Axiomatically Regularized Pre-training for Ad hoc Search

Jia Chen
BNRist, DCST, Tsinghua University
Beijing 100084, China
chenjia0831@gmail.com

Yiqun Liu*
BNRist, DCST, Tsinghua University
Beijing 100084, China
yiqunliu@tsinghua.edu.cn

Yan Fang
BNRist, DCST, Tsinghua University
Beijing 100084, China
fangy21@mails.tsinghua.edu.cn

Jiaxin Mao
GSAI, Renmin University of China
Beijing 100872, China
maojiaxin@gmail.com

Hui Fang
DECE, University of Delaware
Newark, USA
hfang@udel.edu

Shenghao Yang
BNRist, DCST, Tsinghua University
Beijing 100084, China
ysh21@mails.tsinghua.edu.cn

Xiaohui Xie
BNRist, DCST, Tsinghua University
Beijing 100084, China
xiexiaohui@mail.tsinghua.edu.cn

Min Zhang
BNRist, DCST, Tsinghua University
Beijing 100084, China
z-m@tsinghua.edu.cn

Shaoping Ma
BNRist, DCST, Tsinghua University
Beijing 100084, China
msp@tsinghua.edu.cn

ABSTRACT

Recently, pre-training methods tailored for IR tasks have achieved great success. However, as the mechanisms behind the performance improvement remain under-investigated, the interpretability and robustness of these pre-trained models still need to be improved. Axiomatic IR aims to identify a set of desirable properties expressed mathematically as formal constraints to guide the design of ranking models. Existing studies have already shown that considering certain axioms may help improve the effectiveness and interpretability of IR models. However, there still lack efforts of incorporating these IR axioms into pre-training methodologies. To shed light on this research question, we propose a novel pre-training method with Axiomatic Regularization for ad hoc Search (ARES). In the ARES framework, a number of existing IR axioms are re-organized to generate training samples to be fitted in the pre-training process. These training samples then guide neural rankers to learn the desirable ranking properties. Compared to existing pre-training approaches, ARES is more intuitive and explainable. Experimental results on multiple publicly available benchmark datasets have shown the effectiveness of ARES in both full-resource and low-resource (e.g., zero-shot and few-shot) settings. An intuitive case study also indicates that ARES has learned useful knowledge that existing pre-trained models (e.g., BERT and PROP) fail to possess. This work provides insights into improving the interpretability of pre-trained models and the guidance of incorporating IR axioms or human heuristics into pre-training methods.

*Corresponding author, yiqunliu@tsinghua.edu.cn

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531943

CCS CONCEPTS

• **Information systems → Retrieval models and ranking**;

KEYWORDS

Pre-training Method, Pre-trained Language Model, Axiomatic IR, Ad hoc Search

ACM Reference Format:
Jia Chen, Yiqun Liu*, Yan Fang, Jiaxin Mao, Hui Fang, Shenghao Yang, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Axiomatically Regularized Pre-training for Ad hoc Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531943

1 INTRODUCTION

These years have witnessed the flourish of pre-trained language models in Natural Language Processing (NLP) [11, 34, 39, 47]. Based on the pre-training and fine-tuning paradigm, classical transformer models such as BERT [11] have achieved state-of-the-art performance in various downstream tasks. Recently, the great success of these pre-trained models (PTMs) has also attracted much attention from the Information Retrieval (IR) community [45, 46, 49]. Besides applying PTMs in various tasks [12], researchers also aimed at designing pre-training approaches tailored to improve performance of specific tasks such as ad hoc retrieval [4, 26, 28]. Although their work achieved promising retrieval performance, the mechanism behind the performance improvement remains under-investigated. Most existing pre-trained models are like black boxes because their off-the-shelf parameters will be gradually updated in the fine-tuning stage. In this regard, these models may lack interpretability and hence be vulnerable to potential malicious textual attacks.

To increase the interpretability, effectiveness, and robustness of ranking models, researchers have attempted to introduce certain IR axioms or human heuristics into model training [8, 20, 22, 36]. Generally, axiomatic IR aims at formalizing a set of desirable constraints that any reasonable IR models should (at least partially) satisfy. Each axiom mainly focuses on a single attribute that a good ranking function should possess. For example, the basic TFC1 [13]

axiom states that we should give a higher score to the document with more occurrences of a query term. Similarly, the STM [15] and the PROX [20] axiom families focus on the semantic term matching and term proximity constraints, respectively. Either by adding constraints with these axioms on the learning objective or augmenting the perturbed pairwise cases to the training set, retrieval models can be improved by a certain margin in terms of ranking performance. However, to our best knowledge, there are still no existing efforts that consider these axioms in the pre-training process. In addition, most axioms are defined to provide pairwise preference on a document pair given a specific query. It is therefore relatively difficult to directly apply them in the pre-training process where only documents are available and queries need to be generated.

To incorporate the IR axioms into the pre-training process of neural ranking models, we propose a novel pre-training framework with Axiomatic Regularization for ad hoc Search (ARES). ARES mainly consists of three stages: 1) Pseudo Query Sampling (PQS), 2) Preference Predictor Constructing (PPC), and 3) Axiomatically Regularized Pre-training (ARP). In the PQS stage, we sampled a set of pseudo queries from each document in the corpus with a simple yet effective contrastive sampling strategy. Then in the PPC stage, we collected ordered query pairs through four sampling settings and extracted axiomatic features for each pair. The axiomatic feature map and the weak preference labels will be further used to train a preference predictor, i.e., an axiomatic binary decision model. In the final stage of ARP, the query pairs regularized with axioms will be fed into the pre-training process, meaning that we rely on query pairs instead of document pairs to teach the ranking models.

Compared with existing pre-trained methods, our proposed ARES training strategy helps learn the model designing knowledge concluded by the IR community in the last decades in the form of IR axioms. We can control the rules to teach ranking models by using different subsets of axioms. Therefore, ARES does not require large-scale supervision data to fine-tune in different task settings as much as existing IR-oriented pre-trained models do. Compared with these models, ARES is expected to be more interpretable and able to gain better performance in low-resource scenarios (e.g., few-shot and zero-shot settings [27, 28]).

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

- We propose a novel axiomatic-based pre-training method, namely ARES. Compared to existing approaches, the pre-training stage of ARES is more explainable and better fits low-resource settings.
- We summarize nine adaptive axioms from existing axioms or heuristics and categorize them into groups so that they can be easily applied in the pre-training process.
- Experimental results on multiple public datasets have shown the effectiveness of ARES in both full-resource and low-resource (zero-shot/few-shot) settings. We further find that ARES is the only PTM that outperforms BM25 on all datasets in low-resource settings. An intuitive case study also reveals that ARES has learned retrieval knowledge described in IR axioms as expected.

2 RELATED WORK

2.1 Pre-trained Language Models

In recent years, pre-trained language models including BERT [11], Open AI GPT [34], and XLNET [47] have led the trend in the Natural Language Processing (NLP) field. By leveraging the two-stage paradigm (first pre-training the model on a large-scale unlabeled corpus by optimizing a self-supervised learning loss function and then fine-tuning it on limited supervised data), these models can achieve significantly better performance on a number of downstream tasks. Due to its strong ability to learn contextualized textual representations, Transformer [39]-based architectures have become the basic module in models dealing with various IR tasks [12], such as dense retrieval [21, 45, 49], query expansion [31], and context-aware ranking [5, 51]. Despite the promising performance achieved by directly employing BERT, researchers have found that designing learning objectives tailored for IR can help the model better handle the ranking task. For example, Ma et al. [26] proposed Representative Words Prediction (ROP) task for pre-training by assuming that a sampled word set with a higher query likelihood score is more "representative" to the document. By modeling different dependencies between the hyperlinks and anchor texts in Wikipedia pages, Ma et al. [28] presented a new model named HARP, which has achieved state-of-the-art performance in the ad hoc retrieval task. However, the mechanisms behind these models are far from being thoroughly discussed. Unlike B-PROP [27] (improves the query sampling strategy by bootstrapping), HARP [28] (leverages external knowledge such as hyperlink relationships) and Condenser/coCondenser [16, 17] (designs more efficient model architectures), we focus more on increasing the interpretability of pre-trained models by considering certain IR axioms. Furthermore, while previous PTMs are usually data-hungry, we find that incorporating certain axioms into pre-training can also lead to promising zero-shot performance.

2.2 Axiomatic Information Retrieval

The utilization of axioms or retrieval heuristics to better understand and improve information retrieval techniques has been well established. It were Fang et al. [13] who first introduced several text matching-based heuristics that good retrieval models should follow to effectively handle various retrieval tasks. In the past two decades, more than 20 axioms have been proposed so far. According to the problem they aim to focus, these axioms can be mainly divided into several groups: term frequency [13, 14], document length [13], lower bounds [25], query aspects [18, 43, 50], semantic similarity [15], and term proximity [20]. Each group focuses on one particular aspect, e.g., document length axioms define the constraints on the length of a document while term proximity axioms restrain the positions of each query term appearing in the document. There exist studies aiming at analyzing neural rankers with existing axioms [2, 3, 6, 40]. For example, Câmara and Hauff [3] constructed a diagnostic dataset to explore whether BERT can learn some existing heuristics. Their results have shown that BERT, while performing significantly better than traditional models for ad hoc retrieval, does not fulfill most retrieval heuristics created by IR experts. Besides analyzing ranking models, IR axioms and heuristics have also been employed for improving ranking models [8, 20, 22, 36]. By adding the axiom-based constraints on the learning objective, Rosset et al. [36] improved the performance of neural models such as Conv-KNRM by a certain margin. From another perspective, Hagen et al. [20] adopted the learning-to-rank idea to re-rank the top-k results directly using promising axiom

combinations. Although these studies have achieved some success in better understanding IR models, whether considering certain axioms while pre-training is useful and how to incorporate them into the pre-training process remain under-investigated.

3 AXIOMS FOR PRE-TRAINING

In this section, we will briefly introduce the basic knowledge of axiomatic ideas for information retrieval and the adaptive axioms we use for pre-training. In Section §3.1, we first give an overview of classical axioms. These axioms usually provide the preference judgment between paired documents given a specific query under some assumptions. In this regard, they can hardly be directly exploited in our pre-training setting where 1) queries need to be generated from the document, and 2) the preference for a query pair w.r.t. a document needs to be decided. Therefore, we modify these axioms into query-centric ones to adapt to the pre-training process (as described in Section §3.2).

3.1 Review: IR Axioms

Through decades of development, a system of IR axioms has been well established. Existing axioms can be broadly divided into six following groups according to the aspects they emphasize:

- *Term frequency*: e.g., TFC1-TFC2 [13], TFC3 [14], TDC [13].
- *Document length*: e.g., LNC1 [13], LNC2 [13], TF-LNC [13].
- *Lower-bounding term frequency*: e.g., LB1-LB2 [25].
- *Query aspects*: e.g., REG [43], AND [50], DIV [18].
- *Semantic similarity*: e.g., STM1-STM3 [15].
- *Term proximity*: e.g., PROX1-PROX5 [20].

Among them, term frequency and document length constraints are the first to be proposed and are also the most fundamental ones. For example, TFC1 states that *we should give a higher score to a document with more occurrences of a query term.* LNC1 says that *the score of a document should decrease if we add an extra occurrence of a non-relevant word.* These two groups of axioms are too general; hence they may play little role in improving the performance of the sophisticated transformer model. The lower-bounding constraints emphasize the presence-absence gap (0-1 gap) of a query term, i.e., the marginal effect. From another perspective, the REG axiom describes that a document that covers more query aspects should be assigned a higher score. Previous work [40] has found that the REG axiom can do well in explaining the neural ranking models. Therefore, here we regard REG as a good axiom to increase the interpretability of pre-trained models. Furthermore, the semantic similarity and term proximity constraints have also been proved to be effective in improving existing models [36, 40]. We will also consider them later in the pre-training process.

For a comprehensive overview of axiomatic thinking for IR, we recommend readers to refer to this guideline[1].

3.2 Adaptive Axioms

To better utilize these heuristics in the pre-training process, we consider nine different axioms and rearrange them into five groups as shown in Table 1. The main difference between these adaptive axioms and the existing ones is that adaptive axioms give the preference judgment on a pair of queries given a specific document ($< q_1, q_2, d >$). The five groups are RANK, REP, PROX, REG, and STM, respectively. Among them, we regard RANK and REP as the basic axioms while the rest ones as the auxiliary axioms. Here "basic" means "fundamental" or "necessary" for pre-training rather than "simple". Basic axioms usually describe a complex, high-level concept closer to some ranking function or the definition of "relevance". They may already cover multiple aspects that a good ranking model should consider and thus can be used alone without being combined with other axioms. By contrast, the auxiliary axioms merely capture a single aspect and can hardly be leveraged alone. Next, we will give a detailed description of these adaptive axioms.

3.2.1 RANK. The main idea of the **RANK** axiom is that for any document, there may exist one best query that a reasonable ranking function can rank the document at the top position among the corpus. To determine the RANK value, a query, a document, and the corpus are necessary. Empirically, the ranking function should be an efficient and effective one so that the cost of retrieving documents for a large number of queries from the corpus is affordable. To this end, here we adopt BM25 as the ranking model and retrieve the top 50 documents from the whole corpus to determine the RANK value of a query. If a query cannot rank the document within the top 50 positions among the corpus, then we set $RANK = +\infty$.

3.2.2 REP. Following previous work [26, 27], we also consider the representativeness of a query with regard to the corresponding document. In summary, REP requires that a good query should be more representative of the given document than the randomly sampled ones. To formalize the representativeness of a query, we calculate the normalized Query Likelihood (QL) and TF-IDF scores, denoted as **REP-QL** and **REP-TFIDF**, respectively. Here the normalization is conducted to avoid the length bias (e.g., the longer the query, the smaller/higher the QL/TF-IDF score). The REP axioms mainly differ from the RANK axiom in two ways: I) RANK can be regarded as the relative order of REP scores; hence the distribution of REP scores can be denser than that of RANK because there may exist two queries with the same RANK value but with different REP-QL scores; II) The REP scores can be directly computed given a query and document pair without going deep into the corpus.

3.2.3 PROX. The PROX axioms aim to bridge the connection between query quality and the positions at which query terms appear in the document. For example, the **PROX-1** axiom prefers the queries whose terms appear more closely to each other in the document. Given a tuple $< q_1, q_2, d >$, if all terms in q_1 and q_2 appear in d, then the average position difference of term pairs for q within d can be calculated as:

$$\pi(q, d) = \frac{1}{|P|} \sum_{(t_i, t_j) \in P} \delta(d, t_i, t_j); \tag{1}$$

where $P = \{(t_i, t_j | t_i, t_j \in q, t_i \neq t_j)\}$ is the set of all possible query term pairs, and $\delta(d, t_i, t_j)$ is the average number of the words appearing between term t_i and t_j. If $\pi(q_1, d) < \pi(q_2, d)$, then q_1 is better than q_2. Intuitively, PROX-1 may emphasize the sentence coherence and highlight the matching of bigram or trigram phrases, while the REP axioms merely focus on single terms.

[1] https://www.eecis.udel.edu/~hfang/AX.html

From another perspective, the **PROX-2** axiom prefers the queries whose terms appear at earlier positions in a document. The average position of the first occurrence of all terms t in q given a document d can be formalized as:

$$\mu(q,d) = \frac{1}{N}\sum_{t\in q}\theta(t,d); \tag{2}$$

where $\theta(t,d)$ is the position of the first occurrence of term t in d, and N is the number of unique terms in q. The smaller $\mu(q,d)$ is, the better the query is. The assumption of PROX-2 is consistent with the vertically decaying human reading behavior [22, 23]. The essence of a document, such as the title and the summary, tends to appear at the head of a document.

3.2.4 REG. As mentioned before, some axioms describe the nature of the query itself. Concretely, the **REG** axiom prefers the queries whose most diverse term appears in a document with more times. In other words, we should give a higher score to a query if the document can cover more aspects of this query. To obtain the most diverse term, we calculate the semantic similarity between each term and the rest part of the query. The term with the lowest similarity will be further selected to count the REG value.

3.2.5 STM. Previous axioms mainly consider the syntactic connections between query and document. To capture the query-document relationship at the semantic level, we consider three semantic matching heuristics. The **STM-1** axiom prefers the query with a higher semantic similarity with the document. For convenience, we use the average-pooled word vectors to represent a query/document embedding. As supplementary, the **STM-2** axiom is used to distinguish two queries if they have very close similarities with the same document. Under this circumstance, the query with more terms appearing in the document is better. Accordingly, this axiom attaches more importance to exact matches than semantically similar terms. Finally, the **STM-3** axiom favors the query with more terms that have a similarity with the document higher than a threshold.

4 ARES

In this section, we will describe the details of the ARES framework as shown in Figure 1. The key point of our approach is to better resemble the relevance relationship between query and document in the pre-training process by leveraging the adaptive axioms defined in Section §3.2. ARES mainly consists of three stages: 1) Pseudo Query Sampling (PQS), 2) Preference Predictor Constructing (PPC), and 3) Axiomatically Regularized Pre-training (ARP).

4.1 Pseudo Query Sampling

Our work inherits the spirit of PROP [26] and B-PROP [27] which adopt Representative Words Prediction (ROP) as the pre-training task. In their assumption, a query with a higher query likelihood score is more "representative" of the corresponding document thus should be assigned a higher score. Similarly, ARES aims to train a transformer model by predicting the pairwise preference between two sampled queries based on their axiomatic preference for a document. To this end, we need to sample a number of pseudo queries from each document in the corpus for comparison.

Table 1: Adaptive axiom descriptions. Here "$\succ$" denotes the preference of a specific axiom.

Axiom	Description ($< q_1, q_2, d >$)
RANK	Given d, If a reasonable ranking function ϕ can rank d higher under q_1 than under q_2 among the whole corpus, then $q_1 \succ q_2$.
REP-QL	Given d, if q_1 can be generated from d with a higher query likelihood score than q_2, then $q_1 \succ q_2$.
REP-TFIDF	Given d, if q_1 has a higher normalized TF-IDF score than q_2, then $q_1 \succ q_2$.
PROX-1	If terms in q_1 appear more closely to each other in d than q_2, then $q_1 \succ q_2$.
PROX-2	If the first occurrences of terms in q_1 appearing in d precede that of q_2, then $q_1 \succ q_2$.
REG	Given d, if d can cover more aspects of q_1, then $q_1 \succ q_2$.
STM-1	Given d, if q_1 is more semantically related to d than q_2, then $q_1 \succ q_2$.
STM-2	Given that the similarity difference between q_1 and q_2 with d is smaller than a threshold, if there are more q_1 terms appearing in d than q_2, then $q_1 \succ q_2$.
STM-3	If there are more terms semantically similar to d in q_1 than q_2, then $q_1 \succ q_2$.

In this work, we do not focus on bootstrapping the sampling approach with complicated self-attention architectures in BERT. Inspired by the divergence-from-randomness idea [1], we adopt a simple yet effective strategy based on the contrastive term distribution. The main assumption of the strategy is to sample more representative queries so that the pre-training task will be relatively more challenging and the model can learn more useful knowledge.

To begin with, the term distribution for a specific document $P(w|\theta_D)$ and the general term distribution of the corpus $P(w|\theta_C)$ can be calculated as follows:

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

$$P(w|\theta_C) = \frac{DF(w) + 1}{\sum_{w'\in V} DF(w') + |V|}, \tag{4}$$

where $c(w,D)$ denotes the number of term w in a document D, μ is the Dirichlet smoothing parameter, $DF(w)$ represents the document frequency of term w, and V is the vocabulary set.

Then we obtain the contrastive term distribution by computing the divergence between the document term distribution and the general term distribution:

$$\gamma_w = -P(w|\theta_D)\log P(w|\theta_C), \tag{5}$$

$$P(w|\theta_{contrastive}) = \frac{exp(\gamma_w)}{\sum_{w\in V} exp(\gamma_w)}; \tag{6}$$

Here $P(w|\theta_{contrastive})$ is the contrastive term distribution. If a term w is more representative of the document then this probability will be higher. The softmax function ensures that the summation of probabilities over all terms is 1.

Given a document and its contrastive term distribution, we sample queries with equal length (so that some axioms can be directly

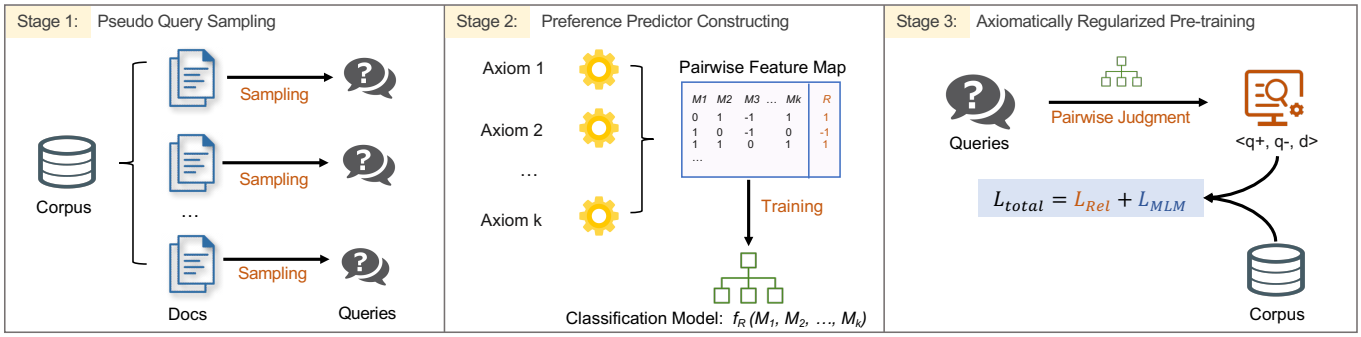


Figure 1: Illustration of the proposed ARES framework. Generally, there are three main stages: 1) Pseudo Query Sampling (sample some query term sets from a given document), 2) Preference Predictor Constructing (extract pairwise axiomatic features and preference labels to train a classification model), 3) Axiomatically Regularized Pre-training (judge the preference for sampled query pairs, and then pre-train the vanilla BERT model with a multi-task learning objective).

applied without modification). Following previous work [26–28], we first draw a Possion distribution to sample a query length l and then sample the pseudo query q (i.e., an unordered word set) according to the length based on the contrastive term distribution:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x}, x = 1, 2, 3, ..., \tag{7}$$

$$q = \{w_1, ...w_l\}, w_k \sim P(w|\theta_{contrastive}) \tag{8}$$

By investigating the RANK scores of all pseudo queries, the distribution is categorized into five intervals: $[1, 2)$ (32.08%), $[2, 5]$ (11.19%), $[6, 10]$ (4.28%), $[11, 50]$ (10.07%), and $[51, +\infty)$ (42.38%). More than 30% of these queries can rank the corresponding document first among the whole corpus. The overall distribution is also not highly concentrated, indicating the rationality of our sampling strategy.

4.2 Preference Predictor Constructing

Before preparing reasonable $< q+, q-, d >$ tuples for pre-training, we first need to construct an axiomatic preference predictor. Once the preference predictor is trained, it can be applied in any corpus to generate axiomatic preference labels for query pairs and no labeled data is needed anymore for pre-training. To better explain the role each axiom plays in the decision process and to improve the prediction accuracy, here we chose XGBoost [7] as the classification model. We then used the training set of the well-known MS MARCO document ranking [30] task to sample the query pairs for judgment. Each document labeled as "relevant" to one or multiple queries may be used to generate training pairs. Here we regarded all training queries as positive examples of their relevant documents.

As for negative queries, we designed four settings to sample them from the whole pseudo query set that we obtained in Section §4.1. Detailed descriptions for each setting are given in Table 2. To capture discrimination at various levels, we collected different subsets of pseudo queries by adding some constraints. For I, we randomly sampled negative queries from all pseudo queries. As a certain proportion of low-quality queries are sampled in setting I, it will be relatively simple for the model to learn the difference. The decisions are increasingly intractable from setting II to IV, as negative queries become more competitive against the positive ones, e.g., setting IV only selects the queries which can rank the corresponding document within the top five positions. For each document, we sample one negative query to organize the training cases.

Table 2: Four settings of sampling negative queries. The discrimination difficulties are incremental from setting I-IV.

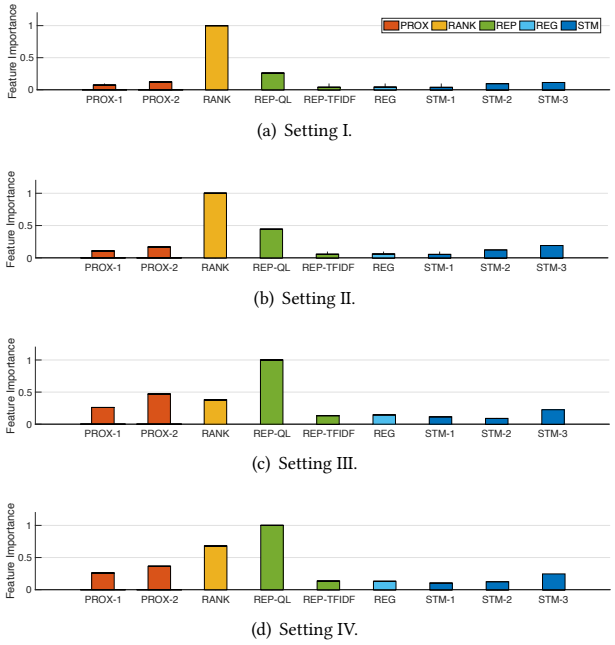
Setting	Sampling strategies	AUC
I	Randomly sample negative queries from all pseudo queries for a document.	0.9027
II	Sample according to $p = \text{softmax}(1/RANK)$.	0.8714
III	Ignore the documents if none of its pseudo queries has a *RANK* value of 1. Then sample queries from the rest documents as done in I.	0.7734
IV	Only consider those queries whose *RANK* values are less than or equal to 5.	0.8258

Given all positive and negative query pairs, we further extract preference features for each axiom. We first randomly shuffled all query pairs to balance the number of preference labels (0/1). Then for each axiom A for each $< q_i, q_j, d >$ tuple, we collect a feature matrix M as follows:

$$M_A[i, j] = \begin{cases} 1, & \text{if } q_i \succ_A q_j, \\ 0, & \text{if } q_i =_A q_j, \\ -1, & otherwise. \end{cases} \tag{9}$$

Here $q_i \succ_A q_j$ represents that the axiom A prefers q_i over q_j and $q_i =_A q_j$ says that A deems the two queries as the same. We split all pairs into training and testing set with a ratio of 9:1. As the data size is considerable (all settings contain over 150k cases), we applied two-fold cross-validation while training and used the best-performing model to predict the testing cases. The averaged predicting AUC scores are presented in the third column of Table 2. We can observe that the prediction accuracy roughly decreases from setting I to IV, which is consistent with our expectation because the negative queries become harder.

To further investigate the role each axiom plays in the decision process, we plot the distribution of feature importance (based on information gain) in Figure 2. Generally, in setting I and II, the



(a) Setting I.

(b) Setting II.

(c) Setting III.

(d) Setting IV.

Figure 2: Axiom importance learned by XGBoost (normalized by the axiom with the highest importance).

decision process is dominated by RANK, REP-QL, PROX-2, and STM-3, while in the setting III and IV, REP-QL, PROXs, and RANK are more important. The importance of the RANK axiom is much higher than that of other axioms in setting I and II. This phenomenon is reasonable because the negative queries are relatively easy to be distinguished from the positive ones in these two settings. Hence, only using the RANK axiom can achieve a rather promising prediction accuracy. However, in setting III and IV, negative queries have almost the same RANK value as the positive ones. More discriminative axioms such as REP-QL and PROXs will have greater impacts in these scenarios. The feature distributions of setting III/IV also provide empirical support for the basic idea of PROP and B-PROP, which mainly consider the REP-QL axiom and have achieved promising performance in the ad hoc retrieval task.

As the axiom importance distribution is similar between the I/II and III/IV settings, we will only consider the setting I (denoted as ARES_{simple}) and IV (denoted as ARES_{hard}) hereinafter. In addition, we consider the setting where all axioms do not conflict (denoted as ARES_{strict}), i.e., we only select the cases where all axioms prefer the same query. To validate the effectiveness of both basic and auxiliary axioms, we also consider two other variants ARES_{REP} and ARES_{RANK} by leveraging only the REP and RANK axioms.

4.3 Axiomatically Regularized Pre-training

Based on the pseudo queries we sampled from each document in the corpus, we applied the axiomatic preference predictor trained in Section §4.2 to conduct pairwise preference judgment among these queries. Generally, ARES aims to jointly optimize a loss function as presented in Equation 10:

$$\mathcal{L}_{total} = \mathcal{L}_{Rel} + \mathcal{L}_{MLM}; \tag{10}$$

Here $\mathcal{L}_{Rel}$ is a pairwise loss (similar to the ROP objective) which can help the model better fit the definition of relevance, and $\mathcal{L}_{MLM}$ is the Masked Language Modeling (MLM) [11] objective. For the pairwise loss, we use margin ranking loss (a.k.a., hinge loss) to ensure that the model can learn the axiomatic knowledge in the pre-training process (Equation 11):

$$\mathcal{L}_{Rel} = max(0, margin - P(q_+|d) + P(q_-|d)); \tag{11}$$

$P(q_+|d)$ and $P(q_-|d)$ denote how relevant q_+ and q_- is to the document d based on the model prediction. Following previous work [26, 28], we empiricially set $margin = 1$ here. For all transformer-based models, $P(q|d)$ can be obtained by calculating $MLP(h_{[CLS]})$, where $h_{[CLS]}$ is the pooled output of transformer.

By reconstructing the language pattern, the MLM objective has been proved crucial to learn good contextual representations for queries and documents. It can be defined as follows:

$$\mathcal{L}_{MLM} = -\sum_{\hat{x} \in m(x)} \log p(\hat{x}|x_{\backslash m(x)}); \tag{12}$$

where x denotes the input sequence, $m(x)$ and $x_{\backslash m(x)}$ are the masked word set and the rest words in x, respectively.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 Datasets. We leverage MS MARCO document collection [30] as the pre-training corpus. It contains over 3.2M high-quality web page documents, which is sufficient to support pre-training.

For downstream tasks, we fine-tune ARES on five widely-used ad hoc retrieval benchmarks: MS MARCO Document Ranking (MS MARCO) [30], TREC 2019 Deep Learning Track (TREC DL 2019) [9], Robust04 [41], Million Query Track 2007 (MQ2007) [33], and TREC COVID [42]. Basic statistics of these datasets are presented in Table 3. Although sharing the same training set, DL 2019 collects finer-grained human labels for 43 queries while MS MARCO contains 0/1 labels for 5,193 testing queries. The data sizes of Robust04 and MQ2007 are relatively small. For years, they have been widely used for evaluating the performance of ranking models. By contrast, TREC COVID is a new dataset that contains the questions and articles concerning the pandemic of COVID-19.

Among them, MS MARCO, TREC DL 2019, and TREC COVID contain plenty of training cases (over 300k). The fine-tuned system performance may tell little about the knowledge learned by off-the-shelf pre-trained models. To this end, we also aim to investigate the effectiveness of various PTMs by testing their zero-shot and few-shot performances on these three datasets. In addition, to test the adaptive ability of each model, we also adopt the testing set of EntityQuestions (EQ) [37] dataset (including 22,036 queries) to evaluate their zero-shot performance.

5.1.2 Baselines. We consider three groups of baseline models for performance comparison:

- Traditional IR Models:
  - **BM25** [35] is a classical and highly effective probabilistic retrieval model, usually used for first-stage retrieval.

Table 3: Basic statistics of all datasets. The superscript "1/2" denotes that the dataset will be used for fine-tuning/evaluating the zero-shot performance, respectively.

Dataset	Genre	#Queries	#Documents
MS MARCO[1,2]	web pages	0.37M	3.2M
TREC DL 2019[1,2]	web pages	0.37M	3.2M
Robust04[1]	news	250	0.5M
MQ2007[1]	.gov pages	1,692	25M
TREC COVID[1,2]	biomedical articles	0.32M	8.8M
EntityQuestions[2]	wikipedia pages	0.22M	21M

  - **QL** [48] is one of the best performing language models that are based on Dirichlet smoothing.
- Neural IR Models:
  - **KNRM** [44] is an interactive-based neural ranking model that uses kernel-pooling to provide soft matching signals for queries and documents.
  - **Conv-KNRM** [10] fuses the contextual information of surrounding words for matching by adding a convolutional layer based on KNRM.
- Pre-trained Models:
  - **BERT** [11] is a multi-layer bi-directional Transformer pre-trained with Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks.
  - **Transformer**$_{ICT}$ [4] is designed for passage retrieval in question answering scenarios. It jointly optimizes the Inverse Cloze Task (ICT) with MLM.
  - **PROP** [26] adopts the Representative Words Prediction (ROP) task to better learn the matching between the sampled word sets and the document. In our paper, we consider both the released checkpoints PROP$_{wiki}$ and PROP$_{marco}$[2].
  - **HARP** [28] introduces the hyperlinks and anchor texts into pre-training and achieves the state-of-the-art performance on ad hoc retrieval. As no pre-trained HARP checkpoint is publicly available, we only cite its reference performance reported in the corresponding published paper.

5.1.3 Evaluation Metrics. For the two small datasets (i.e., Robust04 and MQ2007), we randomly divide all queries into five folds and then evaluate the model performance by conducting 5-fold cross-validation. The final results are obtained by averaging the performance on each testing fold. Following previous work [19, 29], we report precision at rank 20 (P@20) and normalized discounted cumulative gain at rank 20 (NDCG@20) on Robust04. As for MQ2007, we report NDCG@10 and NDCG@20 values. On TREC COVID, P@20 and NDCG@10 are used to measure the model performance. For MS MARCO and DL 2019, we follow the official instructions to use MRR@10 (MRR@100) and NDCG@10 (NDCG@100) as the evaluation metrics.

5.1.4 Implementation. For traditional models, we use the Anserini toolkit[3]. The best parameters ($k_1 = 3.8, b = 0.87$) reported for retrieval are employed for BM25. For KNRM and Conv-KNRM, we use the OpenMatch[4] toolkit as the implementation. Here the 300d GloVe [32] vectors are used to initialize the word embeddings. For BERT, we adopt the Pytorch version of the BERT-base checkpoint released by Google[5]. For PROP, two pre-trained checkpoints[6] (PROP$_{wiki}$ and PROP$_{marco}$) are directly used for fine-tuning. Finally, we implement ARES and reproduce the Transformer$_{ICT}$ model by using the popular Huggingface Transformer library[7].

In the pre-training stage, we set the length expectation $\lambda = 3$ while sampling pseudo queries. As for the MLM learning objective, we follow the masking strategy of BERT (randomly selecting 15% words in the input sequence, and the selected words will be replaced by the [MASK] token in 80% of time, by a random token in 10% of time, and unchanged in 10% of time). For each document, we generate ten pseudo queries via the contrastive sampling strategy and then randomly sample two query pairs for pre-training (Equation 11). To save the effort of training our model from scratch, BERT-base is used to initialize the parameters of ARES. We adopt the AdamW [24] optimizer with a linear warm-up rate of 0.1 to update model parameters. To support a larger batch size with limited GPU resources, we employ mixed-precision training and parallel training techniques in our implementation. The maximum input length for all models is 512. We pre-train ARES with a learning rate of 5e-5 and a batch size of 168 (28*6) for one epoch. The pre-training process normally takes about two days on six Nvidia GeForce RTX 3090 24G GPUs. To find the best checkpoint, we sample 5,000 queries from the MS MARCO training set and test the zero-shot performance of ARES checkpoint every 10k steps. The one with the best performance on these queries will be further selected for fine-tuning.

In the fine-tuning process, we train each model with supervised data and then apply them to rerank the document candidates. Generation approaches of document candidates are quite different across datasets. For Robust04, we rerank the top 200 BM25 candidates. As for TREC COVID, following OpenMatch, we use the top 60 candidates provided by the BM25-fusion method. In MQ2007, each query is officially provided with about 40 candidate documents. For MS MARCO and DL 2019, we use both the official top 100 candidates and the top 100 candidates generated by an effective dense retrieval approach named ADORE+STAR [49] (denoted as "AS" in what follows). We concatenate query text with document content and feed the input sequence ([CLS];q;[SEP];d;[SEP]) into various transformers. In MS MARCO and DL2019, we use the concatenation of the title, URL, and body as document content, which is a common practice in dealing with these two datasets. The output representations of [CLS] will be leveraged to calculate a pair-wise loss similar to Equation 11. All pre-trained models are fine-tuned with a learning rate of 1e-5 and a batch size of 320 (40*8) for 20 epochs. It takes about 100 minutes to fine-tune one epoch on eight Nvidia Tesla V100-32GB GPUs.

To facilitate the reproductivity of our results, we release the source code for our experiments as well as the pre-trained ARES checkpoints in the link below [8].

[2] We do not consider B-PROP as a baseline in this paper because 1) no pre-trained checkpoint is publicly available, 2) we focus more on judging the preference given a query pair rather than improving the query sampling strategy.
[3] https://github.com/castorini/anserini
[4] https://github.com/thunlp/OpenMatch
[5] https://github.com/google-research/bert
[6] https://github.com/Albert-Ma/PROP
[7] https://github.com/huggingface/transformers
[8] https://github.com/xuanyuan14/ARES-master

Table 4: Overall performance of ARES and other baselines on two large-scale datasets. "†" denotes the result is significantly worse than our ARES using paired t-test at $p < 0.05$ level. The best results are in bold and the second-best results are underlined. Note that for HARP, the reported metrics are *reference values* and the significance test can not be conducted.

Model Type	Model Name	MS MARCO				TREC DL 2019			
		Official Top100		AS Top100		Official Top100		AS Top100	
		MRR@10	MRR@100	MRR@10	MRR@100	nDCG@10	nDCG@100	nDCG@10	nDCG@100
Traditional Models	BM25	.2656†	.2767†	.2962†	.3107†	.5315†	.4996†	.5776†	.4795†
	QL	.2143†	.2268†	.2664†	.2819†	.5234†	.4983†	.6227†	.4981†
Neural IR Models	KNRM	.1526†	.1685†	.1721†	.1913†	.3071†	.4591†	.3427†	.4387†
	Conv-KNRM	.1554†	.1792†	.1833†	.2251†	.3112†	.4762†	.3612†	.4565†
Pre-trained Models	BERT	.3826†	.3881†	.4105†	.4197†	**.6540**	.5325	.6351	.5001†
	Transformer$_{ICT}$	.3860†	.3913†	.4113†	.4208†	.6491	.5320	.6344	.4998†
	PROP$_{wiki}$	.3866†	.3922†	.4124†	.4219†	.6399†	.5311	.6237†	.4998†
	PROP$_{marco}$	.3930†	.3980†	.4186†	.4278†	.6425†	.5318	**.6447**	.5038
	HARP	.3961	.4012	N/A	N/A	.6562	.5337	N/A	N/A
Our Approach	ARES$_{simple}$	**.3995**	**.4041**	**.4302**	**.4386**	.6505	**.5353**	.6378	**.5054**
	(ARES best)	(.3995^1)	(.4046^2)	(.4302^1)	(.4386^1)	(.6666^3)	(.5397^3)	(.6460^2)	(.5079^3)

5.2 Overall Performance

Table 4 systematically reports the performance of various models on MS MARCO and TREC DL 2019. Note that in this table, we only report the performance of the best ARES variant (ARES$_{simple}$) and the best metrics achieved by different ARES variants (denoted as "ARES best", the superscript $1/2/3$ denotes ARES$_{simple}$, ARES$_{hard}$, and ARES$_{REP}$, respectively). Through the experimental results, we have the following findings:

- All pre-trained models significantly outperform traditional and neural IR models. This indicates the effectiveness of the transformer architecture and the two-stage training paradigm. Through pre-training, these models may have learned helpful knowledge for text matching and thus perform substantially better.
- Pre-trained models tailored for IR such as PROP and HARP perform significantly better than BERT. Without a specially designed training objective, BERT is like a dilettante in terms of ranking. Based on the ROP task, PROP can better capture the matching between queries and documents than BERT. By learning the contrastive relationships buried in the Wikipedia hyperlinks, HARP is slightly superior to other PTMs on the two datasets.
- In general, ARES performs best among all the models in most metrics, even better than HARP which introduces external knowledge on Wikipedia. The best performing variant is ARES$_{simple}$ which emphasizes the RANK axioms while also considering other heuristics. The improvement on the DL 2019 dataset is not that significant because the testing set is so small (with only 43 queries). However, we find that ARES$_{REP}$ can do much better by achieving the highest nDCG@10 value of 0.6666. As the performance for most PTMs on the official top 100 candidates is relatively better than that on the AS candidates, we guess that this dataset focuses more on the exact matches between queries and documents. Therefore, ARES$_{REP}$ may have an advantage over other models by merely leveraging two REP axioms.

On the three small datasets, we also find similar trends as in MS MARCO. As shown in Table 5, ARES performs best on Robust04 and MQ2007 and achieves competitive performance on the TREC-COVID dataset. The improvement of ARES over the PROP models is not very significant on the TREC-COVID dataset. This may be because the number of training cases in TREC-COVID is huge (320k pairs) while there are only 35 testing queries. After fine-tuning, most PTMs show close performance. Therefore, evaluation results on this dataset may not be very typical.

Table 5: Overall performance of ARES and other baselines on three small datasets. "N" stands for NDCG. "†" denotes the result is significantly worse than ARES at $p < 0.05$ level using pairwise t-test.

Model Name	TREC-COVID		Robust04		MQ2007	
	P@20	N@10	P@20	N@20	N@5	N@10
BM25	.4857†	.4792†	.3670†	.4265†	.3835†	.4142†
QL	.4729†	.4683†	.3540†	.4135†	.3749†	.4033†
KNRM	.3986†	.3619†	.3408†	.3871†	.3295†	.3594†
Conv-KNRM	.4043†	.3490†	.3600†	.4140†	.3378†	.3706†
BERT	.5386	.5580†	.3855†	.4526†	.4532†	.4768†
Transformer$_{ICT}$	.5286†	.5418†	.3928†	.4590†	.4512†	.4755†
PROP$_{wiki}$	**.5429**	**.6104**	.3892†	.4604†	.4606†	.4793†
PROP$_{marco}$	.5257†	.5944	.3910†	.4644†	.4628†	.4841
ARES$_{simple}$	.5400	.5969	**.4048**	**.4810**	**.4729**	**.4901**

5.3 Low-resource Settings

As the fine-tuning process will cover the original nature of pre-trained models, we further investigate the model effectiveness in low-resource settings, i.e., zero-shot and few-shot scenarios.

5.3.1 Zero-shot performance. We compare the zero-shot performance of various transformers with BM25. In this scenario, we directly test the re-ranking performance of the off-the-shelf pre-trained models without any supervision data for fine-tuning. As revealed in Table 6, ARES$_{simple}$ significantly outperforms all other models, and it is also the only model that substantially outperforms

Table 6: Zero-shot performance of various transformers. "M", "N", and "P" stand for MRR, NDCG, and Precision. "†" denotes the result is significantly worse than ARES_{simple} at $p < 0.05$ level using pairwise t-test.

Model Name	MS MARCO		DL 2019		COVID	EQ
	M@10	M@100	N@10	N@100	P@20	P@10
BM25	.2962	.3107	.5776†	.4795†	.4857†	.6690†
BERT	.1820†	.2012†	.4059†	.4198†	.4314†	.6055†
PROP_{wiki}	.2429†	.2596†	.5088†	.4525†	.4857†	.5991†
PROP_{marco}	.2763†	.2914†	.5317†	.4623†	.4829†	.6454†
ARES_{strict}	.2630†	.2785†	.4942†	.4504†	.4786†	**.6923**
ARES_{hard}	.2627†	.2780†	.5189†	.4613†	.4943	.6822†
ARES_{simple}	**.2991**	**.3130**	**.5955**	**.4863**	**.4957**	.6916

Table 7: Ablation study on ARES variants. "†" denotes the result is significantly worse than ARES_{simple} at $p < 0.05$ level using pairwise t-test.

Variant	MS MARCO			
	Official Top100		AS Top100	
	MRR@10	MRR@100	MRR@10	MRR@100
ARES_{REP}	.3946†	.3997	.4235†	.4324†
ARES_{RANK}	.3920†	.3971†	.4159†	.4253†
ARES_{strict}	.3967	.4016	.4251†	.4339
ARES_{hard}	.3995	**.4046**	.4290	.4380
ARES_{simple}	**.3995**	.4041	**.4302**	**.4386**

BM25 across various datasets. On the contrary, other ARES counterparts also achieve competitive performances without fine-tuning, especially on the EntityQuestions (EQ) benchmark whose testing set is much larger than other datasets. This observation indicates the effectiveness of incorporating IR axioms into the pre-training process. In fact, the search domains of TREC COVID and EntityQuestions are very different from that of MS MARCO documents. According to the promising adaptive ability on these datasets, ARES variants may have learned more ubiquitous rules concerning relevance and thus own higher robustness.

5.3.2 Few-shot performance. To test the model effectiveness from various angles, we also compare the performance of ARES with PROP after fine-tuned with limited supervised data on various datasets. Here we adopt the same experimental settings as in the full-resource training. As shown in Figure 3, ARES outperforms PROP on all datasets using the same number of training queries. Note that we do not report the performance of BERT here because its few-shot performance is much worse than ARES and PROP. On TREC COVID, DL 2019 and MS MARCO, PROP needs about 1k-2k queries for training to surpass BM25 while ARES needs none. These results have shown the great potential of ARES in scenarios where limited or even no supervised data is available.

5.4 Ablation Study

To further verify the effectiveness of different axioms in ARES, we conduct an ablation study by comparing the performance among various ARES variants on the MS MARCO dataset, including ARES_{REP}, ARES_{RANK}, ARES_{strict}, ARES_{hard}, and ARES_{simple}. Among them, the latter three counterparts incorporate all axioms into pre-training while the first two only consider one group of basic axioms. The results are given in Table 7. We can observe that leveraging all axioms normally significantly improves the system performance compared to only using a proportion of them. The performance of ARES_{hard} and ARES_{simple} are rather close, slightly better than ARES_{strict}. This is reasonable because forcing the positive queries to satisfy all the constraints may be too strict. A wiser approach may be learning a classification model (i.e., a predictor) to balance the importance of each axiom in the pairwise decision process. ARES_{simple} performs overall the best among all variants. There may be two reasons: 1) leveraging the preference predictor trained on randomly sampling negative queries (setting I) may help the pre-trained model learn a more general distribution of query difference, 2) as the prediction accuracy is the highest in the setting I, using the corresponding predictor may introduce less noisy data into the pre-training process. We further find that ARES_{REP} outperforms ARES_{RANK}, especially on AS top 100 candidates. As the discriminative power of the RANK heuristic is low, totally depending on it for pre-training may cause problems in distinguishing two high-quality queries. In addition, ARES_{REP} slightly outperforms PROP, indicating the effectiveness of combining TF-IDF with QL scores.

5.5 Case Study

To analyze the difference of the ranking mechanism behind ARES and the best baseline PROP, we use Integrated Gradient (IG) [6, 38] as the interpretation method. In a nutshell, IG computes the integral of integrated gradients to show the importance of each input attribution for the output. We visualize the attribution results of ARES_{simple} and PROP on one case in the TREC DL 2019 dataset. As revealed in Figure 4, distributions of attribution attention for ARES and PROP are quite different. We can observe that positive terms are more concentrated on the front part of the document for ARES while the distribution is more scattered in the whole document-wide for PROP. This phenomenon implies that ARES pays more attention to the head content by employing the PROX-2 axiom. Besides, ARES focuses more on the bigram phrases such as "goldfish grow" and "make goldfish". Without the guidance of PROX-1, PROP usually attends to single terms. As a central query term, "goldfish" is emphasized in ARES, while PROP captures the less informative word "do" and gives a negative attribution to "goldfish". Therefore, ARES can better estimate the document's relevance (highly relevant) and thus rank the document higher.

6 CONCLUSION

In this work, we have proposed a novel pre-training method with Axiomatic Regularization for ad hoc Search, namely ARES. We first sample a set of pseudo queries via an effective and efficient contrastive sampling strategy for each document in the corpus. Then an axiomatic preference predictor (i.e., a decision tree) is built by fitting a constructed pairwise dataset. The distribution of feature importance learned in this process intuitively indicates the role each axiom plays in the decision process. We further apply the trained axiomatic predictors to judge the preference for the pseudo queries and pre-train the BERT-base model with these query

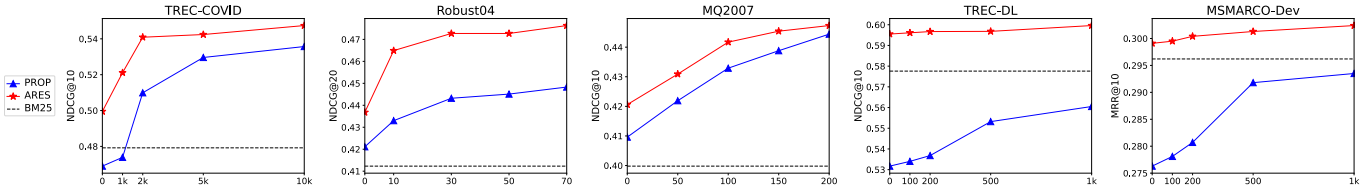


Figure 3: Few-shot performance of ARES and PROP when fine-tuned with different number of limited supervised data.

[CLS] do goldfish grow [SEP] https : / / answers . yahoo . com / question / index ? qid = 20100226170159aawholxhow to make goldfish grow faster ? " pets fish how to make goldfish grow faster ? just wondering ? update : what kind of foods could i use ? would warmer water help ? update 2 : gabe tech , retard they aren ' t in a bowl and if i did what you said , they ' d die ! follow 18 answers answers relevance rating newest oldest best answer : really people ? if you put a small child into a large house , will he grow faster ? no ! a tank that is too small will slow his growth down and even stop it but a bigger tank than needed won ' t have any effect . make sure his water is good and that he has adequate room and food , and he will grow at his own pace . really people fish are just like any other animal on the planet they aren ' t little aliens . t he only thing weird about how a fish grows is that they put out a hormone into the water that will slow down the growth of other fish and them selves . and dont put fill your bowl with juice thats an acid and it will kill your fish

(a) Without fine-tuning, ARES$_{simple}$ ranks the relevant document at the **first** position.

[CLS] do goldfish grow [SEP] https : / / answers . yahoo . com / question / index ? qid = 20100226170159aawholxhow to make goldfish grow faster ? " pets fish how to make goldfish grow faster ? just wondering ? update : what kind of foods could i use ? would warmer water help ? update 2 : gabe tech , retard they aren ' t in a bowl and if i did what you said , they ' d die ! follow 18 answers answers relevance rating newest oldest best answer : really people ? if you put a small child into a large house , will he grow faster ? no ! a tank that is too small will slow his growth down and even stop it but a bigger tank than needed won ' t have any effect . make sure his water is good and that he has adequate room and food , and he will grow at his own pace . really people fish are just like any other animal on the planet they aren ' t little aliens . t he only thing weird about how a fish grows is that they put out a hormone into the water that will slow down the growth of other fish and them selves . and dont put fill your bowl with juice thats an acid and it will kill your fish

(b) Without fine-tuning, PROP ranks the relevant document at the **14th** position.

Figure 4: Attribution results of the zero-shot performance of ARES$_{simple}$ and PROP on a TREC DL2019 case (qid: 489204, docid: D897966). The color of each term indicates the attribution value, where red is positive, blue is negative, and white is neutral. The deeper the color is, the larger the absolute value is. This figure is best viewed in color.

pairs. According to the experimental results on multiple datasets, ARES can achieve competitive ranking performance compared to existing state-of-the-art approaches. Specifically, ARES has shown promising performance in low-resource settings, i.e., it is the only pre-trained model that substantially outperforms BM25 on various benchmark datasets without any supervision signals for fine-tuning. Results of the ablation study have also implied the necessity of combining all axioms in the pre-training process. We further conduct a case study by visualizing the attribution of query and document terms at the zero-shot inferencing stage. The case study indicates that ARES has learned the desirable knowledge covered by certain axioms such as PROX-1 and PROX-2.

Our work is a primary attempt at increasing the interpretability of pre-training methods. As with any research, there are limitations to our work. These limitations may inspire interesting future directions. Firstly, there still exists an overfitting problem in the pre-training process. At this stage, we adopt the early-stop strategy with validation to alleviate this problem. In the future, more robust regularization techniques can be explored to better fix it. Secondly, although we interpret the concept of relevance by using several axioms, it is still a long way before we thoroughly figure out the oracle definition of relevance. In the future, more intensive work needs to be done on exploring the undetected aspects concerning relevance.

7 ACKNOWLEDGEMENTS

This work is supported by the Natural Science Foundation of China (Grant No. 61732008, 61902209), Tsinghua University Guoqiang Research Institute, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

REFERENCES

[1] Gianni Amati and Cornelis Joost Van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 357–389.
[2] Siddhant Arora and Andrew Yates. 2019. Investigating Retrieval Method Selection with Axiomatic Features. *arXiv preprint arXiv:1904.05737* (2019).
[3] Arthur Câmara and Claudia Hauff. 2020. Diagnosing bert with retrieval heuristics. *Advances in Information Retrieval* 12035 (2020), 605.
[4] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932* (2020).
[5] Jia Chen, Jiaxin Mao, Yiqun Liu, Ziyi Ye, Weizhi Ma, Chao Wang, Min Zhang, and Shaoping Ma. 2021. A Hybrid Framework for Session Context Modeling. *ACM Transactions on Information Systems (TOIS)* 39, 3 (2021), 1–35.
[6] Lijuan Chen, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2021. Toward the Understanding of Deep Text Matching Models for Information Retrieval. *arXiv preprint arXiv:2108.07081* (2021).
[7] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.
[8] Zitong Cheng and Hui Fang. 2020. Utilizing Axiomatic Perturbations to Guide Neural Ranking Models. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*. 153–156.

[9] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2020. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820* (2020).

[10] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 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.

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

[12] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, and Yiqun Liu. 2021. Pre-training Methods in Information Retrieval. arXiv:cs.IR/2111.13853

[13] Hui Fang, Tao Tao, and ChengXiang Zhai. 2004. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 49–56.

[14] Hui Fang, Tao Tao, and Chengxiang Zhai. 2011. Diagnostic evaluation of information retrieval models. *ACM Transactions on Information Systems (TOIS)* 29, 2 (2011), 1–42.

[15] Hui Fang and ChengXiang Zhai. 2006. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 115–122.

[16] Luyu Gao and Jamie Callan. 2021. Condenser: a Pre-training Architecture for Dense Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 981–993.

[17] Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540* (2021).

[18] Sreenivas Gollapudi and Aneesh Sharma. 2009. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*. 381–390.

[19] 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*. 55–64.

[20] Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. 2016. Axiomatic result re-ranking. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 721–730.

[21] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 39–48.

[22] Xiangsheng Li, Jiaxin Mao, Chao Wang, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. Teach machine how to read: reading behavior inspired relevance estimation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 795–804.

[23] Yiqun Liu, Chao Wang, Ke Zhou, Jianyun Nie, Min Zhang, and Shaoping Ma. 2014. From skimming to reading: A two-stage examination model for web search. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 849–858.

[24] Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam. (2018).

[25] Yuanhua Lv and ChengXiang Zhai. 2011. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*. 7–16.

[26] 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 14th ACM International Conference on Web Search and Data Mining*. 283–291.

[27] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021. B-PROP: Bootstrapped Pre-training with Representative Words Prediction for Ad-hoc Retrieval. *arXiv preprint arXiv:2104.09791* (2021).

[28] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021. 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.

[29] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1101–1104.

[30] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*.

[31] Ramith Padaki, Zhuyun Dai, and Jamie Callan. 2020. Rethinking query expansion for bert reranking. *Advances in Information Retrieval* 12036 (2020), 297.

[32] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

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

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

[35] Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

[36] Corby Rosset, Bhaskar Mitra, Chenyan Xiong, Nick Craswell, Xia Song, and Saurabh Tiwary. 2019. An axiomatic approach to regularizing neural ranking models. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 981–984.

[37] Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. *arXiv preprint arXiv:2109.08535* (2021).

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*. PMLR, 3319–3328.

[39] 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.

[40] Michael Völske, Alexander Bondarenko, Maik Fröbe, Benno Stein, Jaspreet Singh, Matthias Hagen, and Avishek Anand. 2021. Towards Axiomatic Explanations for Neural Ranking Models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*. 13–22.

[41] Ellen M Voorhees. 2005. The TREC robust retrieval track. In *ACM SIGIR Forum*, Vol. 39. ACM New York, NY, USA, 11–20.

[42] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, Kathryn Funk, Rodney Kinney, Ziyang Liu, William Merrill, et al. 2020. Cord-19: The covid-19 open research dataset. *ArXiv* (2020).

[43] Hao Wu and Hui Fang. 2012. Relation based term weighting regularization. In *European Conference on Information Retrieval*. Springer, 109–120.

[44] 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*. 55–64.

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

[46] Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Simple applications of BERT for ad hoc document retrieval. *arXiv preprint arXiv:1903.10972* (2019).

[47] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32 (2019).

[48] ChengXiang Zhai. 2008. Statistical language models for information retrieval. *Synthesis lectures on human language technologies* 1, 1 (2008), 1–141.

[49] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. *arXiv preprint arXiv:2104.08051* (2021).

[50] Wei Zheng and Hui Fang. 2010. Query aspect based term weighting regularization in information retrieval. In *European Conference on Information Retrieval*. Springer, 344–356.

[51] Yujia Zhou, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2021. PSSL: Self-supervised Learning for Personalized Search with Contrastive Sampling. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2749–2758.