PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji and Xueqi Cheng
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{maxinyu17g,guojiafeng,zhangruqing,fanyixing,jixiang19s,cxq}@ict.ac.cn

ABSTRACT

Recently pre-trained language representation models such as BERT have shown great success when fine-tuned on downstream tasks including information retrieval (IR). However, pre-training objectives tailored for ad-hoc retrieval have not been well explored. In this paper, we propose Pre-training with Representative wOrds Prediction (PROP) for ad-hoc retrieval. PROP is inspired by the classical statistical language model for IR, specifically the query likelihood model, which assumes that the query is generated as the piece of text representative of the "ideal" document. Based on this idea, we construct the representative words prediction (ROP) task for pre-training. Given an input document, we sample a pair of word sets according to the document language model, where the set with higher likelihood is deemed as more representative of the document. We then pre-train the Transformer model to predict the pairwise preference between the two word sets, jointly with the Masked Language Model (MLM) objective. By further fine-tuning on a variety of representative downstream ad-hoc retrieval tasks, PROP achieves significant improvements over baselines without pre-training or with other pre-training methods. We also show that PROP can achieve exciting performance under both the zero- and low-resource IR settings. The code is available at https://github.com/Albert-Ma/PROP.

CCS CONCEPTS

• **Information systems → Information retrieval**.

KEYWORDS

Pre-training; Statistical Language Model; Ad-hoc Retrieval

ACM Reference Format:
Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji and Xueqi Cheng. 2021. PROP: Pre-training with Representative Words Prediction for Ad-hoc Retrieval. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441777

1 INTRODUCTION

Recent advances have shown that pre-trained language representation models, such as OpenAI GPT [30], BERT [11] and XLNET [41], can capture rich language information from text, and achieve state-of-the-art accuracy in many downstream natural language processing (NLP) tasks including summarization [35], sentiment classification [33], and named entity recognition [32], which usually have limited supervised data. The success of pre-trained models in NLP has also attracted a lot of attention in the IR community. Researchers have explored the popular models, e.g., ELMo [26] and BERT, in the context of ad-hoc document ranking, and showed that they can also largely benefit the search tasks where training data are limited [9, 20, 23, 24, 40].

Despite the exciting performance of pre-trained models on IR tasks, however, pre-training objectives tailored for ad-hoc retrieval have not been well explored. On the one hand, most existing pre-training objectives that come from NLP can be summarized into two folds, i.e., sequence-based and sentence pair-based tasks. Sequence-based pre-training tasks, such as Masked Language Modeling (MLM) [11] and Permuted Language Modeling (PLM) [41], aim to learn contextual representations for a word based on the sequence-level co-occurrence information. Sentence pair-based pre-training tasks, such as Next Sentence Prediction (NSP) [11] and Sentence Order Prediction (SOP) [38], attempt to teach the model to better understand the inter-sentence coherence relationship [29]. On the other hand, IR tasks such as ad-hoc retrieval typically handle short (keyword-based) queries and long (multi-sentence-based) documents. It requires not only understanding the text content of a query and a document, but also modeling the relevance relationship between the two. When we look at those existing pre-training objectives from the IR perspective, we may find that: 1) Sequence-based pre-training tasks could in general contribute to build good contextual representations for the query and the document; 2) The learning objectives of sentence pair-based tasks, however, are quite diverged from the IR requirement, not just due to the input difference (sentence-pair vs. query-document) but also the relation type (coherence vs. relevance). It is generally hypothesized that using a pre-training objective that more closely resembles the downstream task leads to better fine-tuning performance [44]. In this sense, we argue that the power of pre-training has not been fully exploited for ad-hoc retrieval tasks.

Yet there has been little effort to design pre-training objectives towards ad-hoc retrieval. The most related work in this direction focused on passage retrieval in question answering (QA) [5, 18], where three types of pre-training tasks have been proposed including: (1) Inverse Cloze Task (ICT): The query is a sentence randomly

WSDM '21, March 8–12, 2021, Virtual Event, Israel
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441777

drawn from the passage and the document is the rest of sentences; (2) Body First Selection (BFS): The query is a random sentence in the first section of a Wikipedia page, and the document is a random passage from the same page; and (3) Wiki Link Prediction (WLP): The query is a random sentence in the first section of a Wikipedia page, and the document is a passage from another page where there is a hyperlink link to the page of the query. As we can see, these tasks attempt to resemble the relevance relationship between natural language questions and answer passages. Some tasks even depend on certain special document structure, e.g., hyperlink. When applying pre-trained models based on these tasks to ad-hoc retrieval, marginal benefit could be observed on typical benchmark datasets as shown in Section 4.5.

In this paper, therefore, we aim to design a novel pre-training objective tailored for IR which more closely resembles the relevance relationship between query and document in ad-hoc retrieval. The key idea is inspired by the traditional statistical language model for IR, specifically the query likelihood model [27] which was proposed in the last century. The query likelihood model assumes that the query is generated as the piece of text representative of the "ideal" document [19]. Based on the Bayesian theorem, the relevance relationship between query and document could then be approximated by the query likelihood given the document language model under some mild prior assumption. Based on the classical IR theory, we propose the Representative wOrds Prediction (ROP) task for pre-training. Specifically, given an input document, we sample a pair of word sets according to the document language model, which is defined by a popular multinomial unigram language model with Dirichlet prior smoothing. The word set with higher likelihood is deemed as more "representative" of the document. We then pre-train the Transformer model to predict the pairwise preference between the two sets of words, jointly with the Masked Language Model (MLM) objective. The pre-trained model, namely PROP for short, could then be fine-tuned on a variety of downstream ad-hoc retrieval tasks. The key advantage of PROP lies in that it roots in a good theoretical foundation of IR and could be universally trained over large scale text corpus without any special document structure (e.g. hyperlinks) requirement.

We pre-train PROP based on two kinds of large scale text corpus respectively. One is the English Wikipedia which contains tens of millions of well-formed wiki-articles, and the other is the MS MARCO Document Ranking dataset which contains about 4 million Web documents. We then fine-tune PROP on 5 representative downstream ad-hoc retrieval datasets, including Robust04, ClueWeb09-B, Gov2, MQ2007 and MQ2008. The empirical experimental results demonstrated that PROP can achieve significant improvements over baselines without pre-training or with other pre-training methods, and further push forward the state-of-the-art. Large-scale labeled IR datasets are rare and in practice it is often time-consuming to collect sufficient relevance labels over queries. The most common setting is that of zero- or low-resource ad-hoc retrieval. We simulate both settings and show that our model is capable of obtaining state-of-the-art results when fine-tuning with small numbers of supervised pairs. The contributions of this work are listed as follows:

- We propose PROP, a new pre-training objective for ad-hoc retrieval which has a good theoretical IR foundation and could be universally trained over large scale text corpus without any special document structure requirement.
- We evaluate PROP on a variety of downstream ad-hoc retrieval tasks and demonstrate that our model can surpass the state-of-the-art methods.
- We show how good ad-hoc retrieval performance can be achieved across different datasets with very little supervision by fine-tuning the PROP model.

2 BACKGROUND

We first briefly review the classical statistical language model for IR, specifically the query likelihood model, which is the theoretical foundation of our pre-training method. The basic idea of the query likelihood model assumes that the user has a reasonable idea of the terms that are likely to appear in the "ideal" document that can satisfy his/her information need [27].

The query is thus generated as the piece of text representative of the "ideal" document [19].

Such a query-generation idea could then be formulated as a probabilistic model using the Bayesian theorem. Specifically, given a query $Q = q_1...q_m$ and a document $D = w_1...w_n$, we have:

$$P(D|Q) \propto P(Q|\theta_D)P(D), \tag{1}$$

where θ_D is a document language model estimated for every document. The prior probability $P(D)$ is usually assumed to be uniform across all documents and thus can be ignored. Based on this simplification, the estimation of the relevance of a document to a query $P(D|Q)$ could be approximated by the query likelihood $P(Q|\theta_D)$, i.e., the query generation probability, according to the document language model θ_D.

Different methods have been proposed for the document language model θ_D, among which a multinomial unigram language model has been most popular and most successful [42]. Assuming a multinomial language model, one would generate a sequence of words by generating each word independently. In this way, the query likelihood would be

$$\begin{aligned} P(Q|D) &= \prod_{i}^{m} P(q_i|\theta_D) \\ &= \prod_{w \in V} P(w|\theta_D)^{c(w,Q)}, \end{aligned} \tag{2}$$

where V is the corpus vocabulary and $c(w,Q)$ is the count of word w in query Q.

To better estimate the document language model and eliminate zero probabilities for unseen words, many smoothing methods have been proposed to improve the accuracy of the estimated language model. Among all these methods, Dirichlet prior smoothing appears to work the best, especially for keyword queries (non-verbose queries) [43], which is defined as

$$P(w|D) = \frac{c(w,D) + \mu P(w|C)}{|D| + \mu}, \tag{3}$$

where $c(w,D)$ is the count of word w in document D, $|D|$ is the length of document D (i.e., the total word counts), $P(w|C)$ is a background (collection) language model estimated based on word counts in the entire collection and μ is a smoothing parameter.

Algorithm 1 Sampling a Pair of Representative Word Sets

```
Input: Document $D$, Vocabulary $V = \{w_i\}_1^N$, probability of word $w_i$ generated by the document language model with Dirichlet smoothing $P(w_i|D)$, Query likelihood score function $QL(w_i, D)$
// Choose length
$l = Sample(X), x \sim Poisson(\lambda), x = 1, 2, 3...$
$S_1, S_2 = \emptyset, \emptyset$
// Paired Sampling
for $k \leftarrow 1$ to $l$ do
    $S_1 = S_1 \cup Sample(V), w_i \sim P(w_i|D)$
    $S_2 = S_2 \cup Sample(V), w_i \sim P(w_i|D)$
end for
// Higher likelihood deemed as more representative
$S_1\_score = \prod_i^l QL(w_i, D), w_i \in S_1$
$S_2\_score = \prod_i^l QL(w_i, D), w_i \in S_2$
if $S_1\_score > S_2\_score$ then
    Output:$(S_1^+, S_2^-, D)$
else
    Output:$(S_1^-, S_2^+, D)$
end if
```

For more details about statistical language models for IR, we refer reader to these papers [19, 22, 42].

3 PROP

In this section, we present the new pre-training objective PROP which is tailored for ad-hoc retrieval in detail. We also provide some discussions on the differences and connections of PROP with respect to weak supervision methods for IR and existing pre-training objectives respectively.

3.1 Pre-training Methods

Existing work has demonstrated that using a pre-training objective that more closely resembles the downstream task leads to better fine-tuning performance [34, 44]. Given our intended use for ad-hoc retrieval, we aim to introduce a new pre-training task that better resembles the relevance relationship between query and document in IR.

The key idea is inspired by the above query likelihood model which assumes that the query is generated as the piece of text representative of the "ideal" document. Based on this assumption, we construct the representative words prediction (ROP) task for pre-training. Specifically, given an input document, we sample a pair of word sets according to the document language model. Intuitively, each sampled word set could be viewed as a generated pseudo query from the document. In this way, the word set with higher likelihood is deemed as a more "representative" query of the document. We then pre-train the Transformer model to predict the pairwise preference between the two word set, i.e., the ROP objective, jointly with Masked Language Model (MLM) objective. The pre-trained model is named as PROP for short. The detailed pre-training procedures are as follows.

Representative Word Sets Sampling. Given a document, we sample a pair of word sets, each as a generated pseudo query, according to the document language model. To simulate the varied query length in practice [1, 3], we first use a Poisson distribution [7] to sample a positive integer l as the size of the word set, which is defined by

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x}, x = 1, 2...,$$

where λ is a hyper-parameter that indicates the expectation of interval. We then sample a pair of word sets S_1 and S_2 with the same size l in parallel according to the document language model. Specifically, for each word set, l words are sampled from the corpus vocabulary $V = \{w_i\}_1^N$ independently according to the multinomial unigram language model with Dirichlet prior smoothing as defined by Equation (3). The detailed sampling process is shown in Algorithm 1.

Representative Words Prediction (ROP). Given the pair of word sets sampled above, we compute the likelihood of each set according to Equation (2), and the set with the higher likelihood is regarded as more representative for the document. We then pre-train a Transformer model to predict the pairwise preference between the two word sets.

Specially, the word set S and the document D are concatenated as a single input sequence and fed into the Transformer with special delimiting tokens, i.e., $[CLS]$+S+$[SEP]$+D+$[SEP]$. Each word in the concatenated sequence is represented by summing its distributed, segment, and positional embeddings. Then, the hidden state of the special token [CLS], i.e. $\mathbf{H}^{[CLS]}$, is obtained by,

$$\mathbf{H}^{[CLS]} = Transformer_L([CLS] + S + [SEP] + D + [SEP]), \quad (4)$$

where L is a hyper-parameter denoting the number of Transformer layers. Finally, the likelihood $P(S|D)$, which denotes how representative the word set is to the document, is obtained by applying a multi-layer perceptron (MLP) function over the $\mathbf{H}^{[CLS]}$ following previous studies [9, 23, 24].

Now we denote the pair of word sets sampled and the corresponding document as a triple (S_1, S_2, D). Suppose set S_1 has a higher likelihood score than S_2 according to Equation (2), the ROP task can then be formulated by a typical pairwise loss, i.e., hinge loss, for the pre-training.

$$\mathcal{L}_{ROP} = max(0, 1 - P(S_1|D) + s(S_2|D)), \quad (5)$$

Masked Language Modeling (MLM). MLM is firstly proposed by Taylor [36] in the literature, which is a fill-in-the-blank task. MLM first masks out some tokens from the input and then trains the model to predict the masked tokens by the rest of the tokens. As mentioned in the Introduction, the MLM objective could in general contribute to building good contextual representations for the query and the document. Therefore, similar to BERT, PROP also adopts MLM as one of its pre-training objectives besides the pairwise preference prediction objective.

Specifically, the MLM loss $\mathcal{L}_{MLM}$ is defined as:

$$\mathcal{L}_{MLM} = -\sum_{\hat{x} \in m(\mathbf{x})} \log p(\hat{x}|\mathbf{x}_{\backslash m(\mathbf{x})}), \quad (6)$$

where $\mathbf{x}$ denotes the input sentences, $m(x)$ and $\mathbf{x}_{\backslash m(\mathbf{x})}$ denotes the masked words and the rest words from $\mathbf{x}$, respectively.

3.2 Discussions

There might be some confusion between the proposed pre-training model PROP and those weak supervision methods [2, 10, 21] in IR, which also leverage some classical models (e.g., BM25) to train neural ranking models. In fact, there are three major differences between the two. For those weak supervision methods: 1) Both queries and documents are available but relevance labels are missing; 2) The learning objective of weak supervision is the same as the final ranking objective; 3) The weak supervision is typically designed for each specific retrieval task. In contrary, for PROP: 1) Only documents are available while either queries or relevance labels are missing; 2) The PROP objective is not the same as the final ranking objective; 3) The pre-trained PROP model could be fine-tuned on a variety of downstream ranking tasks.

Among the pre-training objectives, the ROP objective in PROP belongs to the category of model-based pre-training objective, where the labels are produced by some automatic model rather than simple MASKs. Similar pre-training objectives in this category include Electra [6] which leverages a generative model to replace masked tokens for pre-training the language model, and PEGASUS [44] which leverages the ROUGE1-F1 score to select top-m sentences for pre-training the abstractive summarization.

4 EXPERIMENTS

In this section, we conduct experiments to verify the effectiveness of PROP on benchmark collections.

4.1 Datasets

We first introduce the two large text corpora for pre-training and five downstream ad-hoc retrieval datasets.

4.1.1 Pre-training Corpus. We use two large document corpora, including English Wikipedia and MS MARCO Document Ranking dataset, to pre-train PROP since (1) They are publicly available and easy to collect; (2) A large collection of documents in these datasets could well support our pre-training method.

- **English Wikipedia** contains tens of millions of documents which has been widely used in many pre-training methods and we download the latest dump[1] and extract the text with a public script[2].
- **MS MARCO Document Ranking dataset** is another large-scale document collection which contains about 4 million available documents. This dataset was used in the TREC Deep Learning Track 2019[3] and 2020[4]. Documents are extracted from real Web documents using the Bing search engine.

By pre-training PROP on English Wikipedia and MS MARCO Document Ranking dataset respectively, we obtain two types of models denoted as $\textbf{PROP}_{\textbf{Wikipedia}}$ and $\textbf{PROP}_{\textbf{MSMARCO}}$.

4.1.2 Downstream Datasets. To verify the effectiveness of PROP, we conduct experiments on 5 representative ad-hoc retrieval datasets.

- **Robust04** consists of 250 queries and 0.5M news articles, whose topics are collected from TREC 2004 Robust Track.

Table 1: Statistics of the ad-hoc retrieval datasets

dataset	#genre	#queries	#documents
Robust04	news	250	0.5M
ClueWeb09-B	web pages	150	50M
Gov2	.gov pages	150	25M
MQ2007	.gov pages	1,692	25M
MQ2008	.gov pages	784	25M

- **ClueWeb09-B** is a large Web collection with 150 queries and over 50M English documents, whose topics are accumulated from TREC Web Track 2009, 2010, and 2011.
- **Gov2** is a crawl of the .gov domain Web pages with 25M documents. We use 150 topic queries that are accumulated from TREC Terabyte Tracks 2004, 2005, and 2006.
- **Million Query Track 2007 (MQ2007)** is a LETOR [28] benchmark dataset with 1692 queries, which uses the Gov2 Web collection.
- **Million Query Track 2008 (MQ2008)** is another LETOR benchmark dataset with 784 queries, which also leverages the Gov2 Web collection.

Note that ClueWeb09-B is filtered to the set of documents with spam scores in the 60th percentile, using the Waterloo Fusion spam scores [8]. The detailed statistics of these datasets are shown in Table 1. As we can see, there is a significant difference between the number of queries and documents, which poses the challenge of training deep neural models with such a few queries. Therefore, pre-training on large text corpus to learn universal properties can be beneficial for downstream ad-hoc retrieval tasks and avoid training deep neural model from scratch.

4.2 Baselines

We adopt three types of baseline methods for comparison, including traditional retrieve models, pre-trained models, and previous state-of-the-art neural ranking models.

For traditional retrieval models, we take two representative ranking methods:

- **QL**: Query likelihood model [43] is one of the best performing language models based on Dirichlet smoothing.
- **BM25**: The BM25 formula [31] is another highly effective retrieval model that represents the classical probabilistic retrieval model.

The pre-trained models include:

- **BERT**: The key technical innovation of BERT [11] is applying the multi-layer bidirectional Transformer encoder architecture for language modeling. BERT uses two different types of pre-training objectives, including Masked Language Model (MLM) and Next Sentence Prediction (NSP). Currently, BERT has become a strong baseline model for ad-hoc retrieval tasks due to its powerful contextual language representations. Different from passage-level

[1] https://dumps.wikimedia.org/
[2] https://github.com/attardi/wikiextractor
[3] https://microsoft.github.io/TREC-2019-Deep-Learning/
[4] https://microsoft.github.io/TREC-2020-Deep-Learning/

and sentence-level approaches [9, 40], we truncate the single input sequence of the concatenated query and document to BERT's max-length limit.

- **Transformer$_{ICT}$**: Inverse Cloze Task (ICT) [18] is specifically designed for passage retrieval in QA scenario which teaches model to predict the removed sentence given a context text. As observed in Chang et.al [5], ICT outperforms BFS and WLP task. Thus, we only choose ICT as the baseline for comparison. We pre-train the Transformer model on Wikipedia corpus with ICT and MLM for a fair comparison, and other experimental settings are set the same as PROP.

Besides the above baselines, we also compare PROP with existing state-of-the-art models (SOTA) on these five datasets, including CEDR-KNRM [20] on Robust04, BERT-maxP [9] on ClueWeb09-B, NWT [14] on Gov2, and HiNT [12] on MQ2007 and MQ2008. We only fetch the best results of these models from the original paper. For ClueWeb09-B, previous SOTA BERT-maxP from Dai and Callan [9] used a different set of queries, and thus we use their implementation to run the same query set for comparison.

4.3 Evaluation Methodology

Given the limited number of queries for each collection, we conduct 5-fold cross-validation to minimize overfitting without reducing the number of learning instances. The parameters for each model are tuned on 4-of-5 folds. The final fold in each case is used to evaluate the optimal parameters. As for evaluation measures, two standard evaluation metrics, i.e., normalized discounted cumulative gain (nDCG) and precision (P), are used in experiments. Specifically, for Robust04, ClueWeb09-B and Gov2, we report normalized discounted cumulative gain at rank 20 (NDCG@20) and precision at rank 20 (P@20) following existing works [9, 13, 20]. For MQ2007 and MQ2008, we report two official metrics used in LETOR 4.0: precision at rank 10 (P@10) and normalized discounted cumulative gain at rank 10 (NDCG@10) following existing works [12, 25], since there are less document candidates for each dataset on the two datasets.

4.4 Implementation Details

Here, we describe the implementation details of PROP, including model architecture, pre-training process and fine-tuning process.

4.4.1 Model Architecture. We use the Transformer encoder architecture similar to BERT$_{base}$ version [11], where the number of layers is 12, the hidden size is 768, the feed-forward layer size is 3072, the number of self-attention heads is 12, and the total parameters is 110M. For a fair comparison, PROP, BERT and Transformer$_{ICT}$ use the same model architecture in experiment. Specifically, we use the popular transformers library PyTorch-Transformers[5] for the implementation of PROP.

4.4.2 Pre-training Process. For representative word sets sampling, the expectation of interval λ is set to 3. To avoid sampling frequent words, we perform stopwords removal using the INQUERY stop list, discard the words that occur less than 50 times and use a subsampling of frequent words with sampling threshold of 10^{-5} as suggested by Word2Vec[6]. Word sets are sampled with replacement, i.e. the probability for each word remains the same for multi-sampling. We sample 5 pairs of word sets for each document.

For the representative words prediction, we lowercase the pre-training text and do not perform stemming or stop words removal. The single input sequence which is concatenated by the word set and the document, is fed to PROP. We use Adam optimizer with a linear warm-up over the first 10% steps and linear decay for later steps, and the learning rate is set to $2e-5$. Dropout with probability of 0.1 is applied on all layers. We train with batch size of 128 and sequence length of 512 for about 100-300k steps. Considering the large cost of training from scratch, we adopt the parameters of BERT$_{base}$ released by Google[7] to initialize the Transformer encoder. We pre-train PROP on 4 Nvidia Telsa V100-32GB GPUs.

For the masked language modeling, following BERT, we randomly select 15% words in the input document, and the selected words are (1) 80% of time replaced by a mask token [MASK], or (2) 10% of time replaced by a random token, or (3) 10% of time unchanged. Note that sampled words in representative word sets are not considered to be masked.

4.4.3 Fine-tuning Process. We adopt a re-ranking strategy for efficient computation. An initial retrieval is performed using the Anserini toolkit with BM25 model to obtain the top 200 ranked documents. We then use PROP to re-rank these top candidate documents. For all five downstream datasets, we conduct 5-fold cross-validation where each iteration uses three folds for training, one for validation, and a final held-out fold for testing. We employ a batch size among 16 and 32 and select the best fine-tuning learning rate of Adam among $1e-5$ and $2e-5$ on the validation set. For Robust04, ClueWeb09-B and Gov2 datasets, we perform the evaluation by using the five folds provided by Huston and Croft [17]. And for MQ2007 and MQ2008 datasets, we follow the data partition in LETOR4.0 [28]. For all pre-trained models including BERT, Transformer$_{ICT}$ and PROP, we use raw text as the input. The reason is that using standard stop words removal and words stemming will hurt performance for these pre-trained models in the fine-tuning phase since it is inconsistent with pre-training process.

4.5 Baseline Comparison

The performance comparisons between PROP and baselines are shown in Table 2. We can observe that: (1) Traditional retrieval models (i.e., *QL* and *BM25*) are strong baselines which perform pretty well on all downstream tasks. (2) By automatically learning text representations and relevance matching patterns between queries and documents, previous state-of-the-art neural ranking models can achieve better results than traditional retrieval models. For Robust04 and ClueWeb09-B, BERT-based models, i.e. CEDR and BERT-maxP, achieve significant improvements over *QL* and *BM25*, while traditional neural ranking models including DRMM and NWT shows slight improvements over *QL* and *BM25* on other three datasets (i.e., Gov2, MQ2007, and MQ2008). One possible reason is that it is difficult for a deep neural model training from scratch

[5] https://github.com/huggingface/transformers

[6] https://github.com/tmikolov/word2vec

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

Table 2: Comparisons between PROP and the baselines. $*, \dagger$ **and** $\ddagger$ **indicate statistically significance with** $p - value \leq 0.05$ **over BM25, BERT and Transformer$_{ICT}$, respectively.**

Model	Robust04		ClueWeb09-B		Gov2		MQ2007		MQ2008	
	nDCG@20	P@20	nDCG@20	P@20	nDCG@20	P@20	nDCG@10	P@10	nDCG@10	P@10
QL	0.413	0.367	0.225	0.326	0.409	0.510	0.423	0.371	0.223	0.241
BM25	0.412	0.363	0.230	0.334	0.421	0.523	0.414	0.366	0.220	0.245
Previous SOTA	**0.538**	**0.467**	0.296	-	0.422	0.524	0.490	0.418	0.244	0.255
BERT	0.459*	0.389*	0.295*	0.367*	0.495*	0.586*	0.506*	0.419*	0.247*	0.256*
Transformer$_{ICT}$	0.460*	0.388 *	0.298*	0.369*	0.499*†	0.587*	0.508*	0.420*	0.245*	0.256*
PROP$_{Wikipedia}$	**0.502***†‡	**0.421***†‡	0.316*†‡	0.384*†‡	0.519*†‡	0.593*†‡	**0.523***†‡	**0.432***†‡	0.262*†‡	0.267*†‡
PROP$_{MSMARCO}$	0.484*†‡	0.408*†‡	**0.329***†‡	**0.391***†‡	**0.525***†‡	**0.594***†‡	0.522*†‡	0.430*†‡	**0.266***†‡	**0.269***†‡

with such a few supervised pairs. (3) The improvements of *BERT* and *Transformer$_{ICT}$* over previous *SOTA* on Gov2, MQ2007 and MQ2008 datasets, demonstrate that pre-training and fine-tuning are helpful for downstream tasks.

When we look at the two types of *PROP* models pre-trained on Wikipedia and MS MARCO respectively, we find that: *PROP$_{Wikipedia}$* achieves better results than *PROP$_{MSMARCO}$* on Robust04, while *PROP$_{MSMARCO}$* performs better than *PROP$_{Wikipedia}$* on ClueWeb09-B. The reason might be that the news articles in Robust04 are similar with the well-formed articles in Wikipedia while the Web pages in ClueWeb09-B are similar with the Web documents in MS MARCO. These results suggest that employing the pre-trained model from a related domain for the downstream task is much more effective.

Finally, we can see that the best *PROP* model achieves the best performance in terms of all the evaluation metrics in 4 of 5 datasets. The observations are as follows: (1) PROP outperforms traditional retrieval models (i.e., *QL* and *BM25*) by a substantial margin. For example, the relative improvement of *PROP* over *BM25* is about 46% in terms of nDCG@20 on ClueWeb09-B. The results indicate the effectiveness of our pre-training method for ad-hoc retrieval. (2) Compared with previous *SOTA*, the relative improvements are about 8.9%, 24.4%, 6.7% and 9% in terms of nDCG@20 for ClueWeb09-B, Gov2, MQ2007 and MQ2008 respectively. For Robust04, CEDR-KNRM is better than PROP since CEDR integrates BERT's representations into existing neural ranking models, e.g. KNRM. Nevertheless, the results demonstrate that pre-training on a large corpus and then fine-tuning on downstream tasks is better than training a neural deep ranking model from scratch. (3) The improvements of PROP over existing pre-trained models (i.e., *BERT* and *Transformer$_{ICT}$*) indicate that designing a pre-training objective tailored for IR with a good theoretical foundation is better than directly applying pre-training objectives from NLP on IR tasks.

4.6 Impact of Pre-training Objectives

In this section, we investigate the effect of different pre-training objectives in PROP. Specifically, we pre-train the Transformer model with the ROP and MLM objective respectively on MS MARCO under the same experiment settings in PROP. As shown in Table 3, we report the nDCG results on 5 downstream tasks. We can see that: (1) Pre-training with MLM on MS MARCO shows slight improvements over BERT pre-trained on BooksCorpus and English Wikipedia (as shown in Table 2). It indicates that good representations obtained by MLM may not be sufficient for ad-hoc retrieval tasks. (2) Pre-training with ROP achieves significant improvements over MLM on all downstream tasks, showing the effectiveness of ROP tailored for IR. (3) By pre-training jointly with the ROP and MLM objective, PROP achieves the best performance on all downstream tasks. It indicates that the MLM objective which brings good contextual representations and the ROP objective which resembles the relevance relationship for ad-hoc retrieval tasks can contribute together.

Table 3: Impact of pre-training objectives. $\dagger$ **indicates statistically significance with** $p - value < 0.05$**.**

	nDCG@20			nDCG@10	
	Robust04	ClueWeb09-B	Gov2	MQ2007	MQ2008
w/ MLM	0.467	0.306	0.503	0.511	0.249
w/ ROP	0.481†	0.321†	0.519†	0.520†	0.262†
w/ ROP+MLM	**0.484**†	**0.329**†	**0.525**†	**0.522**†	**0.266**†

4.7 Impact of Sampling Strategies

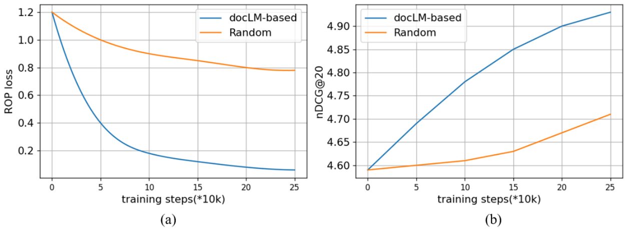
As described in Section 3.1, the representative word sets are sampled according to the document language model. To investigate the impact of different sampling strategies, we compare the document language model-based sampling strategy (docLM-based sampling for short) with the random sampling strategy, which randomly samples words from the corpus vocabulary independently. Specifically, we pre-train the Transformer model on English Wikipedia only with the ROP objective using docLM-based sampling and random

Table 4: Impact of Further Pre-training on Target Tasks. Two-tailed t-tests demonstrate the improvements of PROP with further pre-training to that without further pre-training are statistically significant († indicates p-value < 0.05).

	nDCG@20			nDCG@10	
	Robust04	ClueWeb09-B	Gov2	MQ2007	MQ2008
Without Further Pre-training	0.502	0.329	0.525	0.523	0.266
Further Pre-training	$0.506^\dagger$	$0.334^\dagger$	$0.531^\dagger$	$0.526^\dagger$	$0.270^\dagger$

Table 5: Impact of Different Sampling Strategies. Two-tailed t-tests demonstrate the improvements of document language model-based sampling to the random sampling strategy are statistically significant († indicates p-value < 0.05).

	nDCG@20			nDCG@10	
	Robust04	ClueWeb09-B	Gov2	MQ2007	MQ2008
Random	0.471	0.304	0.505	0.513	0.252
docLM-based	$0.493^\dagger$	$0.317^\dagger$	$0.517^\dagger$	$0.516^\dagger$	$0.257^\dagger$



(a)

(b)

Figure 1: (a) ROP learning curve on Wikipedia over the pre-training steps. (b) The test performance curve on Robust04 in terms of nDCG@20 over the pre-training steps.

sampling respectively. The loss curve of the ROP objective over the pre-training steps using different sampling strategies is depicted in Figure 1 (a). We can find that PROP based on the docLM-based sampling strategy converges much faster than that based on the random sampling strategy.

Moreover, we pre-train the Transformer model at most 250K steps for both sampling strategies, and further fine-tune them on the five downstream datasets. As shown in Table 5, we report the nDCG@20 results on Robust04, ClueWeb09-B, and Gov2 datasets, and the nDCG@10 results on MQ2007 and MQ2008 datasets. We find that PROP based on the docLM-based sampling strategy can achieve significantly better results than that based on the random sampling strategy. We also show the test performance curve in terms of nDCG@20 on Robust04 over the pre-training steps in Figure 1(b). We can observe that PROP based on the docLM-based sampling strategy improves the performance much faster than that based on the random sampling strategy.

All the above results indicate that the docLM-based sampling strategy is a more suitable way than the random sampling strategy to generate representative word sets for a document. The reason might be that the document language model roots in a good theoretical IR foundation and thus contributes to the efficiency and effectiveness of the pre-training process.

4.8 Further Pre-training on Target Tasks

Here, we analyze the impact of further pre-training on the document collections in the target tasks to see how much performance could be improved. Specifically, we further pre-train $\text{PROP}_{Wikipedia}$ on the document collections of Robust04 and MQ2007 respectively, and $\text{PROP}_{MSMARCO}$ on the document collections of ClueWeb09-B, Gov2 and MQ2008 respectively. As shown in Table 4, we can see that PROP with further pre-training on the document collection in the target tasks outperforms that without further pre-training. The results demonstrate that further pre-training on the related-domain corpus could improve the ability of PROP and achieve better performance on the downstream tasks, which is quite consistent with the previous findings [15]. However, the improvement of further pre-training over without further pre-training on MQ2007 dataset is less than that on other four datasets. The reason might be that MQ2007 has much more queries than other datasets, and enough in-domain information can be well captured during the fine-tuning process.

4.9 Zero- and Low-Resource Settings

In real-world practice, it is often time-consuming and difficult to collect a large number of relevance labels in IR evaluation. To simulate the low-resource IR setting, we pick the first 10, 30, 50, 70 queries from Robust04, ClueWeb09-B, and Gov2, and the first 50, 100, 150, 200 queries from MQ2007 and MQ2008 to fine-tune $\text{PROP}_{Wikipedia}$. We fine-tune the models with batch size as three different values (i.e., 4, 8, 16), learning rate as two different values (i.e., 1e-5, 2e-5), and pick the checkpoint with the best validation performance. As shown in Figure 2, we can find that: (1) *PROP* fine-tuned on limited supervised data can achieve comparable performance with BERT fine-tuned on the full supervised datasets in terms of nDCG and Precision. For example, PROP fine-tuned with only 30 queries has outperformed BERT on Robust04, ClueWeb09-B, and Gov2 datasets. (3) Furthermore, fine-tuning *PROP* with only 10 queries can achieve comparable results with traditional retrieval models (i.e., *QL*and *BM25*) for Robust04, ClueWeb09-B, and Gov2. The results demonstrate that by fine-tuning with small numbers of supervised pairs, PROP is able to adapt to the target task quickly. (4) Under the zero resource setting, for example, PROP can achieve about 90% performance of BERT fine-tuned on the full Gov2 dataset in terms of nDCG@20.

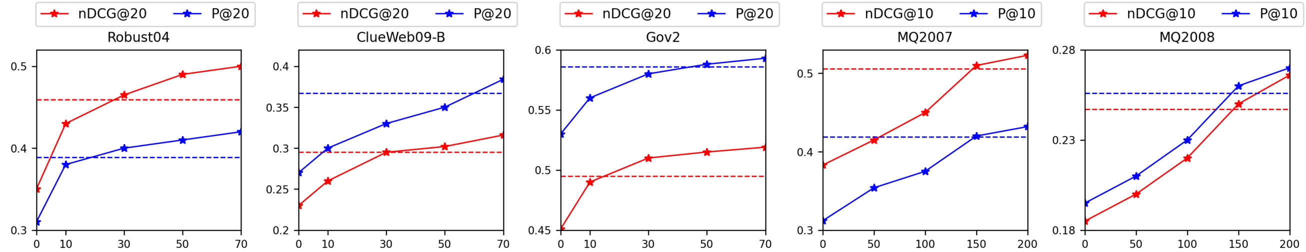


Figure 2: Fine-tuning with limited supervised data. The solid lines are PROP fine-tuned using 0 (zero shot), 10, 30, 50, and 70 queries for Robust04, ClueWeb09-B and Gov2 datasets, using 0 (zero shot), 50, 100, 150, and 200 queries for MQ2007 and MQ2008 datasets. The dashed lines are BERT fine-tuned using the full queries.

5 RELATED WORK

In this section, we briefly review two lines of related work, i.e., pre-trained language models and pre-training objectives for IR.

5.1 Pre-trained Language Models

Recently, pre-trained language representation models such as OpenAI GPT [30], BERT [11] and XLNET [41], have led to significant improvements on many NLP tasks. The key idea is that firstly pre-training a large neural architecture on massive amount of unlabeled data and then fine-tuning on downstream tasks with limited supervised data. Transformer [37] has become the mainstream architecture of pre-trained models due to its powerful capacity. Pretext tasks, such as probabilistic language modeling [4], Masked language modeling [36, 41] and Permuted Language Modeling (PLM) [41], have been proved effective in NLP since they can learn universal language representations and contribute to the downstream NLP tasks.

BERT, as the most prominent one among existing pre-trained models, pre-training the Transformer with MLM and NSP, to obtain contextual language representations and sentence-pair representations. Directly applying BERT to IR can achieve new state-of-the-art performance. A simple approach is to feed the query–document pair to BERT and use an MLP or more complicated module on top of BERT’s output to produce relevance score. Nogueira et al. [23, 24] trained BERT model on MS MARCO passage ranking task and TREC CAR using pointwise and pairwise approaches. Dai et al. [9] and Yang et al. [40] adopted passage-level and sentence-level methods for addressing the document length issue respectively, i.e. applying inference on sentences/passages individually, and then aggregating sentence/passage scores to produce document scores. MacAvaney et al. [20] integrated the representation of BERT’s [CLS] token into existing neural ranking models such as DRMM [13], PACRR [16] and KNRM [39]. Despite the success BERT has achieved in IR community, designing pre-training objectives specially for IR is still of great potential and importance.

5.2 Pre-training Objectives for IR

There has been little effort to design pre-training objectives towards ad-hoc retrieval. Most related work in this direction focused on passage retrieval in question answering (QA) [5, 18]. For example, Lee et al. [18] proposed Inverse Cloze Task (ICT) for passage retrieval, which randomly samples a sentence from passage as pseudo query and takes the rest sentences as the document. However, this method may lose the important exact matching patterns since the pseudo query is removed from the original document. In [5], Chang et al. proposed another two additional pre-training objectives, i.e., Body First Selection (BFS) and Wiki Link Prediction(WLP). BFS randomly samples a sentence in the first section of a Wikipedia page as pseudo query and the document is a randomly sampled paragraph from the same page. WLP chooses a random sentence in the first section of a Wikipedia page as pseudo query, then a document is sampled from another page where there is a hyperlink between these two pages. However, such pre-training objectives rely on special structure of the document (e.g., multiple paragraph segmentations and hyperlinks), which hinder the method to be applied on general text corpus. In summary, all these above pre-triaining tasks attempt to resemble the relevance relationship between natural language questions and answer passages. As demonstrated in our experiments, when applying pre-trained models based on these tasks to ad-hoc retrieval, marginal benefit could be observed on typical benchmark datasets.

6 CONCLUSION

In this paper, we have proposed PROP, a new pre-training method tailed for ad-hoc retrieval. The key idea is to pre-train the Transformer model to predict the pairwise preference between the two sets of words given a document, jointly with the MLM objective. PROP just needs to pre-train one model and then fine tune on a variety of downstream ad-hoc retrieval tasks. Through experiments on 5 benchmark ad-hoc retrieval datasets, PROP achieved significant improvements over the baseline without pre-training or with other pre-training methods. We also show that PROP can achieve strong performance under both the zero- and low-resource IR settings.

For future work, we would like to go beyond the ad-hoc retrieval, and try to test the ability of PROP over other types of downstream IR tasks, such as passage retrieval in QA or response retrieval in dialog systems. We will also investigate new ways to further enhance the pre-training tailored for IR.

7 ACKNOWLEDGMENTS

This work was supported by Beijing Academy of Artificial Intelligence (BAAI) under Grants No. BAAI2019ZD0306, and funded by the National Natural Science Foundation of China (NSFC) under Grants No. 61722211, 61773362, 61872338, 62006218, and 61902381, the Youth Innovation Promotion Association CAS under Grants No. 20144310, and 2016102, the National Key RD Program of China under Grants No. 2016QY02D0405, the Lenovo-CAS Joint Lab Youth Scientist Project, the K.C.Wong Education Foundation, and the Foundation and Frontier Research Key Program of Chongqing Science and Technology Commission (No. cstc2017jcyjBX0059).

REFERENCES

[1] Avi Arampatzis and Jaap Kamps. 2008. A Study of Query Length. In *Proceedings of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 811–812.

[2] Nima Asadi, Donald Metzler, Tamer Elsayed, and Jimmy Lin. 2011. Pseudo Test Collections for Learning Web Search Ranking Functions. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 1073–1082.

[3] Leif Azzopardi, Maarten de Rijke, and Krisztian Balog. 2007. Building Simulated Queries for Known-Item Topics: An Analysis Using Six European Languages. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 455–462.

[4] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *JMLR*, 1137–1155.

[5] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2019. Pre-training Tasks for Embedding-based Large-scale Retrieval. In *ICLR*.

[6] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training Text Encoders as Discriminators rather than Generators. In *ICLR*.

[7] Prem C Consul and Gaurav C Jain. 1973. A Generalization of the Poisson Distribution. In *Technometrics*, Vol. 15. Taylor & Francis, 791–799.

[8] Gordon V Cormack, Mark D Smucker, and Charles LA Clarke. 2011. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. In *Information retrieval*, Vol. 14. Springer, 441–465.

[9] Zhuyun Dai and Jamie Callan. 2019. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 985–988.

[10] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 65–74.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, Stroudsburg, PA, USA, 4171–4186.

[12] Yixing Fan, Jiafeng Guo, Yanyan Lan, Jun Xu, Chengxiang Zhai, and Xueqi Cheng. 2018. Modeling Diverse Relevance Patterns in Ad-hoc Retrieval. In *The 41th international ACM SIGIR conference on research & development in information retrieval*. ACM, New York, NY, USA, 375–384.

[13] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, 55–64.

[14] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. Semantic Matching by Non-linear Word Transportation for Information Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, 701–710.

[15] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. ACL, Stroudsburg, PA, USA, 328–339.

[16] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. ACL, Stroudsburg, PA, USA, 1049–1058.

[17] Samuel Huston and W Bruce Croft. 2014. Parameters Learned in the Comparison of Retrieval Models using Term Dependencies. In *IR, UMASS*. Citeseer.

[18] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. ACL, Stroudsburg, PA, USA, 6086–6096.

[19] Xiaoyong Liu and W Bruce Croft. 2005. Statistical Language Modeling for Information Retrieval. In *IR, UMASS*.

[20] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 1101–1104.

[21] Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019. Content-Based Weak Supervision for Ad-Hoc Re-Ranking. In *Proceedings of the 42th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 993–996.

[22] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Frontmatter*. Cambridge University Press, i–iv.

[23] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085*.

[24] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage Document Ranking with BERT. *arXiv preprint arXiv:1910.14424*.

[25] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A New Deep Architecture for Relevance Ranking in Information Retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, 257–266.

[26] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, Stroudsburg, PA, USA, 2227–2237.

[27] Jay M Ponte and W Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, 275–281.

[28] Tao Qin and Tie-Yan Liu. 2013. Introducing LETOR 4.0 datasets. *arXiv preprint arXiv:1306.2597*.

[29] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained Models for Natural Language Processing: A Survey. *arXiv preprint arXiv:2003.08271*.

[30] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

[31] Stephen E Robertson and Steve Walker. 1994. Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer, 232–241.

[32] Erik F Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. ACL, Stroudsburg, PA, USA, 142–147.

[33] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. 1631–1642.

[34] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked Sequence to Sequence Pre-training for Language Generation. In *ICML*. 11328–11339.

[35] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, 3104–3112.

[36] Wilson L Taylor. 1953. "Cloze Procedure": A New Tool for Measuring Readability. In *Journalism quarterly*, Vol. 30. 415–433.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in neural information processing systems*. 5998–6008.

[38] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating Language Structures into Pre-training for Deep Language Understanding. In *ICLR*.

[39] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 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*. ACM, New York, NY, USA, 55–64.

[40] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple Applications of BERT for Ad Hoc Document Retrieval. *arXiv preprint arXiv:1903.10972* (2019).

[41] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 5753–5763.

[42] ChengXiang Zhai. 2008. Statistical Language Models for Information Retrieval. *Synthesis lectures on human language technologies* 1, 1 (2008), 1–141.

[43] Chengxiang Zhai and John Lafferty. 2001. A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval. In *ACM SIGIR Forum*. New York, NY, USA, ACM, 268–276.

[44] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *ICML*. 11328–11339.