Wikiformer: Pre-training with Structured Information of Wikipedia for Ad-Hoc Retrieval

Weihang Su[1]*, Qingyao Ai[2]†, Xiangsheng Li[2], Jia Chen[2], Yiqun Liu[2], Xiaolong Wu[3], Shengluan Hou[3]

[1]Quan Cheng Laboratory & DCST, Tsinghua University & Zhongguancun Laboratory, Beijing, China
[2]DCST, Tsinghua University & Zhongguancun Laboratory, Beijing, China
[3]Huawei Poisson Lab

Abstract

With the development of deep learning and natural language processing techniques, pre-trained language models have been widely used to solve information retrieval (IR) problems. Benefiting from the pre-training and fine-tuning paradigm, these models achieve state-of-the-art performance. In previous works, plain texts in Wikipedia have been widely used in the pre-training stage. However, the rich structured information in Wikipedia, such as the titles, abstracts, hierarchical heading (multi-level title) structure, relationship between articles, references, hyperlink structures, and the writing organizations, has not been fully explored. In this paper, we devise four pre-training objectives tailored for IR tasks based on the structured knowledge of Wikipedia. Compared to existing pre-training methods, our approach can better capture the semantic knowledge in the training corpus by leveraging the human-edited structured data from Wikipedia. Experimental results on multiple IR benchmark datasets show the superior performance of our model in both zero-shot and fine-tuning settings compared to existing strong retrieval baselines. Besides, experimental results in biomedical and legal domains demonstrate that our approach achieves better performance in vertical domains compared to previous models, especially in scenarios where long text similarity matching is needed. The code is available at https://github.com/oneal2000/Wikiformer.

Introduction

Pre-trained Language Models (PLMs) have achieved great success in the field of Natural Language Processing (NLP)(Devlin et al. 2018; Vaswani et al. 2017; Yang et al. 2019; Liu et al. 2019; Yasunaga, Leskovec, and Liang 2022). These models are firstly pre-trained on a large-scale unlabeled text corpus and then fine-tuned on certain downstream tasks. The pre-training and fine-tuning paradigm have achieved state-of-the-art performance in many downstream NLP tasks. Recently, it has also attracted the attention of the Information Retrieval (IR) community. Besides directly applying PLMs to solve downstream IR tasks(Nogueira and Cho 2019), IR researchers have also developed several pre-training methods tailored for IR tasks, especially ad-hoc search(Ma et al. 2021a,d,b; Chang et al. 2020). These studies have shown promising results in conducting IR-specific pre-trained models for downstream tasks.

As one of the largest online knowledge bases, Wikipedia has been widely used as the pre-training corpus. In previous works, IR researchers have devised several pre-training tasks by leveraging the rich textual contents in Wikipedia. For example, PROP(Ma et al. 2021a) utilizes pure texts in Wikipedia, while HARP(Ma et al. 2021d) utilizes hyperlinks and anchor texts in the web pages. However there's more rich knowledge brought by the structured information of Wikipedia, which, to the best of our knowledge, has not been exploited in existing studies. For example, the abstract section of Wikipedia is the summarization of an article. When the user's query is the title of an article, the abstract section is more likely to match the user's information needs compared to other sections within the same article. In addition, every article on Wikipedia has a hierarchical heading (multi-level title) structure, the subtitle is always the representative words or summarization of the corresponding section. Besides, different subsections of the same section share similar ideas. The relationship between different articles also contains rich information, e.g., the See Also section links one article to other articles that contain additional or similar information. Whether this structured knowledge could benefit the pre-trained models for IR remains mostly unknown.

To better incorporate the knowledge of Wikipedia into the pre-training stage, we propose a framework named Wikiformer that fully utilizes the structured information of Wikipedia in the pre-training stage. Wikiformer mainly includes four pre-training tasks: 1) Simulated Re-ranking (SRR), 2) Representative Words Identification (RWI), 3) Abstract Texts Identification (ATI), and 4) Long Texts Matching (LTM). These tasks use the title, subtitles, abstract, hyperlinks, and heading hierarchies to construct pseudo query-document pairs for the pre-training of the retrieval model. Each of them captures the needs of retrieval and ranking in different granularities from different angles. To evaluate the effectiveness of the above pre-training tasks, we test the performance of our model on several IR benchmarks in zero-shot and fine-tuning settings. In the zero-shot setting, no supervised data is used for fine-tuning. Since the fine-tuning process gradually updates the parameters of PLMs, zero-shot performance is a more direct metric to eval-

*First Author: swh22@mails.tsinghua.edu.cn
†Corresponding Author: aiqy@tsinghua.edu.cn

uate the effectiveness of pre-training methods. The experimental results show that Wikiformer can significantly outperform traditional methods, state-of-the-art neural ranking models, and existing pre-trained models for IR in multiple domains with or without human-annotated data.

In summary, the contributions of our work are three folds:

- We propose a novel pre-training framework, i.e., Wikiformer, that makes full use of the structured knowledge of Wikipedia.
- We propose four learning objectives based on pseudo query-document pair sampling during pre-training. Tailored for IR tasks such as retrieval and document reranking, these objectives can better help the model analyze the relevance between queries and documents.
- We evaluate Wikiformer on multiple IR benchmark datasets, and the experimental results show that Wikiformer outperforms state-of-the-art methods in both zero-shot and fine-tuning settings in multiple domains.

Related Work

Pre-trained Language Models

Pre-trained Language Models (PLMs) have achieved great success in recent years(Devlin et al. 2018; Vaswani et al. 2017; Yang et al. 2019; Liu et al. 2019; Yasunaga, Leskovec, and Liang 2022). These models are firstly trained on large-scale unlabeled text corpora and then fine-tuned on certain downstream tasks with labeled data. Benefiting from the self-supervised learning on a large-scale pre-training corpus, these models own a powerful ability on contextual text representation.

Among these PLMs, Transformer based models(Vaswani et al. 2017) show great performance in most downstream NLP tasks. One of the remarkable examples is the BERT model(Devlin et al. 2018), a bi-directional Transformer based pre-trained language model. BERT has two self-supervised tasks in the pre-training stage: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Following BERT, researchers redesign and optimize the pre-training tasks of PLMs. For example, Roberta(Liu et al. 2019) uses a dynamic masking strategy and is trained on a larger text corpus. In addition, some researchers explore the integration of structured information into PLMs(Yasunaga, Leskovec, and Liang 2022; Colon-Hernandez et al. 2021; Kaur et al. 2022; Zhang et al. 2019). For example, LinkBERT(Yasunaga, Leskovec, and Liang 2022) replaces the NSP task of BERT with the Document Relation Prediction (DRP) task, which enables the model to learn cross-document knowledge from hyperlinks among web pages. ERNIE(Zhang et al. 2019) utilizes both textual corpora and Knowledge Graphs to train an enhanced PLM.

Pre-training Methods Tailored for IR

Considering the great success that PTMs have achieved in NLP tasks, the IR community begins to devise pre-training methods tailored for ad-hoc retrieval(Ma et al. 2023, 2021a,d,b; Chang et al. 2020; Fan et al. 2021; Guo et al. 2022; Chen et al. 2022; Su et al. 2023a,b). For example, HARP(Ma et al. 2021d) utilizes hyperlinks and anchor texts in the pre-training stage. As the anchor texts are edited by humans, constructing pseudo query-document pairs from them may be more reliable than an algorithm. Webformer(Guo et al. 2022) is a pre-trained language model based on large-scale web pages, HTML tags, and the DOM (Document Object Model) tree structures of web pages. Ma et al.(Ma et al. 2021a) devised a self-supervised learning task Representative Words Prediction (ROP) based on the Query Likelihood model and train the Transformer encoder with a self-supervised contrastive learning strategy. From another angle, ARES(Chen et al. 2022) propose several pre-training objectives based on Axiomatic Regularization. Experimental results on several IR benchmarks show that ARES, PROP, Webformer, and HARP perform significantly better than traditional methods such as BM25 after fine-tuning. Also, some researchers explore incorporating structure information for entity retrieval(Gerritse, Hasibi, and de Vries 2020; Nikolaev and Kotov 2020; Chatterjee and Dietz 2022; Gerritse, Hasibi, and de Vries 2022).

Different from the above approaches, we propose four new pre-training objectives using the titles, abstracts, hierarchical heading (multi-level title) structure, relationship between articles, references, hyperlink structures, and the writing organizations of Wikipedia to leverage the wisdom of crowds brought by Wikipedia editors. Compared to previous work, Wikiformer captures more internal relationships between the paragraph structure in Wikipedia web pages, which helps it better model relevance matching.

Methodology

In this section, we introduce the details of the pre-training tasks of our proposed model Wikiformer, including Simulated Re-ranking (SRR), Representative Words Identification (RWI), Abstract Texts Identification (ATI), and Long Texts Matching (LTM) tasks.

Simulated Re-ranking (SRR)

The SRR task is inspired by an important IR problem: document re-ranking. In general, the goal of the document re-ranking task is to sort a series of documents that are highly related to the query, and then select the ones that are most related to the query. According to the characteristics of this task, we aim to design a self-supervised learning task to select the most relevant document from a series of documents with similar contents. In the SRR task, we make full use of the hierarchical heading (multi-level title) structure of Wikipedia to achieve the above objective. Every article on Wikipedia is organized by the hierarchical heading (multi-level title) structure, the subtitle corresponding to a certain section tends to be the representative words or summarization of the text. Besides, different subsections of the same section share similar semantics. As a result, through this structure, we can obtain a series of texts that are highly similar but slightly different in content and generate the query through the multi-level titles as shown in Figure 1.

To be specific, we modeled each Wikipedia article into a

Figure 1: Pseudo query-document pairs generated from the tree structure of a Wikipedia article, where q is the query, d^+ is the positive document and d^- are negative documents.

Figure 2: The contrastive sampling strategy of the RWI task, where D is the document, $q+$ is the positive query and q^- is the negative query.

tree structure namely Wiki Structure Tree (WST) based on the hierarchical heading structure. It can be defined as:

$WST =< D, R >$, where D is a finite set containing n nodes, and R is the root node of WST. Each node in D consists of two parts: the subtitle and its corresponding content. The root node R contains the main title and the abstract of this article. Starting from the root node R, recursively take all the corresponding lower-level sections as its child nodes until every section in this article is added to the WST.

After building WST, we use a contrastive sampling strategy to construct pseudo query-document pairs based on the tree. For a non-leaf node F in the WST, we add all its child nodes to the set S. A node d_i is randomly selected from S. Traversing from the root node to node d_i, all the titles on the path are put together to form a query q. This process is shown in Figure 1. The content of the node d_i is defined as d^+, and the content of the other nodes in S is defined as d^-. We use a Transformer based PLM to compute the relevance score of a pseudo query-document pair:

$$Input = [CLS]query[SEP]document[SEP] \quad (1)$$

$$s(query, document) = MLP(Transformer(Input)) \quad (2)$$

where $Transformer(Input)$ is the vector representation of the "[CLS]" token. $MLP(\cdot)$ is a multi-layer perceptron that projects the [CLS] vector to a relevance score s. For the loss function, we use the Softmax Cross Entropy Loss(Cao et al. 2007; Ai et al. 2018; Gao, Dai, and Callan 2021) to optimize the Transformer based model, which is defined as:

$$\mathcal{L}_{SRR} = -\log \frac{exp(s(q, d^+))}{exp(s(q, d^+)) + \sum_{d \in S} exp(s(q, d))} \quad (3)$$

where q, and d^+ are defined above and S is the set of all negative passages generated from WST.

Representative Words Identification (RWI)

RWI task is inspired by an IR axiom which assumes that the user's query is the representative words extracted from the relevant documents. According to the Wikipedia structure, we regard the subtitle of each section as representative words, and then we sample pseudo query-document pair via a simple strategy based on the hierarchical heading (multi-level title) structure, as shown in Figure 2.

Specifically, pseudo query-document pairs are organized as follows: for each Wikipedia article, we first model it as the WST structure. Then we add all nodes of WST except the root node to the set S. A node d_i is randomly selected from S, and we define the depth of this node in WST as n. Traversing from the root node to node d_i, all the titles on the path are put together to form a query q^+. The content of the node d_i is defined as D. For the negative queries, we randomly select $n-1$ nodes from S, and concatenate the

main title and subtitles of the selected nodes to define it as q^-. The relevance score is defined in Equation2. The loss function of the RWI task is defined as:

$$\mathcal{L}_{RWI} = -\log \frac{exp(s(q^+, D))}{exp(s(q^+, D)) + \sum_{q \in S} exp(s(q, D))} \tag{4}$$

where q^+ is the title, D is the content of that article, and S is the set of all negative queries generated from that article.

In this task, although both positive and negative queries contain the subtitles of the document, the positive query is more representative compared to the negative query. The model gives higher scores to the positive query through contrastive learning, so that the model can recognize the representative words in the text, and assign higher weights to these words if they are matched to the query. Therefore, through the RWI task, the model can learn how to identify the keywords in the text, which further leads to a better performance in the IR downstream task.

Abstract Texts Identification (ATI)

In the ATI task, we utilize the abstract and the inner structure of Wikipedia. The abstract (the first section) of Wikipedia is regarded as the summarization of the whole article. Compared with other sections of the same article, the abstract is more likely to meet the user's information needs when the query is the title. Therefore, we extract the title from the Wikipedia article as the query (denoted as q). Then the abstract of the same article is regarded as a positive document (denoted as d^+). For the negative ones, we use the other sections of the same article (denoted as d^-). The relevance score of a pseudo query-document pair is defined in Equation 2. The loss function of the ATI task is defined as:

$$\mathcal{L}_{ATI} = -\log \frac{exp(s(q, d^+))}{exp(s(q, d^+) + \sum_{d \in S} exp(s(q, d))} \tag{5}$$

where q is the title of the article, d^+ is the abstract of the article, and S is the set of all negative documents generated from that article.

Long Texts Matching (LTM)

After pre-training with RWI, ATI, and SRR tasks, Wikiformer acquires the ability to measure the relevance between a short text (query) and a long text. This can help the model better handle the vast majority of ad-hoc retrieval tasks. However, there are also scenarios involving "long queries", such as legal case retrieval and document-to-document search. In these scenarios, the model is required to match the relevance between two long texts. Fortunately, with the structured information of Wikipedia, especially hyperlinks, we can build a series of informative pseudo long query-document pairs. To be specific, we utilize the See Also section of Wikipedia which consists of hyperlinks that link to the other articles related to or comparable to this article. The See Also section is mainly written manually, based on the judgment and common sense of the authors and editors. Thus, we can obtain a series of reliable web pages that are highly related to the content of this page.

To this end, we design the Long Texts Matching (LTM) task to encourage the Wikiformer to learn the relevance matching ability between two long documents. Initially, we transformed the complete Wikipedia corpus into a graph structure by leveraging the interconnections provided by the 'See Also' links. This graph is designated as the See Also Graph (SAG). Each hyperlink in the See Also section can be formally represented as (v_i, v_j), which means that v_j appears in the See Also section of v_i. Consequently, SAG can be defined as a *directed graph*: $SAG = (V, E)$, where E is the above-mentioned set of ordered pairs (v_i, v_j) and V is a set of Wikipedia articles. The order of an edge indicates the direction of hyperlinks. After building SAG, we use a contrastive sampling strategy based on the graph. For each node in SAG, we define its content as query D and define all its adjacent nodes as positive documents d^+. We randomly select other documents as d^-. The relevance score of a pseudo query-document pair is defined in Equation 2. The loss function of the LTM task is defined as:

$$\mathcal{L}_{LTM} = -\log \frac{exp(s(D, d^+))}{exp(s(D, d^+)) + \sum_{d \in S} exp(s(D, d))} \tag{6}$$

where d^+ is the adjacent articles, D is the content of the original article, and S is the set of all negative articles.

Final Training Objective

We add the loss of the proposed four tasks together as the overall loss of the model:

$$\mathcal{L}_{final} = \mathcal{L}_{SRR} + \mathcal{L}_{RWI} + \mathcal{L}_{ATI} + \mathcal{L}_{LTM} \tag{7}$$

Experiments

Dataset Description

For the pre-training dataset, we use the English Wikipedia (version 20220101). For the downstream datasets, we evaluate the performance of Wikiformer on five IR benchmarks. The basic statistics are shown in Table 1. MS MARCO Document Re-ranking (Nguyen et al. 2016) is a large-scale ad-hoc retrieval dataset with 0.37M queries and 3.2M documents. TREC DL 2019 (Craswell et al. 2020) shares the same document collection with MS MARCO but collects finer-grained human labels for 43 queries in the test set. TREC Covid Round2 (Roberts et al. 2021) is an ad-hoc retrieval dataset consisting of biomedical articles. It contains the May 1, 2020 version of the CORD-19 (Wang et al. 2020) document set and 35 queries written by biomedical professionals. LeCaRD (Ma et al. 2021c) is a legal case retrieval dataset, consisting of 107 query cases and 10700 candidate cases. The queries in the LeCaRD dataset are the factual description part of a legal case, while the candidate documents are complete legal cases. CAIL-LCR(Ma 2022) is a case retrieval dataset (document-to-document search) provided by CAIL 2022 consisting of 130 query cases and 100 candidate cases for each query case.

Dataset	Genre	#Queries	#Documents
MS MARCO	web pages	0.37M	3.2M
TREC DL 2019	web pages	43	3.2M
TREC Covid	biomedical	35	59,851
LeCaRD	legal	107	10,700
CAIL-LCR	legal	130	13,000

Table 1: Basic statistics of our benchmark datasets

Baselines for Comparison

We consider three types of IR baselines for comparison, including traditional IR methods, Neural IR models, and pre-trained language models:

Query Likelihood (Zhai 2008) is a language model based on Dirichlet smoothing.

BM25 (Robertson, Zaragoza et al. 2009) is a highly effective retrieval model based on lexical matching.

KNRM(Xiong et al. 2017) is an Interactive-based Neural Ranking Model that uses kernel-pooling to provide matching signals for each query-document pair.

Conv-KNRM(Dai et al. 2018) is a Convolutional Kernel-based Neural Ranking Model that fuses the contextual information of the surrounding words for relevance matching.

BERT (Devlin et al. 2018) is a bi-directional Transformer based Pre-trained Language Model that has a powerful ability on contextual text representations.

PROP_MS (Ma et al. 2021a) adopts the Representative Words Prediction (ROP) task to learn relevance matching from the pseudo query-document pairs. It is pre-trained on MS MARCO.[1]

PROP_WIKI (Ma et al. 2021a) adopts the same pre-training task as PROP_MS. The only difference is that PROP_WIKI is pre-trained on Wikipedia.

HARP (Ma et al. 2021d) utilizes the hyperlinks and anchor texts to generate pseudo query-document pairs and achieves state-of-the-art performance on ad-hoc retrieval.

ARES (Chen et al. 2022) is a pre-trained language model with Axiomatic Regularization for ad hoc Search.

Webformer (Guo et al. 2022) is a pre-trained language model based on large-scale web pages and their DOM (Document Object Model) tree structures.

Implementation Details

For the implementation of KNRM and Conv-KNRM, we use the OpenMatch[2] toolkit, and the 300d GloVe (Pennington, Socher, and Manning 2014) vectors are used to initialize the word embeddings. For the implementation of BM25 and QL, we use the pyserini toolkit[3]. For the hyperparameter of BM25, we set $k1 = 3.8$ and $b = 0.87$[4]. **Note that in our experiments, we use the scores of the BM25 and QL models to re-rank the candidate documents, rather than re-ranking the whole corpus.** For the implementation of BERT, we use the Pytorch version BERT-base released by Google[5]. For the implementation of ARES, PROP_MS, and PROP_WIKI, we directly use the checkpoints released by the original paper. Since the original paper of Webformer and HARP did not release any checkpoints, we reproduce them on the same dataset based on their code and the details provided in their paper.

To facilitate comparison with previous baselines, we adopted the same architecture as BERT-base. This aligns with the settings of previous works such as ARES, HARP, PROP, B-PROP, and Webformer. To save computational resources during training, we initialized our model with BERT-base, following the same setting as previous works such as ARES and HARP. We use the AdamW optimizer with a learning rate of 1e-5 in the first 50k steps and 5e-6 in the following steps. We set the warm-up ratio to 0.1. In the RWI, ATI, and SRR tasks, we set the maximum length of the query as 30 and the maximum length of the documents as 480. In the LTM task, we set the maximum length of both documents as 255. We trained our model on four Nvidia GeForce RTX 3090 GPUs for 60 hours. After training for 50k steps, we save the checkpoint every 5k steps and evaluate the zero-shot performance of each checkpoint on a subset of the MS MARCO training set which has no overlap with our test set. We select the best zero-shot performance checkpoint as the final model.

Evaluation Methodology

For the two large-scale datasets MS MARCO and TREC DL 2019, we use Mean Reciprocal Rank at 10 and 100 (MRR@10 and MRR@100) for MS MARCO and normalized discounted cumulative gain at 10 and 100 (nDCG@10 and nDCG@100) for TREC DL 2019 as the evaluation metrics. For TREC Covid, we follow the setting of OpenMatch which re-ranks the top 60 candidates provided by the BM25-fusion method. We use precision at rank 5 (P@50) and nDCG@10 as the evaluation metrics for TREC Covid. For LeCaRD and CAIL-LCR datasets, we re-rank the candidate documents provided by the original dataset and use nDCG@5 and nDCG@15 as the evaluation metrics.

For the significance test, we adopt Fisher's randomization test (Fisher 1936; Cohen 1995; Box et al. 1978) which is recommended for IR evaluation by previous work (Smucker, Allan, and Carterette 2007).

Experimental Results

Zero-shot Performance Zero-shot performance is the performance of the model without any supervised data for fine-tuning. Thus, it directly reflects the effectiveness of the pre-training tasks. The experimental results are shown in Table 2. We can see that Wikiformer outperforms all baselines on all evaluation metrics which shows the superiority of Wikiformer in the zero-shot setting. Based on the results, we also have the following findings:

[1]As PROP and B-PROP have similar performance, and B-PROP does not have a publicly available model checkpoint, therefore we only choose PROP as the baseline instead of selecting both of them.

[2]https://github.com/thunlp/OpenMatch

[3]https://github.com/castorini/pyserini

[4]This is the best hyperparameter we got after parameter searching.

[5]https://github.com/google-research/bert

		Zero-shot				Fine-tuned			
		MS MARCO		TREC DL 2019		MS MARCO		TREC DL 2019	
Model Type	Model Name	M@10	M@100	N@10	N@100	M@10	M@100	N@10	N@100
Traditional Models	BM25	0.2656*	0.2767*	0.5315*	0.4996*	0.2656*	0.2767*	0.5315*	0.4996*
	QL	0.2143*	0.2268*	0.5234*	0.4983*	0.2143*	0.2268*	0.5234*	0.4983*
Neural IR Models	KNRM	NA	NA	NA	NA	0.1526*	0.1685*	0.3071*	0.4591*
	Conv-KNRM	NA	NA	NA	NA	0.1554*	0.1792*	0.3112*	0.4762*
Pre-trained Models	BERT	0.1684*	0.1811*	0.3407**	0.4316*	0.3826*	0.3881*	0.6540	0.5325*
	PROP_WIKI	0.2205*	0.2321*	0.4712*	0.4709*	0.3866*	0.3922*	0.6399*	0.5311*
	PROP_MS	0.2585*	0.2696*	0.5203*	0.4810*	0.3930*	0.3980*	0.6425*	0.5318*
	Webformer	0.1664*	0.1756*	0.3758*	0.4550*	0.3984*	0.4036*	0.6479*	0.5335
	HARP	0.2372*	0.2465*	0.5244*	0.4721*	0.3961*	0.4012*	0.6562	0.5337
	ARES	0.2736*	0.2851*	0.5736*	0.4752*	0.3995*	0.4041*	0.6505*	0.5353
Our Approach	Wikiformer	**0.2844**	**0.2911**	**0.5907**	**0.5143**	**0.4085**	**0.4136**	**0.6587**	**0.5392**

Table 2: The experimental results of Wikiformer and other baselines on three datasets in the zero-shot and fine-tuning setting. "*" denotes the result is significantly worse than Wikiformer with $p < 0.05$ level. The best results are in bold. "N" stands for nDCG, and "M" stands for MRR. The zero-shot performance of both KNRM and Conv-KNRM methods is the same as randomized ranking. Therefore, their zero-shot performance is not shown in the table.

Pre-trained models tailored for IR such as PROP, ARES, and Wikiformer perform significantly better than BERT in zero-shot settings. This shows the effectiveness of the pre-training tasks tailored for IR and that these models have indeed learned useful knowledge for relevance matching. Wikiformer performs the best among all the baselines in both benchmarks in zero-shot settings. Since the model architecture and parameter size of Wikiformer are the same as the other pre-trained models, this shows the effectiveness of our pre-training method. Besides, Wikiformer, Webformer, and PROP-Wiki are all pre-trained on the Wikipedia corpus. The superior performance of Wikiformer shows that it has made better use of Wikipedia and learned the rich knowledge that is helpful to solve IR problems through structured information on Wikipedia.

Fine-tuned Performance Table2 reports the performance of Wikiformer and other baselines after fine-tuning. Through the experimental results, we have the following findings: (1) Although the performance of most pre-trained language models (PLMs) is inferior to traditional methods like BM25 and QL in the zero-shot setting, they surpass BM25 and QL significantly after fine-tuning. However, even after fine-tuning, Neural IR Models still underperform BM25 and QL. (2) On the MS MARCO dataset, IR PLMs consistently outperform BERT under the fine-tuning setting. This indicates that the knowledge acquired by IR PLMs during the pre-training stage remains valuable even after fine-tuning. HARP and Webformer, due to the incorporation of external knowledge such as hyperlinks, DOM Tree, and HTML tags, exhibit better performance than PROP-WIKI and PROP-MS. (3) Wikiformer significantly outperforms other baselines on both datasets. Note that the model structure and fine-tuning dataset for Wikiformer are the same as other baselines. Therefore, these experimental results indicate that Wikiformer has acquired more information retrieval knowledge during the pre-training stage compared to other baselines. This demonstrates the value of our pre-training task.

	TREC Covid rnd2	
	Zero-shot N@10	Fine-tuned N@10
QL	0.4683*	0.4683*
BM25	0.4792*	0.4792*
BERT	0.4018*	0.5580*
PROP_MS	0.4994*	0.5944*
PROP_WIKI	0.4137*	0.6104*
Webformer	0.3845*	0.6032*
HARP	0.4027*	0.5832*
ARES	0.4993*	0.5969*
Wikiformer	**0.5449**	**0.6197**

Table 3: The experimental results of Wikiformer and other baselines on the TREC Covid rnd2 dataset. The best results are in bold. "*" denotes the result is significantly worse than Wikiformer with $p < 0.05$ level. "N" stands for nDCG.

Performance on Vertical Domains We conducted experiments on the legal domain dataset LeCaRD and CAIL-SCR as well as the biomedical domain dataset TREC Covid to explore the performance of Wikiformer in vertical domains. The experimental results are presented in Tables4 and Table3. The experimental results indicate that Wikiformer outperforms previous pre-trained models significantly in both the legal and biomedical domains. This suggests that Wikiformer possesses a domain-specific adaptability and effectiveness that allows it to excel in information retrieval tasks within these specialized fields. Its superior highlights the potential of utilizing Wikiformer for improving search and retrieval tasks across diverse domains.

	LeCaRD		CAIL-LCR	
	N@5	**N@15**	**N@5**	**N@15**
BM25	0.6843*	0.7303*	0.7105*	0.7490*
QL	0.6906*	0.7411*	0.7389*	0.7756*
BERT	0.7553*	0.7966*	0.7993*	0.8085
Wikiformer	**0.7722**	**0.8073**	**0.8095**	**0.8134**

Table 4: The experimental results of Wikiformer and other baselines on LeCaRD and CAIL-LCR in fine-tuning setting. The best results are in bold. "*" denotes the result is significantly worse than Wikiformer with $p < 0.05$ level. N@5 and N@15 respectively represent nDCG@5 and nDCG@15.

	MS MARCO		LeCaRD
	MRR@10	**MRR@100**	**nDCG@5**
w/o **SRR**	0.2334*	0.2441*	0.7613*
w/o **RWI**	0.2596*	0.2712*	0.7685*
w/o **ATI**	0.2641*	0.2751*	0.7627*
w/o **LTM**	0.2726*	0.2835*	0.7574*
w/o **Pre-training**	0.1684	0.1811	0.7553
All Four Tasks	**0.2844***	**0.2911***	**0.7722***

Table 5: Ablation study results. The best results are in bold and the worst results are underlined."*" denotes the performance is significantly better than the backbone model (BERT) with $p < 0.01$ level.

Long Text Matching Performance The performance of Wikiformer and other baselines on LeCaRD and CAIL-LCR are reported in Table 4. LeCaRD and CAIL-LCR are Chinese legal case retrieval tasks that have relatively long queries and candidate documents. Thus, experiments on these datasets can evaluate the long text-matching performance of Wikiformer and the baselines. Since there is no Chinese-centric pre-trained model tailored for IR so far, we only use traditional methods and a Chinese version of the BERT model (Cui et al. 2021) as baselines.

The experimental results show that Wikiformer achieves better performance than traditional statistic methods BM25 and QL but also pre-trained language model BERT in long text-matching tasks. These experimental results highlight the potential of Wikiformer in effectively evaluating long-text similarity and also underscore the effectiveness of the proposed Long Text Matching (LTM) task.

Impact of the Training Data Size

To investigate whether a larger training dataset enhances the performance of the pre-training phase, we evaluate the performance of Wikiformer on different sizes of training data varying from 100 to 1,000,000 pseudo query-document pairs. As shown in Figure 3, Wikiformer surpasses the Query Likelihood model by pre-training with only 100 pseudo query-document pairs in the SRR task. This experimental result shows the effectiveness of our pre-training task and our proposed pseudo query-document pair sampling strategy.

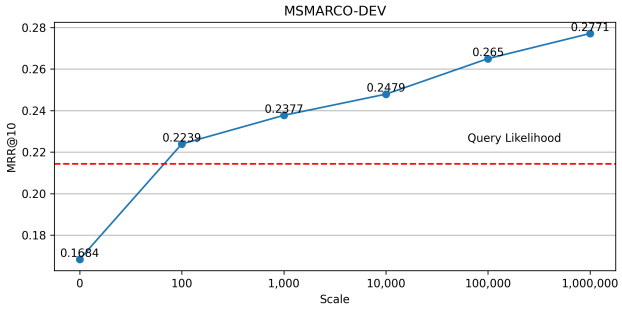

Figure 3: The performance of Wikiformer at different sizes of pre-training data sampled from the SRR task. The red dotted line shows the performance of Query Likelihood.

Ablation Study

To further analyze the effectiveness of each pre-training task, we conduct ablation experiments on MS MARCO (zero-shot) and LeCaRD (fine-tuned). The experimental results in table 5 show that removing any pre-training tasks will lead to a drop in performance, indicating the effectiveness of each pre-training task on downstream IR tasks. On MS MARCO, among the four tasks, removing the SRR task leads to the largest performance degradation, which reveals that the hierarchical heading structure and the writing organization of Wikipedia contain valuable knowledge for ad-hoc retrieval which helps Wikiformer better at handling relevance matching. On LeCaRD, removing the LTM task leads to the largest performance degradation, which reveals that the LTM task is critical for improving the model's ability on long text-matching tasks.

Conclusions

In this paper, we propose Wikiformer, a pre-trained language model tailored for IR that achieves state-of-the-art performance. We propose several pseudo query-document pair sampling strategies based on the structured information on Wikipedia to leverage the wisdom of crowds brought by Wikipedia editors. Extensive experimental results and case studies verify the effectiveness of our pre-training methods. Results of the ablation study have also implied the effectiveness of all pre-training tasks.

Acknowledgements

This work is supported by Quan Cheng Laboratory (Grant No. QCLZD202301), the Natural Science Foundation of China (Grant No. 62002194), and Huawei Poisson Lab.

References

Ai, Q.; Bi, K.; Guo, J.; and Croft, W. B. 2018. Learning a deep listwise context model for ranking refinement. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, 135–144.

Box, G. E.; Hunter, W. H.; Hunter, S.; et al. 1978. *Statistics for experimenters*, volume 664. John Wiley and sons New York.

Cao, Z.; Qin, T.; Liu, T.-Y.; Tsai, M.-F.; and Li, H. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, 129–136.

Chang, W.-C.; Yu, F. X.; Chang, Y.-W.; Yang, Y.; and Kumar, S. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932*.

Chatterjee, S.; and Dietz, L. 2022. BERT-ER: Query-specific BERT Entity Representations for Entity Ranking. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1466–1477.

Chen, J.; Liu, Y.; Fang, Y.; Mao, J.; Fang, H.; Yang, S.; Xie, X.; Zhang, M.; and Ma, S. 2022. Axiomatically Regularized Pre-training for Ad hoc Search.

Cohen, P. R. 1995. *Empirical methods for artificial intelligence*, volume 139. MIT press Cambridge, MA.

Colon-Hernandez, P.; Havasi, C.; Alonso, J.; Huggins, M.; and Breazeal, C. 2021. Combining pre-trained language models and structured knowledge. *arXiv preprint arXiv:2101.12294*.

Craswell, N.; Mitra, B.; Yilmaz, E.; Campos, D.; and Voorhees, E. M. 2020. Overview of the TREC 2019 deep learning track. *arXiv preprint arXiv:2003.07820*.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; and Yang, Z. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29: 3504–3514.

Dai, Z.; Xiong, C.; Callan, J.; and Liu, Z. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 126–134.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fan, Y.; Xie, X.; Cai, Y.; Chen, J.; Ma, X.; Li, X.; Zhang, R.; Guo, J.; and Liu, Y. 2021. Pre-training Methods in Information Retrieval. *arXiv preprint arXiv:2111.13853*.

Fisher, R. A. 1936. Design of experiments. *British Medical Journal*, 1(3923): 554.

Gao, L.; Dai, Z.; and Callan, J. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *European Conference on Information Retrieval*, 280–286. Springer.

Gerritse, E. J.; Hasibi, F.; and de Vries, A. P. 2020. Graph-embedding empowered entity retrieval. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*, 97–110. Springer.

Gerritse, E. J.; Hasibi, F.; and de Vries, A. P. 2022. Entity-aware Transformers for Entity Search. *arXiv preprint arXiv:2205.00820*.

Guo, Y.; Ma, Z.; Mao, J.; Qian, H.; Zhang, X.; Jiang, H.; Cao, Z.; and Dou, Z. 2022. Webformer: Pre-training with Web Pages for Information Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1502–1512.

Kaur, J. N.; Bhatia, S.; Aggarwal, M.; Bansal, R.; and Krishnamurthy, B. 2022. LM-CORE: Language models with contextually relevant external knowledge. *arXiv preprint arXiv:2208.06458*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ma, X.; Guo, J.; Zhang, R.; Fan, Y.; Ji, X.; and Cheng, X. 2021a. Prop: Pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 283–291.

Ma, X.; Guo, J.; Zhang, R.; Fan, Y.; Li, Y.; and Cheng, X. 2021b. B-PROP: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1513–1522.

Ma, Y. 2022. CAIL-SCR. https://github.com/china-ai-law-challenge/CAIL2022/tree/main/lajs.

Ma, Y.; Shao, Y.; Wu, Y.; Liu, Y.; Zhang, R.; Zhang, M.; and Ma, S. 2021c. LeCaRD: a legal case retrieval dataset for Chinese law system. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2342–2348.

Ma, Y.; Wu, Y.; Su, W.; Ai, Q.; and Liu, Y. 2023. CaseEncoder: A Knowledge-enhanced Pre-trained Model for Legal Case Encoding. *arXiv preprint arXiv:2305.05393*.

Ma, Z.; Dou, Z.; Xu, W.; Zhang, X.; Jiang, H.; Cao, Z.; and Wen, J.-R. 2021d. Pre-training for Ad-hoc Retrieval: Hyperlink is Also You Need. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1212–1221.

Nguyen, T.; Rosenberg, M.; Song, X.; Gao, J.; Tiwary, S.; Majumder, R.; and Deng, L. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Nikolaev, F.; and Kotov, A. 2020. Joint word and entity embeddings for entity retrieval from a knowledge graph. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*, 141–155. Springer.

Nogueira, R.; and Cho, K. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Roberts, K.; Alam, T.; Bedrick, S.; Demner-Fushman, D.; Lo, K.; Soboroff, I.; Voorhees, E.; Wang, L. L.; and Hersh, W. R. 2021. Searching for scientific evidence in a pandemic: An overview of TREC-COVID. *Journal of Biomedical Informatics*, 121: 103865.

Robertson, S.; Zaragoza, H.; et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4): 333–389.

Smucker, M. D.; Allan, J.; and Carterette, B. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 623–632.

Su, W.; Ai, Q.; Wu, Y.; Ma, Y.; Li, H.; and Liu, Y. 2023a. Caseformer: Pre-training for Legal Case Retrieval. *arXiv preprint arXiv:2311.00333*.

Su, W.; Li, X.; Liu, Y.; Zhang, M.; and Ma, S. 2023b. THUIR2 at NTCIR-16 Session Search (SS) Task. *arXiv preprint arXiv:2307.00250*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, L. L.; Lo, K.; Chandrasekhar, Y.; Reas, R.; Yang, J.; Eide, D.; Funk, K.; Kinney, R.; Liu, Z.; Merrill, W.; et al. 2020. Cord-19: The covid-19 open research dataset. *ArXiv*.

Xiong, C.; Dai, Z.; Callan, J.; Liu, Z.; and Power, R. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, 55–64.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Yasunaga, M.; Leskovec, J.; and Liang, P. 2022. LinkBERT: Pretraining Language Models with Document Links. *arXiv preprint arXiv:2203.15827*.

Zhai, C. 2008. Statistical language models for information retrieval. *Synthesis lectures on human language technologies*, 1(1): 1–141.

Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; and Liu, Q. 2019. ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.