mathcal{K}$.

4 DATASET CONSTRUCTION

In this section, we present the construction details of T^{2}Ranking. We begin by introducing the overall pipeline of dataset construction, which includes query sampling, passage extraction, and relevance annotation. We then provide important technical details used in the data construction, such as model-based passage segmentation, clustering-based passage de-duplication, and active learning-based data sampling.

4.1 Overall Pipeline

The overall pipeline of constructing T^{2}Ranking involves several steps, including query sampling, document retrieval, passage extraction and fine-grained relevance annotation.

Query sampling. We sample real user queries from the query pool of Sogou and perform pre-processing (e.g. de-duplication and normalization of redundant spaces and question marks) to obtain a clean query dataset. Then, we filter out pornographic, non-interrogative and resource-request type queries and queries that might include user information from T^{2}Ranking using an intent analysis algorithm, to ensure that the dataset consists only of high-quality, question-based queries. Note that resource-request-type queries are used to search for specific music, film resources, etc.

Document retrieval. We retrieve a comprehensive set of documents for each query from popular search engines such as Sogou, Baidu, and Google, taking advantage of their vast resources and expertise in indexing and ranking web content. This helps to reduce the issue of false negatives, as each system covers different parts of the web and can return different relevant documents, hence improving the overall coverage of our dataset.

Passage extraction. The construction of passages in T^{2}Ranking involves segmentation and de-duplication. Rather than using a heuristic approach to segment passages from a given document (e.g, conventionally determining the start or the end of passages by line breaks), we employ a model-based method for passage segmentation to maximize the preservation of complete semantics in each passage (detailed in Section 4.2). Additionally, we introduce a clustering-based technique to enhance the efficiency of annotation and maintain the diversity of the annotated query-passage pairs (detailed in Section 4.3). This approach effectively removes nearly identical passages that are retrieved by a particular query. The resulting segmented and de-duplicated passages are subsequently merged into the passage collection for T^{2}Ranking.

Fine-grained relevance annotation. All hired annotators are experts in providing annotation for search-related tasks and have engaged in labeling work for a long time. At least three annotators provide 4-level fine-grained annotations for each query-passage pair. Specifically, if the annotations are inconsistent among the first three annotators for a particular pair (three annotators provide three different scores), a fourth annotator will be asked to access it. In cases where all four annotators are inconsistent, the query-passage pair is considered to be too ambiguous to determine the required information and will be excluded from the dataset. The final relevance label for each query-passage pair is determined by major voting. Following the criteria of TREC benchmarks [5], we also define the instructions of 4-level relevance annotation as:

- **Level-0.** There is a complete mismatch between the content of the query and the passage.
- **Level-1.** The passage is relevant to the query, but it does not meet the required information needs of this query.
- **Level-2.** The passage is relevant to the query and partly satisfies its information needs.
- **Level-3.** The passage content is customized to satisfy the information needs of the query and precisely contains the answer to the query.

We show several examples in Table 2. The fine-grained 4-level annotation enables accurate evaluation of passage re-ranking tasks. Notably, for the retrieval task, we consider **Level-2** and **Level-3** passages as relevant passages, and all other passages are regarded as irrelevant passages.

Notably, when processing the test queries, we utilize the strategy of annotating all query-passage pairs after the passage segmentation process, which attempts to mitigate the problem of false negatives in our test set and hence provides a more precise evaluation of the retrieval and re-ranking performance. For the training queries, we employ the aforementioned clustering-based method to du-duplicate the passages which are then presented to the recruited expert annotators to obtain 4-level fine-grained annotations. This strategy not only enhances the efficiency of annotation but also maintains diversity in the annotated query-passage pairs. Besides, the success of the active learning strategy motivates us to rich the information involved in our training samples by choosing informative query-passage pairs for annotation. The key idea behind active learning is that by allowing the model to select which training samples it wants to learn from and focus on samples that are most valuable for improving its performance, leading to more efficient annotation. In T^{2}Ranking, we design an active learning-based method to annotate the training data in an iterative manner (detailed in Section 4.4). Overall, the data construction pipeline of T^{2}Ranking is formally defined in Alg. 1.

4.2 Model-based Passage Segmentation

Typically, in existing datasets, the passages are segmented from documents according to a natural paragraph or sliding window with a fixed length. However, the natural paragraph-based segmentation usually results in an excessively long passage containing multiple topics, considering most web documents are not well-written. Besides, the sliding window-based segmentation often leads to a lack of complete semantics in a passage [4, 20], thereby reducing the reliability of the dataset for the evaluation of the passage retrieval and re-ranking.

To address this issue, we propose a model-based method for passage segmentation. A segmentation model is trained on well-written web documents using the sequence labeling task. Specifically, we

Table 2: Examples for annotation of query-passage pair.

#Annotation	Query	Passage	Explanation
0	Does burning tea leaves with chrysanthemums produce tea polyphenols?	Tea and chrysanthemum are allowed to be infused together. Chrysanthemum itself has the effect of...	There is no mention of query in the passage and it does not meet the information needs of query at all.
1	What does cervical cancer stage IIB mean?	Stage IB cervical cancer means that the cancer is confined to the cervix and the colposcopy reveals lesions larger...	This passage pertains to cervical cancer, but it is not suitable for the given query, which requests information on stage IIB.
2	What causes excessive sweating in children?	A child's sweating in bed is due to night sweats in children...	The query is not limited to the topic of sweating during sleep only.
3	What flavour bait does the chub like?	The best bait for chub is a feed with an aromatic, fishy or smelly smell and fresh...	The answer is precisely contained in the passage.

Algorithm 1: The pipeline of dataset construction.

Input: Query pool $\mathcal{Q}$, document pool $\mathcal{D}$, passage segmentation model $Seg(\cdot)$, cross-encoder $CE(\cdot)$ and expert annotator $\mathcal{H}$
Output: Fine-grained relevance labels $\mathcal{L}$

```
DatasetConstruction(Q, D, H) begin
    Q = Sample(Q); % sampling a set of queries Q;
    L = ∅, P = ∅; % initialising a label set and a passage set;
    for q ∈ Q do
        %retrieving a set of documents D for query q via
         multiple search engines;
        D = MultiSearchEngines(q, D);
        for d ∈ D do
            P ∪ Seg(d);% passage segmentation;
        end
        if q is a training query then
            % cluster-based passage de-duplication for
              training query;
            P = De-duplication(P);
            if CE(·) is ready then
                % filtering out the query-passage pairs with
                  high certainty;
                P = Filter(CE(q, P));
            end
        end
        % passages of test queries are all annotated to
          alleviate the false negative issue;
        for p ∈ P do
            % annotation for query-passage pair;
            L ∪ H(q, p);
        end
    end
    return L
end
```

use the Sogou Baike[6], Baidu Baike[7] and Chinese Wikipedia[8] as the training data, given that these web documents are generally well-written and their natural paragraphs are clearly defined. An example English version Wikipedia is shown in Figure 1. Given a web document $\mathbf{d} = \{w_d\}_{d=1}^{|\mathbf{d}|}$, we utilize a segmentation model $Seg(\cdot)$ to determine whether a given word w_d should be separated. Formally, the sequence labeling task can be defined as

$$\hat{y}_d = Seg(\mathbf{d})_{w_d}, \tag{1}$$

$$\mathbf{L}_s = \text{CrossEntropy}(y_d, \hat{y}_d), \tag{2}$$

where the $\hat{y}_d$ is the predicted score for segmentation. The true label y_d represents whether the word w_d is the last word of a paragraph. The segmentation model $Seq(\cdot)$ is trained based on the loss defined in Eq. 2. If $\hat{y}_d \geq \sigma$, then the passage is segmented from its document by the d-th word. σ is a hyperparameter that controls the degree of segmentation. The smaller the value of σ, the more passages are segmented.

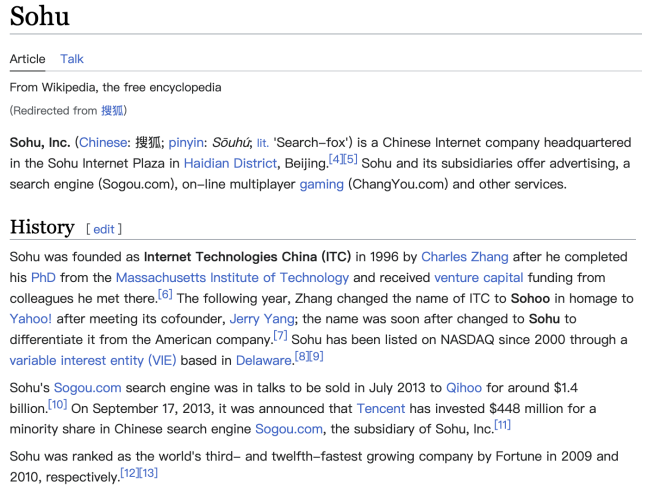
4.3 Clustering-based Passage De-duplication

Annotating a large number of highly similar passages on the web would be redundant and meaningless. In this paper, we propose a clustering-based method for passage de-duplication, which leads to more efficient annotation. Specifically, we employ a hierarchical clustering algorithm, Ward [26], to unsupervisedly cluster similar passages together. The passages in the same cluster are considered nearly duplicated. Consequently, we select only one passage from each cluster for annotation. It is worth noting that we only conduct the de-duplication in the training set. For the queries in the test set, we annotate all the passages obtained from the passage segmentation model to alleviate the false negative issue as much as possible. Intuitively, passages that are nearly identical under a specific query provide little information gain to a ranking model compared to passages with significant differences. Practically, false negatives within the same cluster as an annotated true positive can be easily filtered by a cross-encoder [21]. Therefore, we employ the

[6]https://baike.sogou.com/

[7]https://baike.baidu.com/

[8]https://zh.wikipedia.org/



Sohu

Article Talk

From Wikipedia, the free encyclopedia

(Redirected from 搜狐)

Sohu, Inc. (Chinese: 搜狐; pinyin: *Sōuhú*; lit. 'Search–fox') is a Chinese Internet company headquartered in the Sohu Internet Plaza in Haidian District, Beijing.[4][5] Sohu and its subsidiaries offer advertising, a search engine (Sogou.com), on–line multiplayer gaming (ChangYou.com) and other services.

History [edit]

Sohu was founded as **Internet Technologies China (ITC)** in 1996 by Charles Zhang after he completed his PhD from the Massachusetts Institute of Technology and received venture capital funding from colleagues he met there.[6] The following year, Zhang changed the name of ITC to **Sohoo** in homage to Yahoo! after meeting its cofounder, Jerry Yang; the name was soon after changed to **Sohu** to differentiate it from the American company.[7] Sohu has been listed on NASDAQ since 2000 through a variable interest entity (VIE) based in Delaware.[8][9]

Sohu's Sogou.com search engine was in talks to be sold in July 2013 to Qihoo for around $1.4 billion.[10] On September 17, 2013, it was announced that Tencent has invested $448 million for a minority share in Chinese search engine Sogou.com, the subsidiary of Sohu, Inc.[11]

Sohu was ranked as the world's third– and twelfth–fastest growing company by Fortune in 2009 and 2010, respectively.[12][13]

Figure 1: Illustration for a web document from Wikipedia which is well-written with clearly defined paragraphs.

de-duplication to save the annotation cost while retaining more diverse training samples for improving model performance.

4.4 Active Learning-based Data Sampling

In practice, we observe that not all training samples can further enhance the ranking model's performance. Training samples, that can be easily predicted accurately by a model, are unlikely to provide useful information for model training.

To address this issue, we borrow the light of active learning [22], using a model to choose more informative training samples for further annotations. Active learning is a framework that enables models to participate in the data annotation process. The aim of active learning is to minimize the amount of annotated data required while maintaining or improving model performance. Formally, active learning is an iterative process where the model makes predictions on a pool of unannotated samples. The samples with the highest uncertainty or informativeness are selected for annotation by annotators, and the annotated samples are added to the training data. The model is then updated with the newly annotated data. The framework of active learning is illustrated in Figure 2. Concretely, a query-passage re-ranking model, specifically a cross-encoder, is trained using data constructed from the initial stage. In the second stage, unannotated query-passage pairs are obtained and evaluated for relevance by the cross-encoder. Pairs with high confidence scores are filtered out as they do not provide significant information for further performance improvement, while pairs with low confidence scores, which are typically considered noise samples, are also eliminated. The remaining pairs are submitted to annotators for fine-grained annotation. The annotated query-passage pairs are then added to the training set and the cross-encoder is updated with newly acquired samples.

5 DATA STATISTICS

This section presents the data statistics of T^{2}Ranking.

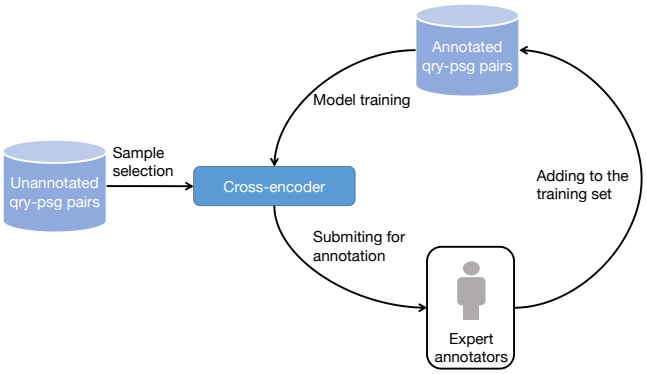


Figure 2: Illustration for the framework of active learning.

Table 3: Statistic of queries in T^{2}Ranking.

	Quantity	Max. length	Mean.length
Training set	258,042	40	11.1
Test set	49,662	38	10.99

Query. Table 3 provides a summary of the statistics of queries in T^{2}Ranking. The maximum and mean lengths of queries in the training and test sets are nearly identical. We further analyze the domain distribution of queries in the training and test sets, as demonstrated in Figure 3. Domain tags are provided by the Sogou search engine. The query domain distribution in the training and test sets is consistent, and the queries cover a broad range of domains. We also demonstrate the diversity level of queries by resorting to the metric, intra-list similarity (ILS) [31] which can be defined as

$$s(\mathbf{q}_i, \mathbf{q}_j) = \frac{\text{BERT}(\mathbf{q}_i)_{[cls]} \cdot \text{BERT}(\mathbf{q}_j)_{[cls]}}{\|\text{BERT}(\mathbf{q}_i)_{[cls]}\| \|\text{BERT}(\mathbf{q}_j)_{[cls]}\|}, \tag{3}$$

$$\text{ILS}_Q = \frac{\sum_{i=1}^{|Q|} \sum_{j=i+1}^{Q} s(\mathbf{q}_i, \mathbf{q}_j)}{\sum_{i=1}^{Q} \sum_{j=i+1}^{Q} 1}, \tag{4}$$

where BERT [13] is a pre-trained language model that is often used as the backbone model for various tasks [7–9, 12, 18]. A lower ILS score indicates a lower similarity between queries in the benchmark, thus indicating a higher level of diversity. We calculated the ILS scores of T^{2}Ranking, as well as those of several popular datasets, such as MSMARCO, Multi-CPR and DuReader$_{\text{retrieval}}$. The results are shown in Table 4. From the table, it is evident that the queries in T^{2}Ranking are more diverse, as indicated by a lower ILS score. Note that, T^{2}Ranking compromises queries with higher diversity even than those in Multi-CPR, which contains queries from different vertical search applications.

Document & Passage. T^{2}Ranking comprises passages extracted from 1,752,482 web documents, with a total of 2,303,643 passages after segmentation. On average, each web document is divided into 1.31 passages of which the mean length is 632.6.

Relevance Annotation. We display the distribution of the 4-level relevance annotations in Figure 4. In the training set, on average, each query is annotated with 6.25 passages, while in the test set, each query is annotated with an average of 15.75 passages.

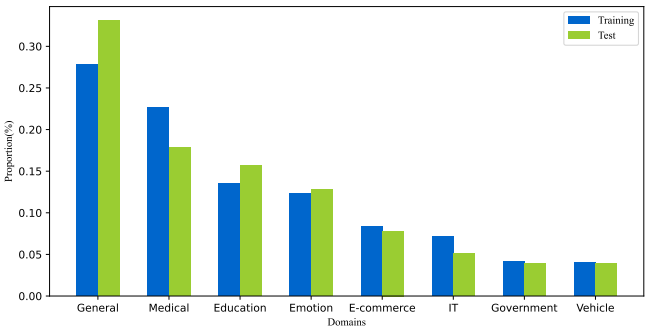
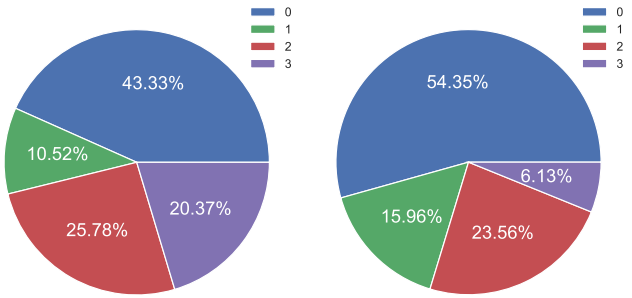


Figure 3: Domain statistics for the training and test queries in T²Ranking.

Table 4: ILS scores of different datasets. Lower ILS scores refer to higher diversity levels of queries.

Dataset	ILS Score
MS-MARCO [16]	0.227
Multi-CPR [15]	0.186
$\text{DuReader}_{\text{retrieval}}$ [20]	0.152
T²Ranking (ours)	**0.144**



(a) The distribution of relevance annotations in the training set.

(b) The distribution of relevance annotations in the test set.

Figure 4: Pie chart of the annotation distribution.

6 EXPERIMENTS AND RESULTS

Consistent with modern information retrieval systems, the *retrieval-then-re-ranking* paradigm is utilized in our experiments. In this section, we examine the performance of commonly-used retrievers and re-rankers on T²Ranking.

6.1 Retrieval Performance

Baselines. Existing retrieval models can be broadly divided into sparse retrieval models and dense retrieval models. Sparse retrieval models focus on exact matching signals to design a relevance scoring function, with BM25 being the most prominent and widely-utilized baseline due to its promising performance. Additionally, dense retrieval models leverage deep neural networks to learn low-dimensional dense embeddings for queries and documents.

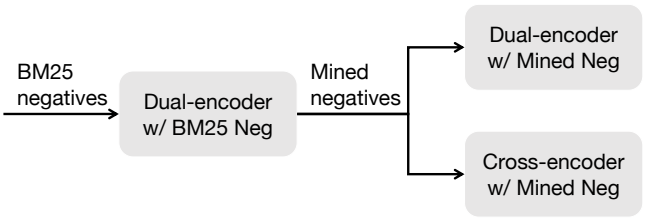
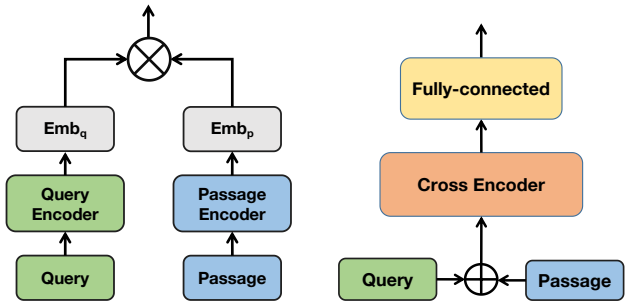


Figure 5: Illustration for the training process of baselines used in our experiments. First, we train a dual-encoder with BM25 negatives, which is similar to DPR [12]. Second, we train the dual-encoder and cross-encoder with the global negative sampling strategy proposed in several studies [15, 20].



(a) Dual-encoder. **(b) Cross-encoder.**

Figure 6: Illustration for the architecture of dual-encoder and cross-encoder.

Generally, most existing dense retrieval methods adhere to the cascade training paradigm [15, 20, 21]. Therefore, to facilitate easier comparison in future studies on our dataset, we simplify the training process as illustrated in Figure 5 as in [15, 20]. Specifically, we utilize the dual-encoder (DE) as the architecture of dense retrieval models, which is illustrated in Figure 6(a). The following methods are employed as our baselines to evaluate the retrieval performance on T²Ranking.

- **QL** (query likelihood) [19] is a representative statistical language model that measures the relevance of passages by modeling the generation of a query.
- **BM25** [23] is a widely-used sparse retrieval baseline.
- **DE w/ BM25 Neg** is equivalent to DPR [12], which is the first work that uses the pre-trained language model as the backbone for the passage retrieval task.
- **DE w/ Mined Neg** enhance the performance of DPR by sampling hard negatives globally from the entire corpus as in ANCE [28] and RocketQA [21].
- **DPTDR** [25] is the first work that employs prompt tuning for dense retrieval.

Among them, QL and BM25 are sparse retrieval models, whereas the others are dense retrieval models

Implementation details. BM25 is implemented by Pyserini [14] with default parameters. The dual-encoder models are implemented

Table 5: Performance of retrieval models on the test set of T^2Ranking.

	MRR@10	Recall @50	Recall@1K
QL	.2803	.3915	.6858
BM25	.3579	.4918	.7426
DE w/ BM25 Neg	.4877	.7123	.9104
DE w/ Mined Neg	.5191	.7357	.9147
DPTDR	**.5285**	**.7423**	**.9211**

by the deep learning framework PyTorch on up to 8 NVIDIA Tesla A100 GPUs (with 80G RAM). We use the off-the-shelf Chinese $\text{BERT}_{\text{base}}$ to initialize the dual-encoder. The maximal length of queries and passages are set to 32 and 256, respectively. The negatives are sampled from the top 200 passages recalled by BM25 or DE w/ BM25 Neg. The ratio of positive:negative is set to 1:1. We train the dual-encoder for 100 epochs with a learning rate of 3e-5.

Metrics. The following evaluation metrics are used in our experiments to examine the retrieval performance of baselines on T^2Ranking: (1) Mean Reciprocal Rank for the top 10 retrieved passages (MRR@10), (2) Recall for the top-K retrieved passages (Recall@K). Notably, for the retrieval task, we consider **Level-2** and **Level-3** passages as relevant passages, and all other passages are regarded as irrelevant passages. For a comprehensive comparison, we report Recall@50 and Recall@1K on the test queries. Following the evaluation settings of MS-MARCO and $\text{DuReader}_{\text{retrieval}}$, MRR is defined as the average of the reciprocal ranks of the *first* relevant passage for a set of queries. The MRR is a value between 0 and 1, with a higher value indicating that the system is better at ranking the most relevant passage higher in the list. Meanwhile, Recall is defined as the fraction of relevant passages that are retrieved among all relevant passages, also with a value between 0 and 1, where a higher value indicates that the system is better at retrieving all relevant passages. MRR and Recall measure different aspects of retrieval performance. MRR@K and Recall@K can be depicted as:

$$MRR@K = \frac{1}{|Q|} \sum_{q \in Q} \frac{\mathbf{I}(rank \leq K)}{rank}, \tag{5}$$

$$Recall@K = \frac{\mathbf{I}(rank_p^{\mathcal{K}^q} \leq K)}{\sum_{q \in Q} \sum_{p \in R_q} 1}. \tag{6}$$

where $\mathbf{I}(\cdot)$ is a indicator function. The $rank$ in Eq. 5 denotes the position of the *first* relevant passage in the retrieved candidates of query q. The R_q and $rank_p^{\mathcal{K}^q}$ represent the relevant passages of query q and the position of passage p in the candidate list $\mathcal{K}^q$.

Retrieval performance. We report the retrieval performance of baselines in Table 5. Compared to the traditional sparse retrieval method BM25, dual-encoder models significantly boost the retrieval performance on our dataset. The improvement can be attributed to the integration of two distinct sources of knowledge, i.e., latent knowledge obtained through unsupervised pre-training of language models on a massive corpus and relevance knowledge acquired through supervised training on our large-scale annotated dataset. Equipped with the strategy of negative mining proposed in recent studies [28], the retrieval performance of dual-encoder models could be further improved on T^2Ranking. It is worth noting that the Recall@K metrics observed in T^2Ranking are lower than those reported in other benchmarks with coarse-grained annotations. For instance, the Recall@50 of BM25 is .601 and .700 on MS-MARCO-DEV Passage and $\text{DuReader}_{\text{retrieval}}$, respectively, and 0.4918 on our dataset. In the test set of T^2Ranking, we have a greater number of passages annotated with fine-grained relevance labels, leading to a 4.74 average positive paragraph per query, which makes the retrieval task more difficult and eases the false negative problem to some extent. This highlights the challenging nature of T^2Ranking and the potential for further improvement in the future.

6.2 Re-ranking Performance

Baselines. Due to the smaller number of passages considered by re-rankers, they tend to use the cross-encoder architecture rather than the dual-encoder architecture. The cross-encoder approach allows for a more detailed interaction between queries and documents, resulting in better performance, although at the expense of lower efficiency. We report the re-ranking performance of the cross-encoder model, which is trained on the hard negatives mined from the entire corpus, as depicted in Figure 5. The architecture of cross-encoder is illustrated in Figure 6(b).

Implementation details. The cross-encoder is implemented in the same experimental environment as the dual-encoder, with a maximum input length of 288. Negatives are sampled from the top 256 passages retrieved by the dual-encoder, and a positive-to-negative ratio of 1:128 is set. The cross-encoder is then trained for 5 epochs with a learning rate of 3e-5.

Metrics. To evaluate the re-ranking performance of the cross-encoder, we use two ranking metrics: MRR@10 and nDCG@K. In the test set of T^2Ranking, the average number of annotated passages per query is 15.7, with a maximum of 100 annotated passages. We report nDCG@20 and nDCG@100 on the test queries. nDCG@K normalizes DCG@K by dividing DCG@K by the iDCG@K, which is the DCG@K of ideal ordering of the passages. DCG@K discounts the graded relevance value of a passage according to the rank that it appears at, which can be defined as:

$$DCG@K = \frac{1}{|Q|} \sum_{q \in Q} \sum_{p \in \mathcal{K}^q} \frac{L(p)}{\log_2(rank_p^{\mathcal{K}^q} + 1)}, \tag{7}$$

$$nDCG@K = \frac{DCG@K}{iDCG@K}, \tag{8}$$

where $L(p)$ is the graded relevance of passage p.

Re-ranking performance. The re-ranking performance of the cross-encoder is shown in Table 6. The results indicate that re-ranking the candidates retrieved by the dual-encoder significantly outperforms re-ranking the candidates retrieved using the BM25 method. The improved performance is attributed to the higher recall rate achieved by the dual-encoder method, which is consistent with previous studies conducted on other benchmarks [15, 20]. The re-ranking performance on T^2Ranking, however, is lower compared to other benchmarks [15, 20]. This can be explained by the presence of more fine-grained annotated relevant passages and queries with higher diversities in T^2Ranking, which makes it a more challenging benchmark but also provides a more accurate reflection of re-ranking performance.

Table 6: Performance of cross-encoder with mined negatives on the test set of T²Ranking.

Candidates	MRR@10	nDCG@20	nDCG@100
BM25's top-1000 psg.	.5184	.4401	.4696
DE's top-1000 psg.	.5520	.5149	.5571

7 CONCLUSION

In this study, we introduce T²Ranking, a large-scale benchmark for Chinese passage ranking that involves both retrieval and re-ranking tasks. To construct a high-quality dataset, we leverage various strategies, including model-based passage segmentation, clustering-based passage de-duplication and active learning-based data sampling. Specifically, we adopt a model-based method for passage segmentation in T²Ranking, which aims to maximize the preservation of complete semantics in each passage. To balance the efficiency of annotation with the diversity of annotated query-passage pairs, we incorporate a clustering-based technique in T²Ranking to remove highly similar passages retrieved by a specific query, which helps streamline the annotation process without compromising the overall quality of the dataset. The adoption of an active learning strategy in the construction of T²Ranking enhances the efficiency of annotating more informative training samples. The active learning framework enables the dataset to be continuously updated with the most valuable samples while minimizing the number of annotations required to achieve optimal performance. Furthermore, to ensure high-quality annotation, expert annotators are involved in the implementation of a 4-level fine-grained annotation scheme for both the training and test sets in T²Ranking. This scheme allows for more nuanced modeling of IR models during training and a more precise evaluation of the models during testing. In summary, T²Ranking encompasses over 300K queries and more than 2M unique passages, with around 2.4 million query-passage pairs annotated with fine-grained relevance labels by expert annotators. To the best of our knowledge, T²Ranking is the largest Chinese benchmark with fine-grained annotation for passage ranking. We believe that this dataset will make a significant contribution to the IR community and the advancement of IR technology.

ACKNOWLEDGMENTS

This work is supported by the Tsinghua-Tencent Tiangong Institute for Intelligent Computing, the Beijing Academy of Artificial Intelligence (BAAI) and the Quan Cheng Laboratory.

REFERENCES

[1] Elif Aktolga, James Allan, and David A Smith. 2011. Passage reranking for question answering using syntactic structures and answer types. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*. Springer, 617–628.

[2] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* 25, 4 (2022), 365–385.

[3] Luiz Bonifacio, Vitor Jeronymo, Hugo Queiroz Abonizio, Israel Campiotti, Marzieh Fadaee, Roberto Lotufo, and Rodrigo Nogueira. 2021. mmarco: A multilingual version of the ms marco passage ranking dataset. *arXiv preprint arXiv:2108.13897* (2021).

[4] Anfeng Cheng, Yiding Liu, Weibin Li, Qian Dong, Shuaiqiang Wang, Zhengjie Huang, Shikun Feng, Zhicong Cheng, and Dawei Yin. 2023. Layout-aware Webpage Quality Assessment. *arXiv preprint arXiv:2301.12152* (2023).

[5] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Ellen M Voorhees, and Ian Soboroff. 2021. TREC deep learning track: Reusable test collections in the large data regime. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 2369–2375.

[6] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview.. In *TREC*.

[7] Qian Dong, Yiding Liu, Suqi Cheng, Shuaiqiang Wang, Zhicong Cheng, Shuzi Niu, and Dawei Yin. 2022. Incorporating Explicit Knowledge in Pre-trained Language Models for Passage Re-ranking. *arXiv preprint arXiv:2204.11673* (2022).

[8] Qian Dong and Shuzi Niu. 2021. Latent Graph Recurrent Network for Document Ranking. In *Database Systems for Advanced Applications: 26th International Conference, DASFAA 2021, Taipei, Taiwan, April 11–14, 2021, Proceedings, Part II 26*. Springer, 88–103.

[9] Qian Dong, Shuzi Niu, Tao Yuan, and Yucheng Li. 2022. Disentangled graph recurrent network for document ranking. *Data Science and Engineering* 7, 1 (2022), 30–43.

[10] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, et al. 2022. Pre-training methods in information retrieval. *Foundations and Trends® in Information Retrieval* 16, 3 (2022), 178–317.

[11] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551* (2017).

[12] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[13] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. 2.

[14] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2356–2362.

[15] Dingkun Long, Qiong Gao, Kuan Zou, Guangwei Xu, Pengjun Xie, Ruijie Guo, Jian Xu, Guanjun Jiang, Luxi Xing, and Ping Yang. 2022. Multi-CPR: A Multi Domain Chinese Dataset for Passage Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3046–3056.

[16] 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 *CoCo@ NIPs*.

[17] Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2018. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 647–656.

[18] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *arXiv preprint arXiv:1910.14424* (2019).

[19] Jay M Ponte and W Bruce Croft. 2017. A language modeling approach to information retrieval. In *ACM SIGIR Forum*, Vol. 51. ACM New York, NY, USA, 202–208.

[20] Yifu Qiu, Hongyu Li, Yingqi Qu, Ying Chen, Qiaoqiao She, Jing Liu, Hua Wu, and Haifeng Wang. 2022. DuReader_retrieval: A Large-scale Chinese Benchmark for Passage Retrieval from Web Search Engine. *arXiv preprint arXiv:2203.10232* (2022).

[21] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2010.08191* (2020).

[22] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)* 54, 9 (2021), 1–40.

[23] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[24] Kevin Roitero, Eddy Maddalena, Gianluca Demartini, and Stefano Mizzaro. 2018. On fine-grained relevance scales. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 675–684.

[25] Zhengyang Tang, Benyou Wang, and Ting Yao. 2022. DPTDR: Deep Prompt Tuning for Dense Passage Retrieval. *arXiv preprint arXiv:2208.11503* (2022).

[26] Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association* 58, 301 (1963), 236–244.

[27] Zhijing Wu, Jiaxin Mao, Yiqun Liu, Jingtao Zhan, Yukun Zheng, Min Zhang, and Shaoping Ma. 2020. Leveraging passage-level cumulative gain for document ranking. In *Proceedings of The Web Conference 2020*. 2421–2431.

[28] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[29] Junqi Zhang, Yiqun Liu, Shaoping Ma, and Qi Tian. 2018. Relevance estimation with multiple information sources on search engine result pages. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 627–636.

[30] Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Sogou-qcl: A new dataset with click relevance label. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1117–1120.

[31] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. 22–32.