Leveraging Estimated Transferability Over Human Intuition for Model Selection in Text Ranking

Jun Bai[1] **Zhuofan Chen**[1] **Zhenzi Li**[1] **Hanhua Hong**[2]
Jianfei Zhang[1] **Chen Li**[1] **Chenghua Lin**[2] **Wenge Rong**[1]
[1] School of Computer Science and Engineering, Beihang University, China
[2] Department of Computer Science, University of Manchester, United Kingdom
{ba1_jun, zhuofanchen, zhenzil, zhangjf, chen.li, w.rong}@buaa.edu.cn
hanhua.hong@postgrad.manchester.ac.uk, chenghua.lin@manchester.ac.uk

Abstract

Text ranking has witnessed significant advancements, attributed to the utilization of dual-encoder enhanced by Pre-trained Language Models (PLMs). Given the proliferation of available PLMs, selecting the most effective one for a given dataset has become a non-trivial challenge. As a promising alternative to human intuition and brute-force fine-tuning, Transferability Estimation (TE) has emerged as an effective approach to model selection. However, current TE methods are primarily designed for classification tasks, and their estimated transferability may not align well with the objectives of text ranking. To address this challenge, we propose to compute the expected rank as transferability, explicitly reflecting the model's ranking capability. Furthermore, to mitigate anisotropy and incorporate training dynamics, we adaptively scale isotropic sentence embeddings to yield an accurate expected rank score. Our resulting method, Adaptive Ranking Transferability (AiRTran), can effectively capture subtle differences between models. On challenging model selection scenarios across various text ranking datasets, it demonstrates significant improvements over previous classification-oriented TE methods, human intuition, and ChatGPT with minor time consumption.

1 Introduction

Ranking relevant documents based on user queries has gained prominence as an essential component for Information Retrieval (IR) (Li et al., 2024), Retrieval-Augmented Generation (RAG) (Cuconasu et al., 2024), etc. As the volume of online content continues to grow, developing efficient and effective text ranking techniques has become increasingly imperative. Fortunately, the advancements in Pre-trained Language Models (PLMs) have spurred the development of dual-encoder capable of meeting these demands (Zhao et al., 2024). Usually, practitioners select the PLM to be employed by their intuition. However, such decision-making becomes unreliable as the number of PLMs rapidly grows (Lin et al., 2024), introducing a new challenge: the accurate and automatic selection of the most potent PLM to achieve the desired performance on given datasets, namely Model Selection (MS) (Bai et al., 2023b).

Early approaches to MS adopted a brute-force method, fine-tuning each model to identify the optimal one (Choi et al., 2020), which suffers from huge computing costs. Hence, it is preferable to estimate model fine-tuning result with computational efficiency. In pursuit of this goal, Transferability Estimation (TE) has garnered increasing attention in recent years, merely leveraging model-encoded features and labels (Agostinelli et al., 2022). Currently, advances in TE have primarily been developed for classification tasks (Bassignana et al., 2022). These methods approximate the log-likelihood from features to labels, aligning with the classification objectives well and demonstrating promising results on classification datasets (Bai et al., 2023b). However, the text ranking task fundamentally differs from classification task in which the matching scores for relevant query-document pairs are expected to be higher than those for irrelevant pairs (Luan et al., 2021). As a result, applying current classification-oriented TE methods directly to text ranking is sub-optimal (Bai et al., 2023a).

Typically, a PLM well-suited to a dataset is expected to generate well-distributed sentence embeddings, where matched pairs exhibit high matching scores and vice versa, even before fine-tuning (Gao et al., 2021). Hence, the expected rank of documents relevant to queries resulted from initial model can serve as explicit and consistent indicators of model fine-tuning performance in text ranking, in contrast to the classification-oriented TE methods. However, a challenge arises from the anisotropy that usually exists in PLMs from small size to large size, manifesting as entangled

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 12356–12374

feature dimensions and a biased distribution of sentence embeddings (Liu et al., 2024). In this case, even if two sentences are very similar in semantics, their vector representations may be far apart or exhibit directional differences. As a consequence, it leads to inaccurate document rank (Su et al., 2021) and hinders the alignment of expected rank scores with ranking transferability. In solving the anisotropy problem, previous research has proposed isotropization methods such as BERT-flow (Li et al., 2020), BERT-whitening (Su et al., 2021), and SimCSE (Gao et al., 2021), aiming to transform sentence embeddings into an isotropic space. Considering BERT-flow and SimCSE both need careful training (Gao et al., 2021), we select BERT-whitening in this work since it only involves efficiently whitening the sentence embeddings. However, the isotropization overlooks the training dynamics involved in adapting model to the text ranking task (Sasaki et al., 2023), hindering the full expression of the fine-tuning ranking capabilities. To address this issue, we introduce an adaptive scaling mechanism after obtaining isotropic sentence embeddings, aiming to simulate the training dynamics by a simple but effective scaling operation, whose scaling weight can be efficiently solved by the least square method (Ding, 2023). Then, our approach allows for a more precise reflection of the model's true transferability through improved expected rank scores.

In summary, we present a promising alternative to human intuition for MS in text ranking: the **A**dapt**i**ve **R**anking **T**ransferability (**AiRTran**). In this approach, the expected rank is computed and further enhanced by the proposed **Ada**ptive **Iso**tropization (**AdaIso**) to estimate ranking transferability. On five widely studied datasets with two challenges pools of small and large candidate PLMs, covering various text ranking scenarios and dual-encoder backbones, our method shows superior efficiency compared to brute-force fine-tuning and show promising performance compared to previous classification-oriented TE methods, human intuition, and ChatGPT.

2 Related Work

Early text ranking methods were primarily based on bag-of-words functions (Aizawa, 2003; Robertson and Zaragoza, 2009), which fall short of comprehensively understanding semantic relevance. In recent years, with the advancements in PLMs, dual-encoder has demonstrated superiority in capturing semantic interaction that goes beyond simple word overlap (Zhao et al., 2024). Prior research on dual-encoder has predominantly focused on improving aspects such as sentence representations (Gao and Callan, 2021; Tang et al., 2022; Wu et al., 2023), negative document sampling (Xiong et al., 2021; Formal et al., 2022; Chen et al., 2023), matching score computation (Khattab et al., 2021; Lassance et al., 2022; Wang et al., 2023), and supervisory signals construction (Wang et al., 2021; Zeng et al., 2022; Lin et al., 2023), etc.

Unlike the above research focus on improving the network structure or training approach, our work explores selecting the most powerful PLM to achieve superior fine-tuning performance. The most direct approaches to MS involve either exhaustively trying all available models (Ni et al., 2022), or training a performance predictor to estimate how well a given model will perform on a dataset (Zhang et al., 2023; Meng et al., 2023), while they are less practical due to the large demands for model training. To address this issue, several TE methods have been developed for efficient MS, primarily in classification tasks (Bai et al., 2023b). These methods, when provided with model-encoded features and corresponding labels, quantify the degree of compatibility between them, treating this as a measure of transferability. Subsequently, they rank candidate models based on their transferability scores to identify the best-performing model. Some of these approaches are grounded in the assumption that an effective model should produce features with a high degree of class separability (Puigcerver et al., 2021; Kumari et al., 2022; Pándy et al., 2022; Xu and Kang, 2023). Another line of research focuses on the approximation of likelihood from features to labels (Bao et al., 2019; You et al., 2021; Pándy et al., 2022; Huang et al., 2022; Shao et al., 2022; Ding et al., 2022). Inspired by these remarkable works, this paper focuses on the selection of text ranking model which is an underexplored and challenging area.

3 Preliminaries

Before delving into the details of our proposed method, we begin by providing foundational knowledge, including the problem definition, dual-encoder model, and an examination of the challenges that previous TE methods have faced.

3.1 Problem Definition

In the context of MS for text ranking, we are presented with a pool of M candidate PLMs, denoted as $\{\phi_i\}_{i=1}^{M}$, and a dataset $\mathcal{D} = \{(q_i, d_i, y_i)\}_{i=1}^{N}$ containing N pairs of (query q_i, document d_i), where the label y_i is 1 if d_i is relevant to q_i and 0 otherwise. To evaluate the performance of MS, we are also provided with the fine-tuning performances $\{T_i(\mathcal{D})\}_{i=1}^{M}$ of all candidate PLMs on $\mathcal{D}$ (e.g., the R@10 score). The goal of MS is to assign scores to all candidate PLMs, denoted as $\{S_i(\mathcal{D})\}_{i=1}^{M}$, which can well correlate with $\{T_i(\mathcal{D})\}_{i=1}^{M}$, allowing us to select the best-performing model.

3.2 Dual-encoder

The dual-encoder is commonly employed to rank desired results from large-scale candidate documents, which consists of PLM-based query and document encoders. The query q and document d are mapped into sentence embeddings by conducting mean pooling on the last layer's outputs of corresponding encoders ($e_q = \phi(q)$ and $e_d = \phi(d)$), where $e_q \in \mathbb{R}^{1\times D}$, $e_d \in \mathbb{R}^{1\times D}$, and D represents the dimension of the embeddings. Subsequently, the matching score between the query and the document is computed by the dot-product[1] between their sentence embeddings ($e_q e_d^T$).

For each query, the training objective of the dual-encoder is to ensure that its matching scores for relevant pairs are higher than those for irrelevant pairs. To achieve this goal, the probability of the relevant document d^+ is typically optimized:

$$p(d^+|q, d^+, \mathcal{I}_q) = \frac{\exp(e_q e_{d^+}^T)}{\exp(e_q e_{d^+}^T) + \sum_{d^-\in\mathcal{I}_q} \exp(e_q e_{d^-}^T)} \quad (1)$$

where d^+ is from $\mathcal{R}_q$ that is the set of documents relevant to q, d^- is the document irrelevant to q, and $\mathcal{I}_q$ is the set of documents irrelevant to q.

3.3 Classification-oriented TE Methods

Previous research has been devoted to MS for classification tasks. Given the PLM-encoded feature f and label y, these methods estimate the expected log conditional probabilities, denoted as $\mathbb{E}[\log(p(y|f))]$. This objective aligns with the training objective of the classification, establishing consistency between the estimated transferability and model fine-tuning performance.

[1]There are alternatives such as cosine and Euclidean distance, we select dot-product due to its ease of use.

However, text ranking involves a more intricate training objective, and merely estimating $\mathbb{E}[\log(p(y|e_q, e_d))]$ may not adequately reflect the model ranking capabilities. To illustrate this point, let's consider a scenario where we have a query q with one relevant document d^+ and two irrelevant documents d_1^- and d_2^-. Suppose the corresponding matching probabilities from ϕ_1 for d^+, d_1^-, and d_2^- are 0.5, 0.45, and 0.45, respectively. In this case, $\mathbb{E}[\log(p(y|q, d))]$ is -0.82, and the rank of d^+ is 1. Now, consider another PLM, ϕ_2, where the matching probabilities for d^+, d_1^-, and d_2^- are 0.6, 0.65, and 0.1. In this case, the expectation is -0.72, but the rank of d^+ is 2. Interestingly, while ϕ_2 has a higher transferability score, it exhibits poorer ranking ability than ϕ_1, underscoring the necessity of estimating transferability in a manner that is more aligned with text ranking performance.

4 Methodology

As illustrated in Figure 1, we introduce the AiR-Tran approach, motivated by the observation that expected rank score is inherently well-aligned with text ranking performance. To further improve this alignment, we propose AdaIso to mitigate the anisotropy problem and adapt raw sentence embeddings to the downstream text ranking task. As a result, the resultant expected rank effectively captures the true ranking transferability of the model. In the following sections, we will first describe how to compute expected rank and then elaborate on how AdaIso further achieves the enhancement.

4.1 Expected Rank as Transferability

The computation of expected rank score necessitates the determination of matching scores between queries and candidate documents. Fortunately, the dual-encoder can readily perform inference once initialized by a PLM, allowing for the direct acquisition of matching scores and resultant expected rank score. Consequently, the computation incurs minimal cost, merely involving a simple forward propagation process of the PLM on the dataset. After obtaining sentence embeddings and corresponding matching scores, we can further compute the ranks of documents relevant to queries and also the expected rank (as Eq. 2):

$$S(\mathcal{D}) = \mathbb{E}_{q\sim p(q)}[\mathbb{E}_{d_+\in\mathcal{R}_q, d_-\in\mathcal{I}_q}[\frac{1}{\text{rank}(d_+)}]] \quad (2)$$

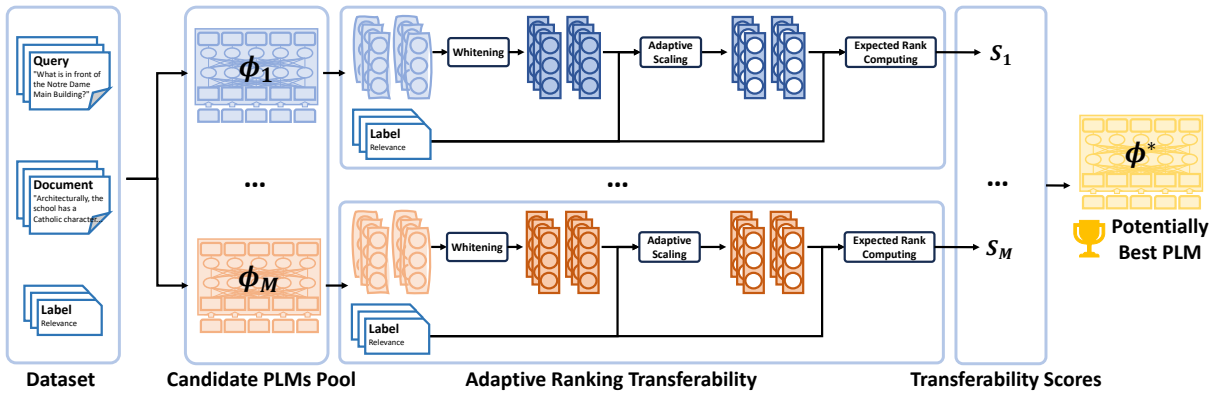


Figure 1: This is the pipeline of model selection in text ranking using AiRTran. First, the queries and documents are encoded to sentence embeddings by each candidate model ϕ. Then, the raw embeddings are transformed by whitening and adaptive scaling sequentially. Finally, the transformed embeddings coupled with labels are used to compute the expected rank as transferability, resulting in the selection of the best-performing model.

4.2 Adaptative Isotropization

Due to the inherent disparity between the pre-training task and the text ranking task, sentence embeddings encoded by PLMs often exhibit anisotropy issue that renders them unsuitable for direct use in text ranking. Anisotropy manifests as severe entanglement among the feature dimensions and distortion of the sentence embeddings' distribution, leading to matching scores that inadequately capture sentence semantics (Li et al., 2020). When working with these perturbed sentence embeddings, the computed expected rank score fails to genuinely reflect the model transferability.

Numerous approaches have been introduced to address this issue, such as BERT-flow (Li et al., 2020), SimCSE (Gao et al., 2021), and BERT-whitening (Su et al., 2021). BERT-flow and SimCSE both necessitate careful model training, where BERT-flow involves the training of a flow-based calibration model, and SimCSE optimizes the contrastive objective to achieve isotropization. However, these methods run counter to the objective of efficient MS, and their training is also unstable. Hence, in this work, we opt for BERT-whitening as our primary isotropization method, which exclusively post-processes sentence embeddings through an efficient whitening operation, bypassing the need for model training (Huang et al., 2021).

In BERT-whitening, the mean sentence embedding $\vec{\mu} \in \mathbb{R}^{1\times D}$ (Eq. 3) and the covariance matrix $\Sigma \in \mathbb{R}^{D\times D}$ (Eq. 4) are initially estimated based on all sentence embeddings $E \in \mathbb{R}^{2N\times D}$:

$$\vec{\mu} = \frac{1}{2N}\vec{1}_{2N}^T E \tag{3}$$

$$\Sigma = \frac{1}{2N-1}(E-\vec{1}_{2N}\vec{\mu})^T(E-\vec{1}_{2N}\vec{\mu})+\epsilon I_D \tag{4}$$

where $\vec{1}_{2N} \in \mathbb{R}^{2N\times 1}$ is the $2N$-dimensional ones vector, $I_D \in \mathbb{R}^{D\times D}$ is the identity matrix, and $\epsilon > 0$ is a small positive number to prevent a singular Σ. Then the sentence embeddings are centered and transformed as Eq. 5:

$$\hat{E} = (E - \vec{1}_{2N}\vec{\mu})U\sqrt{\Lambda^{-1}} \tag{5}$$

where $\Sigma = U\Lambda U^T$, $\hat{E}$ is the whitened embeddings.

However, isotropization does not account for the adaptation of sentence embeddings to the downstream task, which restricts its ability to accurately reveal the true ranking performance of the model. To address this limitation, we propose an adaptive scaling mechanism partially simulating the task adaptation by further scaling the isotropic sentence embeddings, expressed as $\hat{E} \odot (\vec{1}_{2N}\vec{\gamma}^T)$, where $\odot$ denotes the Hadamard product, and $\vec{\gamma} \in \mathbb{R}^{D\times 1}$ represents the scaling weight. The crucial question is how to determine the optimal $\vec{\gamma}^*$. Fortunately, this weight vector can be derived by solving an ordinary least squares problem that minimizes the squared difference between predicted matching scores and ground truth labels. To begin, given the whitened sentence embeddings for all queries ($\hat{E}_q \in \mathbb{R}^{N\times D}$) and for all documents ($\hat{E}_d \in \mathbb{R}^{N\times D}$), the predicted matching scores $\hat{Y} \in \mathbb{R}^{N\times 1}$ can be calculated as:

$$\begin{aligned}\hat{Y} &= (\hat{E}_q \odot (\vec{1}_N\vec{\gamma}^T)) \odot (\hat{E}_d \odot (\vec{1}_N\vec{\gamma}^T))\vec{1}_D \\ &= \vec{1}_N(\vec{\gamma}^2)^T \odot \hat{E}_q \odot \hat{E}_d\vec{1}_D \\ &= (\hat{E}_q \odot \hat{E}_d)\vec{\gamma}^2\end{aligned} \tag{6}$$

where $\vec{1}_N \in \mathbb{R}^{N\times 1}$ and $\vec{1}_D \in \mathbb{R}^{D\times 1}$. Next, given all ground truth labels $Y \in \mathbb{R}^{N\times 1}$, the squared

difference between $\hat{Y}$ and Y is:

$$\begin{aligned}\mathcal{L}(\vec{\gamma}) &= (\hat{Y}-Y)^T(\hat{Y}-Y) \\ &= \hat{Y}^T\hat{Y} - 2\hat{Y}^TY + Y^TY \\ &= (\vec{\gamma}^2)^T\hat{E}_m^T\hat{E}_m\vec{\gamma}^2 - 2(\vec{\gamma}^2)^T\hat{E}_m^TY \\ &\quad + Y^TY\end{aligned} \tag{7}$$

where $\hat{E}_m = \hat{E}_q \odot \hat{E}_d$. Then, by minimizing $\mathcal{L}(\vec{\gamma})$, we can obtain the optimal $\vec{\gamma}^*$:

$$\vec{\gamma}^* = \arg\min_{\vec{\gamma}} \mathcal{L}(\vec{\gamma}) \tag{8}$$

This can be efficiently solved by setting the partial derivative equal to zero:

$$\frac{\partial \mathcal{L}(\vec{\gamma})}{\partial \vec{\gamma}^2} = 2\hat{E}_m^T\hat{E}_m\vec{\gamma}^2 - 2\hat{E}_m^TY = 0 \tag{9}$$

$$\vec{\gamma}^* = \sqrt{(\hat{E}_m^T\hat{E}_m)^{-1}\hat{E}_m^TY} \tag{10}$$

Ultimately, utilizing the sentence embeddings $\hat{E} \odot (\vec{1}_{2N}\vec{\gamma}^{*T})$ enhanced by adaptive isotropization, we can achieve more precise matching scores, which in turn lead to the computation of improved expected rank scores $\{S_i(\mathcal{D})\}_{i=1}^M$.

5 Experiments

5.1 Datasets

To validate the AiRTran, the experiments are conducted on five datasets: SQuAD (Rajpurkar et al., 2016), NQ (Kwiatkowski et al., 2019), BioASQ (Bai et al., 2023a), SciFact (Wadden et al., 2020), and MuTual (Cui et al., 2020), covering typical text ranking scenarios (For dataset details, see Appendix A). To conduct TE, we randomly sample 1,000 queries from the training set of each dataset. The irrelevant documents for each query are then randomly sampled, since constructing irrelevant documents by human annotation or other sophisticated methods can be time-consuming, while this simple strategy also introduces a variable number of samples. Increasing the number of irrelevant documents usually creates a more challenging scenario, effectively highlighting differences in the ranking abilities of models. As a result, our experiments are carried out with varying sizes of candidate documents (consisting of one relevant document and multiple irrelevant documents) for each query, ranging from 2 to 10.

5.2 Candidate Model Pools

We have curated two model pools from Hugging Face's model hub (Jiang et al., 2023), consisting of 25 small PLMs (11M~140M parameters) and 25 large PLMs (1B~8B parameters), respectively. We fully fine-tuned the small PLMs and parameter-efficiently fine-tuned the large ones to obtain their fine-tuning results. Each fine-tuning is conducted using 5 different random seeds, and the results are averaged. Subsequently, the best fine-tuning performances are recorded to form $\{T_i(\mathcal{D})\}_{i=1}^M$ (We record R@10 for SQuAD and NQ since they are text retrieval tasks that focus on recall performance, and record P@1 for BioASQ, SciFact, and MuTual since they are text matching tasks that focus on matching accuracy.) The training details and fine-tuning results are presented in Appendix B.

5.3 Evaluation Metric

Given the target of MS, when S_i is higher than S_j, the corresponding T_i is expected to be higher than T_j. To evaluate TE approaches, we employ Kendall's τ to measure the degree of agreement between $\{S_i(\mathcal{D})\}_{i=1}^M$ and $\{T_i(\mathcal{D})\}_{i=1}^M$. It is particularly valuable when working with ordinal or ranked data, without any assumptions about the underlying data distribution. The formula for calculating Kendall's τ is as follows:

$$\tau = \frac{2\sum_{1\le i\le j\le M}\mathbb{I}(T_i - T_j)\mathbb{I}(S_i - S_j))}{M(M-1)} \tag{11}$$

where $\tau \in [-1, 1]$; the indicator function $\mathbb{I}$ returns 1 for positive number and return -1 otherwise. For each TE method, we compute τ_{Small}, τ_{Large} that denote the Kendall's τ computed when T corresponds to the fine-tuning results of small model pool and large model pool, respectively.

5.4 Compared Approaches

We compare the following competitive TE methods which have proven effective when facing classification tasks: TMI (Xu and Kang, 2023), GBC (Pándy et al., 2022), TransRate (Huang et al., 2022), $\mathcal{N}$LEEP (Li et al., 2021), LinearProxy (Kumari et al., 2022), SFDA (Shao et al., 2022), PACTran (Ding et al., 2022), H-score (Bao et al., 2019), LogME (You et al., 2021), whose details are presented in Appendix C. We run each method using 5 random seeds and report averaged Kendall's τ. Note that these methods assume each data sample corresponds to one feature vector. To adapt

Methods	SQuAD		NQ		BioASQ		SciFact		MuTual	
	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}
TMI	0.252	0.332	0.275	0.379	0.143	0.356	0.496	0.439	0.255	0.248
GBC	0.375	0.513	0.529	0.496	0.180	0.555	0.416	0.541	0.080	0.383
TransRate	0.592	0.680	0.327	0.408	0.313	0.315	0.147	0.520	0.172	0.180
$\mathcal{N}$LEEP	0.552	0.436	0.516	0.456	0.303	0.363	0.559	0.328	0.112	0.448
LinearProxy	0.454	0.477	0.359	0.546	0.339	0.394	0.629	0.551	0.229	0.577
SFDA	0.639	0.597	0.623	0.593	0.441	0.411	0.693	0.498	0.229	0.503
PACTran	0.643	0.656	0.661	0.668	0.395	0.567	0.681	0.571	0.164	0.564
H-score	0.629	0.663	0.608	0.627	0.435	0.596	0.696	**0.617**	0.223	0.513
LogME	0.645	**0.720**	0.621	0.660	0.417	0.583	0.695	0.549	0.125	0.528
AiRTran	**0.667**	0.706	**0.672**	**0.674**	**0.508**	**0.648**	**0.785**	0.600	**0.351**	**0.583**

Table 1: The best τ of all TE methods over different document sizes, where the highest scores are underlined.

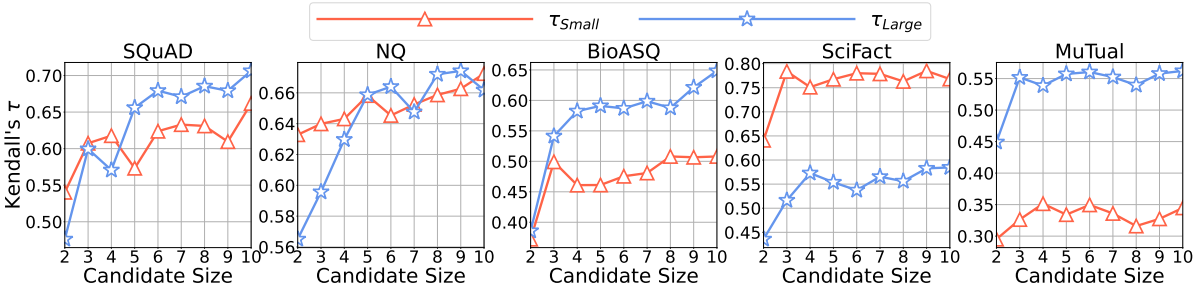


Figure 2: The performance variations of AiRTran over different document sizes.

these methods for text ranking, where each sample consists of a pair of sentence embeddings, we explored Hadamard product, element-wise addition, element-wise subtraction, and concatenation, to combine two sentence embeddings into a single feature vector. We found that all methods achieved their best performance when utilizing the Hadamard product operation[2].

5.5 Model Selection Performance

Table 1 presents the best results of all methods. The anisotropy issue leads to substantial overlap between features of matched and mismatched pairs, which hinders the accurate computation of Gaussian distance and entropy, resulting in inferior performances for TMI, GBC, and TransRate. The methods that simulate training dynamics including LinearProxy, SFDA, PACTran, and LogME exhibit strong performance. H-score also demonstrates notable improvements, attributed to its consideration of feature redundancy in addition to the inter-class variance. Note that $\mathcal{N}$LEEP involves the feature transformation as well, while it doesn't consider the label information, thus resulting in poor performance. Nevertheless, these classification-oriented methods generally trail behind AiRTran due to their lack of alignment with the text ranking ability. With the inclusion of AdaIso which adaptively scales the isotropic sentence embeddings, AiRTran generates expected rank that strongly correlates with model fine-tuning performance. Notably, it achieves the highest τ in 8 out of 10 cases, providing strong validation of its effectiveness. For additional results and specific predictions of AiRTran, please see Appendix D and E.

5.6 Effect of Candidate Document Size

As depicted in Figure 2, we visualize the performance variations of AiRTran, and the visualization results of other methods are presented in Appendix F. It is observed that TMI, TransRate, and $\mathcal{N}$LEEP show significant fluctuation. The other methods indicate relatively regular patterns of performances with the candidate document size increases, while they behave differently when facing different model pools. Both of the observations are attributed to the lack of alignment between their estimation score and model ranking ability.

[2]For implementation details, our code is available at https://github.com/Ba1Jun/model-selection-AiRTran.

Methods	SQuAD		NQ		BioASQ		SciFact		MuTual	
	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}
AiRTran (flow)	0.662	0.630	0.652	0.538	0.451	0.602	0.764	**0.663**	0.349	0.440
AiRTran (simcse)	0.635	0.657	**0.676**	0.602	**0.576**	0.606	**0.811**	0.605	0.320	0.453
AiRTran (whiten)	**0.667**	**0.706**	0.672	**0.674**	0.508	**0.648**	0.785	0.600	**0.351**	**0.583**
w/o Iso	0.646	0.419	0.636	0.449	0.431	0.377	0.720	0.505	0.297	0.448
w/o Ada	0.655	0.688	0.645	0.647	0.235	0.580	0.698	0.556	0.116	0.226
w/o AdaIso	0.561	0.319	0.554	0.398	0.219	0.346	0.484	0.421	0.223	0.406

Table 2: The results of AiRTran when BERT-flow (flow), SimCSE (simcse), and BERT-whitening (whiten) are employed, respectively. The results when different AdaIso components are removed are also listed.

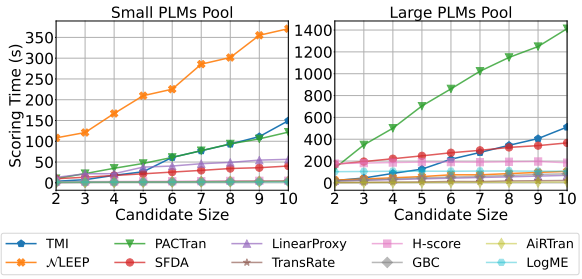


Figure 3: This is the comparison between the time consumption of all methods as the size of candidate documents grows. Note that the encoding time for dataset is not included, since it is shared by all methods.

In contrast, AiRTran generates expected rank score that can harness the benefits of including more irrelevant documents, thereby enhancing the differentiation of ranking capabilities among different models. Consequently, on both small model pool and large model pool, AiRTran's performance generally exhibits an upward trend as the candidate document size increases. This brings convenience to the determination of its hyper-parameters, the desired performance usually can be achieved when candidate document size is set to 10.

5.7 Scoring Time Comparison

The scoring time, i.e., the time required for each TE method to score the transferabilities of all models given the model-encoded features, serves as a reflection of the efficiency of model selection. In Figure 3, we have visualized the trends in scoring time for all TE methods on two pools as the candidate document size increases. Broadly, there exists a linear relationship between the scoring time of all TE methods and the size of the candidate document set. Though the scoring time for all TE methods is significantly lower than the time required for the brute-force fine-tuning (In our case, fine-tuning all PLMs on all datasets took nearly three months using a single NVIDIA GeForce RTX 3090 24G.), the proposed AiRTran consistently operates with a fast runtime, thanks to the efficient whitening operation and the solution to the scaling weight.

5.8 Exploration of AdaIso

Though AiRTran is agnostic to isotropization method, we select whitening due to its efficacy. To verify this choice, we also instantiate AiRTran by BERT-flow and SimCSE, i.e., AiRTran (flow) and AiRTran (simcse), and compare them with AiRTran (whiten). As shown in Table 2, since the training of BERT-flow and SimCSE both need to be carefully tuned, the resultant AiRTran performances are unstable. Specifically, AiRTran (flow) performs best only in the SciFact with the large model. Though AiRTran (simcse) performs well in several datasets (e.g., NQ with small model and SciFact with small model), but overall, it is slightly outperformed by the whitening variant. In contrast, AiRTran (whiten) consistently shows the highest performance across most cases, generally being the most effective instantiation. Moreover, AiRTran (whiten) requires only seconds of CPU runtime to perform the whitening, whereas the other methods require costly GPU resources. We also conducted an ablation study for AiRTran (whiten) by removing isotropization (w/o Iso), adaptive scaling (w/o Ada), and full AdaIso (w/o AdaIso), the results revealed that both adaptive scaling and isotropization play a crucial role, where removing any of them led to a significant drop. Additionally, our observations highlight that isotropization does not consistently yield improvements, e.g., "w/o Ada" performed worse than "w/o AdaIso" on MuTual, primarily due to its unsupervised nature, underscoring the necessity of adaptive scaling.

Methods	SQuAD		BioASQ	
	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}
$\mathcal{Q}$Tran	0.437	0.393	0.267	0.393
w Ada	**0.600**	0.580	0.347	0.573
w Iso	0.467	0.447	0.312	0.520
w AdaIso	0.567	**0.673**	**0.450**	**0.580**

Table 3: The Kendall's τ performance of $\mathcal{Q}$Tran when different components of AdaIso are employed.

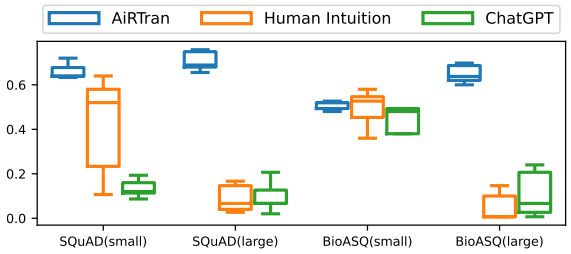


Figure 4: The comparison of Kendall's τ between AiRTran, human intuition, and ChatGPT.

5.9 Relation to Alignment and Uniformity

We further use the following properties to investigate the inner workings of AdaIso: (1) Alignment: It quantifies the expected affinity between sentence embeddings of paired sentences. (2) Uniformity: It measures how uniformly the embeddings of different sentences are distributed (Wang and Isola, 2020). These two properties align well with the objective that matched pairs should remain close in embedding space, while embeddings for random instances should be widely scattered in the context of text ranking. As Eq. 12, we combine them to form a quality score, denoted as $\mathcal{Q}$.

$$\mathcal{Q} = \underbrace{\mathop{\mathbb{E}}_{(q,d^+)\sim P_{pos}} e_q e_{d^+}^T}_{Alignment} + \underbrace{\mathop{\mathbb{E}}_{(x,x')\sim P_{data}} -e_x e_{x'}^T}_{Uniformity} \tag{12}$$

Intuitively, a powerful model should exhibit high alignment and uniformity scores. Consequently, $\mathcal{Q}$ should exhibit a strong correlation with the model's fine-tuning results, serving as a valuable indicator for transferability, i.e., Quality score as Transferability ($\mathcal{Q}$Tran). However, this correlation is challenging to fully establish when the anisotropy problem persists, and training dynamics are not considered. Fortunately, the proposed AdaIso can solve this problem. As demonstrated in Table 3, it is observed that $\mathcal{Q}$Tran computed on raw sentence embeddings only exhibits a weak correlation with model fine-tuning results. After whitening isotropization and adaptive scaling are conducted on the raw sentence embeddings, the correlations improve significantly, facilitating more accurate MS. We also observed that when isotropization is conducted, the further employment of adaptive scaling doesn't always bring improvement (i.e., τ_{Small} in SQuAD), probably because the sentence embeddings lose too much original information after whitening and adaptive scaling weight is over-fitted on such perturbed features.

5.10 Comparison with Human and ChatGPT

Currently, the most prevalent method for MS often relies on human intuition or, alternatively, prompts to ChatGPT (OpenAI, 2022). Although such processes can be completed in a few minutes, the corresponding performances become unreliable when facing challenging MS scenarios. To demonstrate AiRTran's potential as a viable replacement for these methods, we conducted a comparative study among AiRTran, human intuition, and ChatGPT on SQuAD and BioASQ datasets. For MS by human intuition, we engaged 5 NLP practitioners (doctoral and master's students specializing in NLP) who were provided with metadata about the dataset and models. Similarly, ChatGPT was guided using instructions that utilized the same information (see Appendix G), and we parsed the ranking results of candidate PLMs from its responses. This process was repeated 5 times to capture the inherent variability in ChatGPT's responses. The results, depicted in Figure 4, emphasize the underperformance of human intuition and ChatGPT in terms of relevance and stability. This is due to their limited understanding of the dataset and models. In contrast, AiRTran excels in demonstrating transferability by effectively capturing the compatibility between model-encoded features and labels, thus showcasing a significant advantage.

6 Conclusion

Given the limitations of classification-oriented TE methods, we propose the use of AiRTran in this study. It demonstrates superior MS performance compared to competitive TE methods, human experts, and the ChatGPT MS agent, indicating its potential as a solution for MS in text ranking tasks. We hope that our work will offer text ranking practitioners valuable guidance and inspiration when selecting models for their datasets of interest in addition to their intuition.

Acknowledgements

This work was supported by the Natural Science Foundation of China (No. 62377002).

Limitations

To construct irrelevant documents for each query, this study employs a random sampling strategy. While easy to implement and efficient in generating irrelevant pairs, it often results in documents that are too easy to discriminate. In such an uncompetitive ranking scenario, both proficient and average models may perform well and achieve high transferability scores, complicating model selection. Therefore, there is a pressing need to explore more effective strategies for negative sampling in MS tasks. Additionally, although human experts and ChatGPT exhibit lower performance compared to AiRTran, their knowledge demonstrates a sophisticated understanding of model-to-dataset transfer. However, this study does not investigate the potential effectiveness of integrating their insights as a complement to the feature-label interactions captured by TE approaches.

References

Andrea Agostinelli, Michal Pándy, Jasper R. R. Uijlings, Thomas Mensink, and Vittorio Ferrari. 2022. How stable are transferability metrics evaluations? In *Proceedings of the 17th European Conference on Computer Vision*, pages 303–321.

Akiko N. Aizawa. 2003. An information-theoretic perspective of tf-idf measures. *Information Processing & Management*, 39(1):45–65.

Jun Bai, Chuantao Yin, Zimeng Wu, Jianfei Zhang, Yanmeng Wang, Guanyi Jia, Wenge Rong, and Zhang Xiong. 2023a. Improving biomedical ReQA with consistent NLI-transfer and post-whitening. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(3):1864–1875.

Jun Bai, Xiaofeng Zhang, Chen Li, Hanhua Hong, Xi Xu, Chenghua Lin, and Wenge Rong. 2023b. How to determine the most powerful pre-trained language model without brute force fine-tuning? An empirical survey. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 5369–5382.

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas J. Guibas. 2019. An information-theoretic approach to transferability in task transfer learning. In *Proceedings of the 2019 IEEE International Conference on Image Processing*, pages 2309–2313.

Elisa Bassignana, Max Müller-Eberstein, Mike Zhang, and Barbara Plank. 2022. Evidence $>$ textgreater intuition: Transferability estimation for encoder selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4218–4227.

Patrick H. Chen, Wei-Cheng Chang, Jyun-Yu Jiang, Hsiang-Fu Yu, Inderjit S. Dhillon, and Cho-Jui Hsieh. 2023. FINGER: Fast inference for graph-based approximate nearest neighbor search. In *Proceedings of the ACM Web Conference 2023*, pages 3225–3235.

Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2020. Evaluation of BERT and ALBERT sentence embedding performance on downstream NLP tasks. In *Proceedings of the 25th International Conference on Pattern Recognition*, pages 5482–5487.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for RAG systems. *CoRR*, abs/2401.14887.

Leyang Cui, Yu Wu, Shujie Liu, Yue Zhang, and Ming Zhou. 2020. Mutual: A dataset for multi-turn dialogue reasoning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1406–1416.

Feng Ding. 2023. Least squares parameter estimation and multi-innovation least squares methods for linear fitting problems from noisy data. *Journal of Computational and Applied Mathematics*, 426:115107.

Nan Ding, Xi Chen, Tomer Levinboim, Soravit Changpinyo, and Radu Soricut. 2022. PACTran: PAC-Bayesian metrics for estimating the transferability of pretrained models to classification tasks. In *Proceedings of the 17th European Conference on Computer Vision*, pages 252–268.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural IR models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2353–2359.

Luyu Gao and Jamie Callan. 2021. Is your language model ready for dense representation fine-tuning? *CoRR*, abs/2104.08253.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Junjie Huang, Duyu Tang, Wanjun Zhong, Shuai Lu, Linjun Shou, Ming Gong, Daxin Jiang, and Nan Duan. 2021. WhiteningBERT: An easy unsupervised sentence embedding approach. In *Findings of the*

Association for Computational Linguistics: EMNLP 2021, pages 238–244.

Long-Kai Huang, Junzhou Huang, Yu Rong, Qiang Yang, and Ying Wei. 2022. Frustratingly easy transferability estimation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 9201–9225.

Wenxin Jiang, Nicholas Synovic, Matt Hyatt, Taylor R. Schorlemmer, Rohan Sethi, Yung-Hsiang Lu, George K. Thiruvathukal, and James C. Davis. 2023. An empirical study of pre-trained model reuse in the hugging face deep learning model registry. In *Proceedings of the 45th IEEE/ACM International Conference on Software Engineering*, pages 2463–2475.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for OpenQA with ColBERT. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2022. Ensembling off-the-shelf models for GAN training. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10641–10652.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Carlos Lassance, Maroua Maachou, Joohee Park, and Stéphane Clinchant. 2022. Learned token pruning in contextualized late interaction over BERT (ColBERT). In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2232–2236.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 9119–9130.

Yandong Li, Xuhui Jia, Ruoxin Sang, Yukun Zhu, Bradley Green, Liqiang Wang, and Boqing Gong. 2021. Ranking neural checkpoints. In *Proceedings of the 2021 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2673.

Yongqi Li, Nan Yang, Liang Wang, Furu Wei, and Wenjie Li. 2024. Learning to rank in generative retrieval. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pages 8716–8723.

Haowei Lin, Baizhou Huang, Haotian Ye, Qinyu Chen, Zihao Wang, Sujian Li, Jianzhu Ma, Xiaojun Wan, James Zou, and Yitao Liang. 2024. Selecting large language model to fine-tune via rectified scaling law. *CoRR*, abs/2402.02314.

Zhenghao Lin, Yeyun Gong, Xiao Liu, Hang Zhang, Chen Lin, Anlei Dong, Jian Jiao, Jingwen Lu, Daxin Jiang, Rangan Majumder, and Nan Duan. 2023. PROD: Progressive distillation for dense retrieval. In *Proceedings of the ACM Web Conference 2023*, pages 3299–3308.

Bo Liu, Li-Ming Zhan, Zexin Lu, Yujie Feng, Lei Xue, and Xiao-Ming Wu. 2024. How good are llms at out-of-distribution detection? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 8211–8222.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Fanqing Meng, Wenqi Shao, Zhanglin Peng, Chonghe Jiang, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2023. Foundation model is efficient multimodal multitask model selector. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems*.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-T5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the Association for Computational Linguistics: ACL*, pages 1864–1874.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

Michal Pándy, Andrea Agostinelli, Jasper R. R. Uijlings, Vittorio Ferrari, and Thomas Mensink. 2022. Transferability estimation using bhattacharyya class separability. In *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9162–9172.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, Cédric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. 2021. Scalable transfer learning with expert models. In *Proceedings of the 9th International Conference on Learning Representations*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

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

Shota Sasaki, Benjamin Heinzerling, Jun Suzuki, and Kentaro Inui. 2023. Examining the effect of whitening on static and contextualized word embeddings. *Information Processing and Management*, 60(3):103272.

Wenqi Shao, Xun Zhao, Yixiao Ge, Zhaoyang Zhang, Lei Yang, Xiaogang Wang, Ying Shan, and Ping Luo. 2022. Not all models are equal: Predicting model transferability in a self-challenging fisher space. In *Proceedings of the 17th European Conference on Computer Vision*, volume 13694, pages 286–302.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *CoRR*, abs/2103.15316.

Zhengyang Tang, Benyou Wang, and Ting Yao. 2022. DPTDR: Deep prompt tuning for dense passage retrieval. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1193–1202.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7534–7550.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9929–9939.

Xiao Wang, Craig MacDonald, Nicola Tonellotto, and Iadh Ounis. 2023. ColBERT-PRF: Semantic pseudo-relevance feedback for dense passage and document retrieval. *ACM Transactions on the Web*, 17(1):3:1–3:39.

Yanmeng Wang, Jun Bai, Ye Wang, Jianfei Zhang, Wenge Rong, Zongcheng Ji, Shaojun Wang, and Jing Xiao. 2021. Enhancing dual-encoders with question and answer cross-embeddings for answer retrieval. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2306–2315.

Xing Wu, Guangyuan Ma, Meng Lin, Zijia Lin, Zhongyuan Wang, and Songlin Hu. 2023. Contextual masked auto-encoder for dense passage retrieval. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 4738–4746.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *Proceedings of the 9th International Conference on Learning Representations*.

Huiwen Xu and U Kang. 2023. Fast and accurate transferability measurement by evaluating intra-class feature variance. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision*, pages 11440–11448.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. LogME: Practical assessment of pre-trained models for transfer learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12133–12143.

Hansi Zeng, Hamed Zamani, and Vishwa Vinay. 2022. Curriculum learning for dense retrieval distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1979–1983.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. 2023. Model spider: Learning to rank pre-trained models efficiently. In *Proceedings of 37th Annual Conference on Neural Information Processing Systems*.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

A Datasets

The dataset used in this research are:

SQuAD The Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) is a widely-used benchmark for evaluating machine comprehension and passage retrieval. Created by researchers at Stanford University, SQuAD consists of over 100,000 questions derived from a diverse set of Wikipedia articles. Each question is paired with a specific passage from the article. The dataset is designed to test a model's ability to understand context and extract relevant information accurately. SQuAD has been instrumental in advancing the field of natural language processing, providing a rigorous standard for comparing the performance of various passage retrieval models. It has led to significant developments in neural network architectures and training techniques[3].

NQ The Natural Questions (NQ) (Kwiatkowski et al., 2019) dataset is a large-scale corpus created to facilitate research in question answering and passage retrieval. Developed by Google, it consists of real anonymized queries posed by users to the Google search engine, along with corresponding answers derived from Wikipedia articles. The dataset contains over 300,000 questions, with each

[3] https://rajpurkar.github.io/SQuAD-explorer

entry featuring the original query, a long answer (typically a paragraph), and a short answer (usually a span of text). The long answers provide context, while the short answers aim to pinpoint precise information. This dataset is particularly valuable for training and evaluating models in tasks such as machine comprehension, information retrieval, and automated question answering[4].

BioASQ The BioASQ (Bai et al., 2023a) dataset is a comprehensive resource aimed at advancing the field of biomedical question answering. It is part of the larger BioASQ challenge, which focuses on creating systems capable of understanding and answering complex biomedical queries. The dataset comprises a vast collection of questions generated by biomedical experts, along with corresponding answers extracted from scientific literature, including PubMed abstracts and full-text articles. The questions range from simple factoid and list questions to more complex ones requiring detailed and precise answers. The BioASQ dataset promotes the development of advanced natural language processing models that can handle the intricacies of biomedical terminology and information. It is instrumental in pushing forward research in biomedical information retrieval, question answering, and semantic indexing. We select version of 9b which has 5,828 training question-answer pairs, 496 test queries, and 31,682 candidate answers[5].

SciFact The SciFact (Wadden et al., 2020) dataset is a specialized corpus designed to facilitate research in scientific claim verification and fact-checking. Developed by the Allen Institute for AI, SciFact contains a collection of scientific claims and corresponding evidence sentences extracted from peer-reviewed biomedical research papers. The dataset includes over 1,400 claims, each accompanied by supporting or refuting evidence, enabling the development and evaluation of models that can assess the veracity of scientific statements. SciFact aims to address the challenges of verifying complex, technical claims and provides a valuable resource for advancing NLP models in the context of scientific discourse. This dataset is instrumental in promoting the development of AI systems capable of critical evaluation in scientific research[6].

MuTual The MuTual (Cui et al., 2020) dataset is a benchmark designed to evaluate dialogue systems, particularly in the context of multi-turn reasoning. Created by researchers at Tsinghua University, MuTual consists of over 8,800 dialogues, which are sourced from real-world Chinese student English listening comprehension exams and then translated into English. Each dialogue includes a series of turns between participants, with a final turn containing four potential responses. The task is to choose the most appropriate response based on the preceding conversation. The MuTual dataset emphasizes the need for models to grasp context, manage dialogue coherence, and understand nuanced interactions. It serves as a critical resource for advancing dialogue systems and enhancing their capability to engage in meaningful, coherent, and contextually appropriate conversations[7].

B Fine-tuning Candidate Models

In this work, we construct two candidate model pools of different model sizes to meet different computing needs, formed by 25 small PLMs and 25 large PLMs widely used. We employ full fine-tuning and parameter-efficient fine-tuning for small PLMs and large PLMs, respectively, considering their common fine-tuning strategy. Specifically, the training batch size is set to 32, and fine-tuning is terminated if there is no improvement in validation performance for 3 consecutive epochs, with a maximum of 10 training epochs. We select AdamW (Loshchilov and Hutter, 2019) as optimizer, and we mainly tune the learning rate, which varies within [5e-6, 1e-5, 2e-5, 3e-5, 4e-5, 5e-5], as it significantly impacts the fine-tuning performance. The official model names of these models presented in Hugging Face's model hub and their fine-tuning results on each dataset are listed in Tables 6.

C Details of Compared Methods

TMI It views transferability as the generalization of a pre-trained model on a target task by measuring intra-class feature variance using conditional entropy (Xu and Kang, 2023).

GBC It models the in-class features by Gaussian distribution, then computes the inter-Gaussian distance as the inter-class overlap. Then it selects model by the assumption that smaller overlap results in greater transferability (Pándy et al., 2022).

[4] https://ai.google.com/research/NaturalQuestions
[5] http://participants-area.bioasq.org/datasets
[6] https://leaderboard.allenai.org/scifact/submissions/public
[7] https://nealcly.github.io/MuTual-leaderboard

Methods	SQuAD		NQ		BioASQ		SciFact		MuTual	
	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}	τ_{Small}	τ_{Large}
TMI	14.2	2.0	5.6	2.0	6.0	8.0	7.0	2.0	**10.0**	10.0
GBC	16.8	**1.0**	4.0	**1.0**	6.0	6.0	2.8	2.0	24.0	1.0
TransRate	24.0	23.0	23.0	19.6	**2.0**	24.0	21.0	2.8	25.0	25.0
$\mathcal{N}$LEEP	8.6	10.0	3.4	4.4	4.6	5.2	4.4	3.2	21.6	**1.0**
LinearProxy	2.0	6.2	3.0	7.6	14.2	1.4	**1.8**	5.8	13.8	1.8
SFDA	3.0	**1.0**	4.0	**1.0**	5.0	22.6	2.4	2.0	23.2	4.0
PACTran	3.4	2.2	3.0	3.0	5.0	9.0	3.0	3.8	23.0	3.8
H-score	2.8	2.0	3.0	3.0	5.0	7.0	2.0	2.4	23.2	5.0
LogME	3.6	2.2	3.0	3.0	5.0	11.0	2.8	2.0	24.0	**1.0**
AiRTran	**1.2**	2.6	**2.6**	1.8	6.0	**1.2**	2.2	**1.8**	22.8	**1.0**

Table 4: All TE methods' best mean estimated rank results of the best-performing model over different document sizes, where the highest scores are underlined.

TransRate It estimates the mutual information between model-encoded features and labels as transferability, where the challenge of mutual information estimation is overcome by resorting to coding rate (Huang et al., 2022).

$\mathcal{N}$LEEP It fits a Gaussian Mixture Model (GMM) to model-encoded features, then computes the likelihood of labels given the GMM-produced posterior cluster assignment to measure the fine-tuning performance (Li et al., 2021).

LinearProxy It assumes a logistic classification model is on the top of frozen pre-trained model, then uses the cross-validation performance of such model as transferability (Kumari et al., 2022).

SFDA It first embeds the static features into a Fisher space and refines them for better separability. Then, it uses a self-challenging mechanism to compute the log-loss as transferability, which encourages different pre-trained models to differentiate on hard examples (Shao et al., 2022).

PACTran It seeks an optimal yet efficient PAC-Bayesian bound to the generalization error in a transfer learning setting, and the error is based on the cross-entropy loss between the prediction and the labels, measuring the generalization gap of model (Ding et al., 2022).

H-score Based on statistical and information-theoretic principles, it shows that the expected log-loss of using a model-encoded feature to predict the label of a given task under the probabilistic model can be characterized by an analytical expression, serving as model transferability (Bao et al., 2019).

Methods	AirTran
TMI	1.83e-24
GBC	1.31e-11
TransRate	3.63e-13
$\mathcal{N}$LEEP	1.26e-12
LinearProxy	3.43e-09
SFDA	2.44e-04
PACTran	2.78e-02
H-score	4.13e-02
LogME	2.90e-02

Table 5: The p-values of the t-test conducted on the results of AirTran and other model selection methods.

LogME It calculates the maximum evidence to measure a model's performance on a new task. The process involves extracting features from the pre-trained model, assuming a linear model for the new task, and computing the log marginal likelihood to assess feature suitability (You et al., 2021).

D Additional Model Selection Results

To validate the improvements, we conduct significance test between the results of AiRTran and other methods. Table 5 shows the p-values of the t-test for the results in Table 1, it is observed that all p-values are less than 0.05.

We also provide all methods' mean estimated rank results of the best-performing model. As shown in Table 4, we can observe that AiRTran can also rank the best model at the top in most cases, achieving 5 out of 10 best performances.

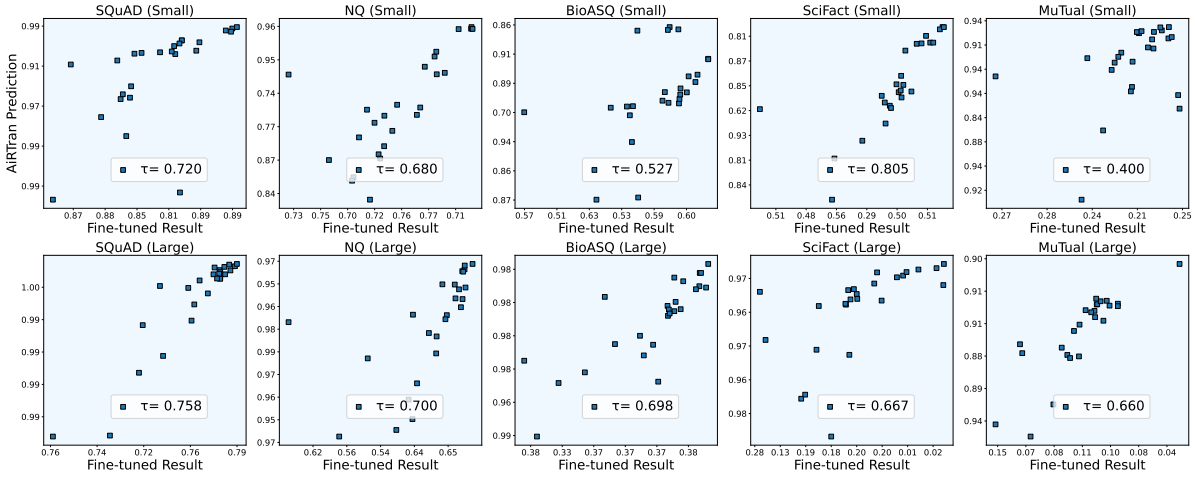


Figure 5: The predictions of AiRTran against the fine-tuning results with the best Kendall's τ performance.

E The Predictions of AiRTran

The best-performing predictions of AiRTran against the models' fine-tuning results on SQuAD, NQ, BioASQ, SciFact, and MuTual across runs are visualized in Figure 5.

F Performance Variations of TE Methods

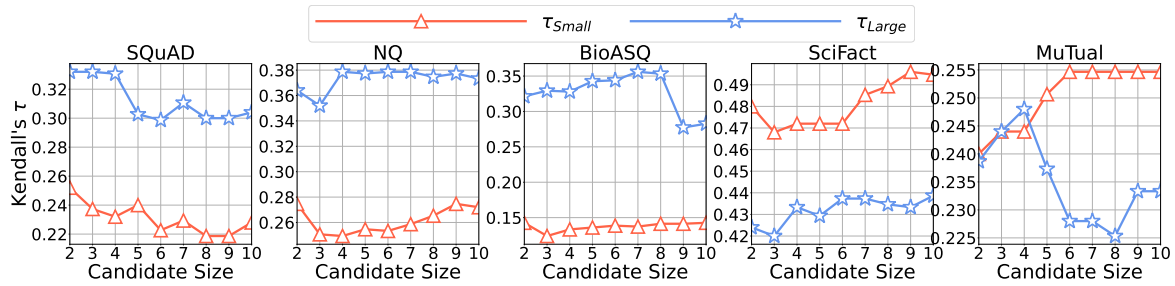
The performance variations of all TE methods across varying candidate sizes are as depicted in Figures 6 and 7.

G Selection by Human and ChatGPT

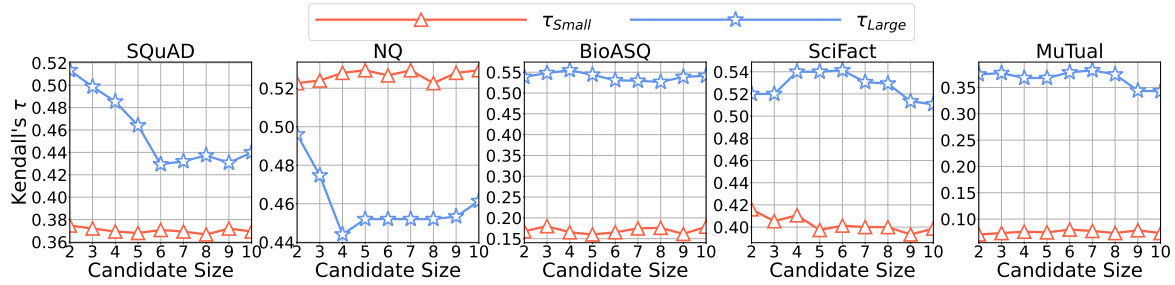
The instruction to guide human experts and ChatGPT to conduct model selection is illustrated in Table 7. Table 8 and Table 9 present the examples of dataset meta-information and model meta-information listed in the instruction, respectively. For ChatGPT, all the experiments are conducted 5 times using ChatGPT-4o.

PLMs	SQuAD	NQ	BioASQ	SciFact	MuTual
Small PLMs (Full Fine-tuning)					
bert-base-uncased	0.888	0.766	0.588	0.515	0.262
bert-base-cased	0.874	0.733	0.572	0.505	0.268
roberta-base	0.884	0.751	0.508	0.476	0.277
biobert-base-cased-v1.1	0.851	0.704	0.633	0.565	0.244
electra-base-discriminator	0.807	0.717	0.525	0.285	0.212
unsup-simcse-bert-base-uncased	0.888	0.764	0.591	0.498	0.252
sup-simcse-bert-base-uncased	0.887	0.766	0.603	0.513	0.275
openai-gpt	0.848	0.714	0.483	0.480	0.195
bart-base	0.899	0.656	0.563	0.521	0.285
scibert_scivocab_cased	0.850	0.703	0.617	0.527	0.222
scibert_scivocab_uncased	0.856	0.727	0.614	0.581	0.232
distilbert-base-cased	0.863	0.727	0.567	0.459	0.249
distilbert-base-uncased	0.883	0.757	0.589	0.512	0.255
ernie-2.0-base-en	0.897	0.772	0.590	0.547	0.284
distilroberta-base	0.882	0.737	0.517	0.449	0.268
distilgpt2	0.818	0.686	0.350	0.047	0.093
distilbert-base-multilingual-cased	0.858	0.720	0.513	0.462	0.225
albert-base-v2	0.837	0.708	0.516	0.385	0.188
PubMedBERT-base-uncased	0.856	0.723	0.633	0.609	0.245
BioLinkBERT-base	0.853	0.724	0.600	0.619	0.243
sentence-t5-base	0.887	0.754	0.461	0.293	0.229
gte-base	0.920	0.791	0.574	0.650	0.295
contriever	0.916	0.782	0.524	0.596	0.266
e5-base	0.923	0.792	0.571	0.641	0.288
bge-base-en	0.919	0.793	0.587	0.654	0.297
Large PLMs (Parameter-efficient Fine-tuning)					
falcon-rw-1b	0.763	0.632	0.380	0.234	0.089
SGPT-1.3B-mean-nli	0.764	0.622	0.378	0.279	0.149
cosmo-1b	0.737	0.565	0.329	0.126	0.065
gpt-neo-1.3B	0.720	0.543	0.367	0.194	0.078
TinyDolphin-2.8-1.1b	0.764	0.643	0.368	0.179	0.105
opt-1.3b	0.786	0.649	0.369	0.201	0.099
Qwen1.5-1.8B	0.750	0.583	0.378	0.200	0.078
bloom-1b7	0.635	0.128	0.267	0.010	0.036
deepseek-coder-1.3b-base	0.598	0.362	0.140	0.021	0.037
pythia-1.4b	0.764	0.596	0.418	0.291	0.090
phi-1_5	0.774	0.592	0.388	0.178	0.095
TinyLlama-1.1B	0.773	0.651	0.370	0.183	0.097
Yuan2-2B-hf	0.703	0.483	0.352	0.091	0.074
stable-code-3b	0.696	0.496	0.344	0.188	0.069
Apollo-2B	0.590	0.494	0.194	0.121	0.071
gemma-2b	0.527	0.447	0.236	0.099	0.043
phi-2	0.796	0.638	0.428	0.240	0.091
EMO-2B	0.641	0.508	0.323	0.186	0.059
rocket-3B	0.708	0.620	0.284	0.249	0.105
stablelm-3b-4e1t	0.752	0.642	0.431	0.321	0.086
gemma-7b	0.405	0.278	0.160	0.150	0.019
chatglm3-6b	0.762	0.566	0.417	0.370	0.093
Meta-Llama-3-8B	0.800	0.672	0.420	0.371	0.090
Qwen1.5-7B	0.757	0.645	0.412	0.298	0.089
Mistral-7B-v0.3	0.783	0.649	0.392	0.357	0.082

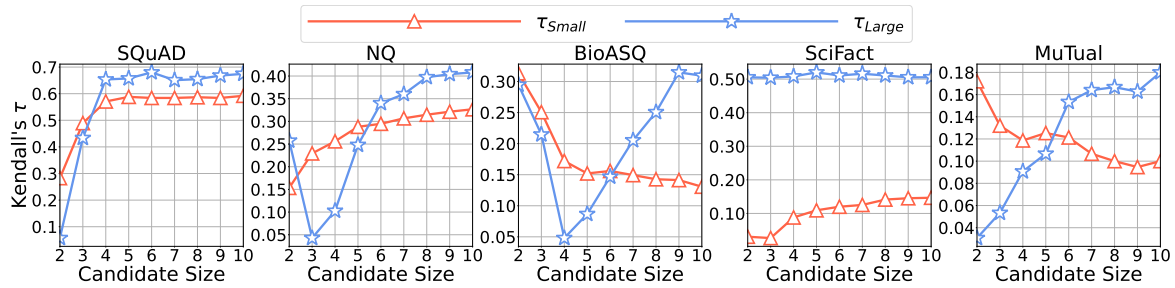
Table 6: The best fine-tuning performance of selected PLMs.



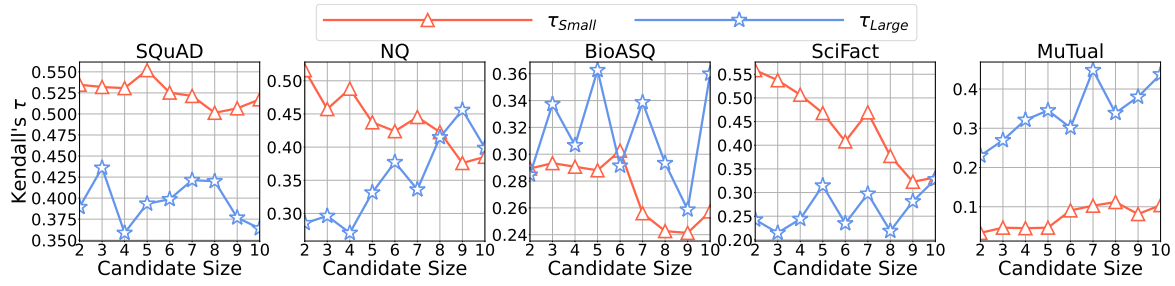
(a) TMI



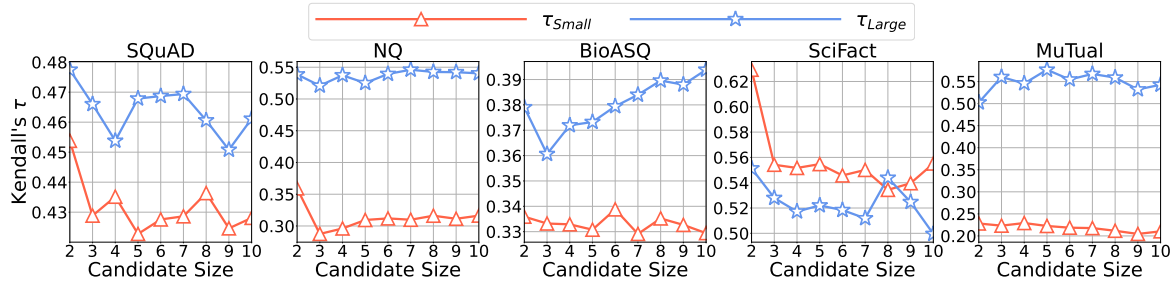
(b) GBC



(c) TransRate

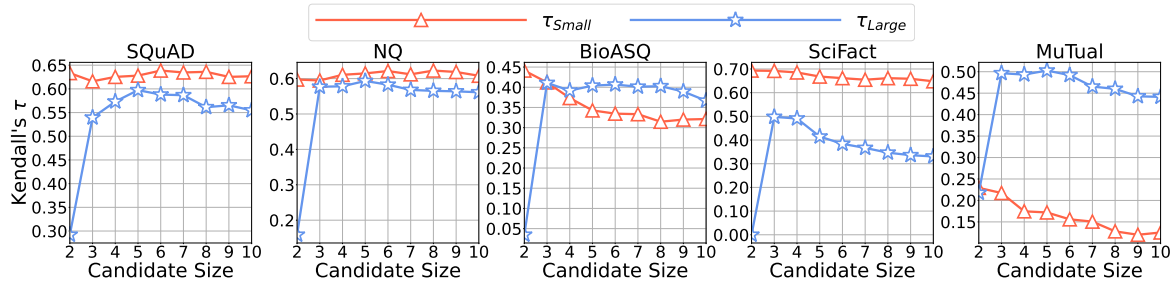


(d) $\mathcal{N}$LEEP

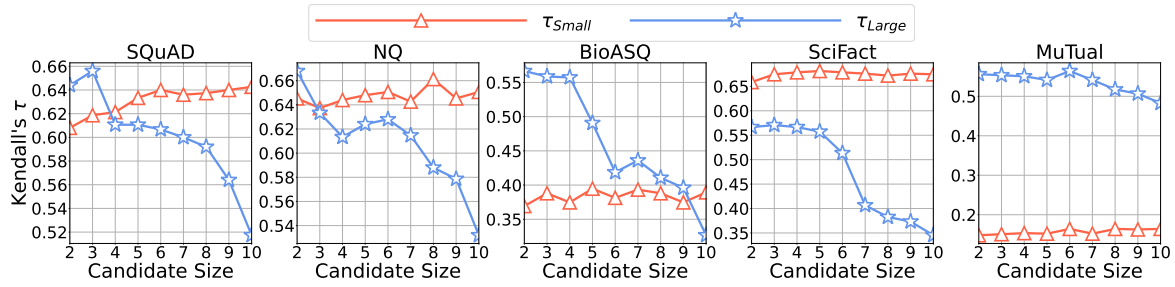


(e) LinearProxy

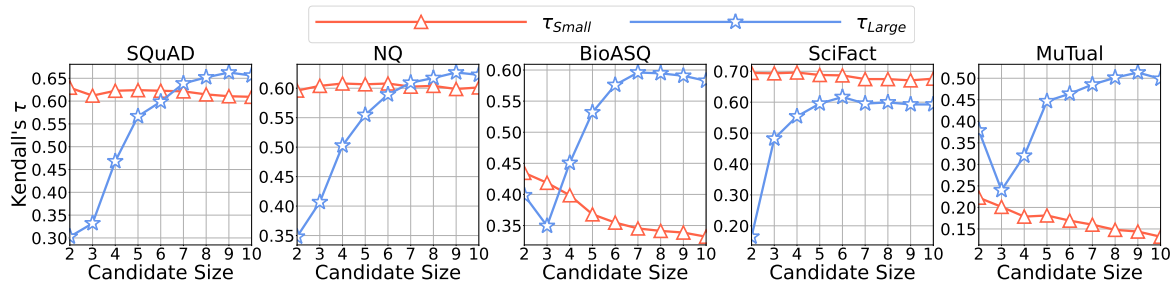
Figure 6: The performance variations of TMI, GBC, TransRate, $\mathcal{N}$LEEP, and LinearProxy over different sizes of candidate documents on SQuAD, NQ, BioASQ, SciFact, and MuTual datasets.



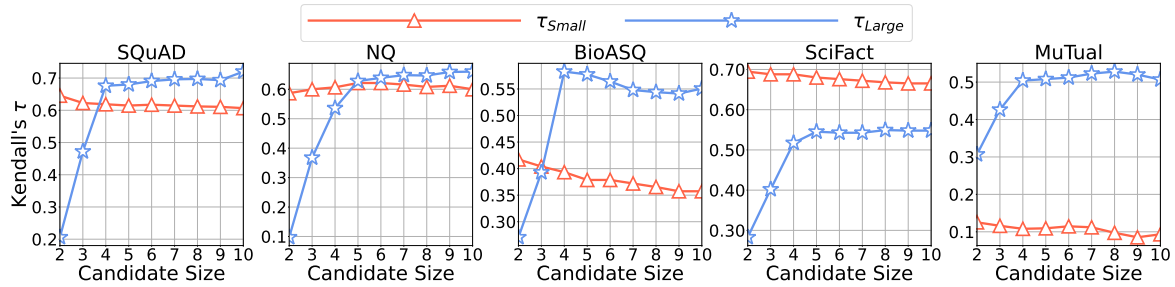
(a) SFDA



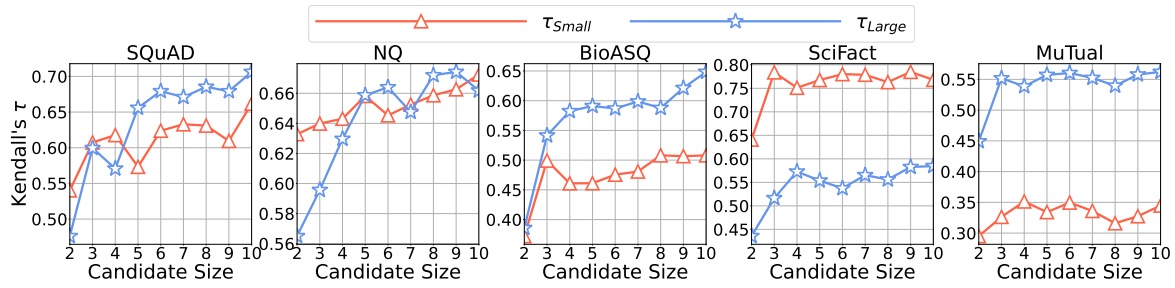
(b) PACTran



(c) H-score



(d) LogME



(e) AiRTran

Figure 7: The performance variations of SFDA, PACTran, H-score, LogME, and AiRTran over different sizes of candidate documents on SQuAD, NQ, BioASQ, SciFact, and MuTual datasets.

—— Instruction for Model Selection ——

Background:
The number of open-access Pre-trained Language Models (PLMs) is growing fast in recent years. When a dataset is given, one usually doesn't know which PLM should be selected to achieve the best fine-tuning performance. Since fine-tuning all PLMs to solve this problem is not practical, there is a need of efficient model selection.

Instructions:
You will receive the detailed meta-information for the specified dataset. Following that, you will receive the meta-information for each candidate model in the model pool. Your task is to score and rank these models based on their suitability for the given dataset, using the provided meta-information as the basis for your decisions, where the score is between 0 and 1, the higher the score, the better the fine-tuning performance of the model. After ranking the models, you must provide a detailed explanation for your rankings, citing specific aspects of the meta-information that influenced your decisions.

The dataset meta-information is:
...

The model meta-information is:
(1) ...
(2) ...
...

Output Format Requirement is:
You must output ranking results in the form of JSON to represent the order of the candidate models. The JSON format should be:

```
{
    "dataset_info": {
        "dataset_name": "Dataset Name",
        "description": "Brief description of the dataset."
    },
    "model_ranking": [
        {
            "rank": 1,
            "model": "Model Name",
            "score": "Score Value",
            "explanation": "Detailed explanation for the ranking."
        },
        {
            "rank": 2,
            "model": "Model Name",
            "score": "Score Value",
            "explanation": "Detailed explanation for the ranking."
        },
        {
            "rank": 3,
            "model": "Model Name",
            "score": "Score Value",
            "explanation": "Detailed explanation for the ranking."
        },
        ...
    ]
}
```

Table 7: Instruction for model selection using ChatGPT.

—— Meta-information of BioASQ Dataset ——
Dataset Name: BioASQ.
Description: BioASQ is designed to facilitate research in biomedical answer retrieval by transforming existing BioASQ datasets into a format suitable for ReQA tasks...
Task: Given a question, the target is to retrieve corresponding answer from answer candidates.
Source: Question-answer pairs annotated on PubMed by biomedical experts.
Data Size: 3742 questions and 5828 answers in train set, 496 questions and 31682 candidates in test set.
Language: English.
Metric: P@1, Precision at 1, a measure of how often the top-ranked result is correct.
Benchmark Performance: The P@1 of Dual-BioBERT is 57.92, ...
Related Papers: [1] Jun Bai, Chuantao Yin, Zimeng Wu, Jianfei Zhang, Yanmeng Wang, Guanyi Jia, Wenge Rong, and Zhang Xiong. 'Improving Biomedical ReQA With Consistent NLI-Transfer and Post-Whitening.' IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 20, no. 3, 2023.
Data Sample Cases: "Question 1": "Are gut microbiota profiles altered by irradiation?", "Answer 1": "Yes, Irradiation profoundly impacted gut microbiota profiles" ...

Table 8: The example of dataset's meta-information.

—— Meta-information of LLaMA-3-8B ——
Model Name: meta-llama/Meta-Llama-3-8B.
Model Structure:
Auto-regressive language model, decoder-only structure, 32 layers, grouped-query attention, context length is 8k, hidden size is 4096, totally 8B parameters.
Pre-training Details: Pre-training by causal language modeling on Over 15T tokens from publicly available sources, including a mix of high-quality non-English data to cover over 30 languages.
Model Performance: The Meta-Llama 3-8B model demonstrates impressive performance across various benchmarks, significantly surpassing several competing models. On the MMLU 5-shot benchmark, Meta-Llama 3-8B scored 66.6, outperforming Mistral 7B by 3.1 points and Gemma 7B by 2.3 points, and achieving a remarkable 20.9-point improvement over Llama 2-7B. In the AGIEval English 3-5 shot test, it achieved 45.9, which is 4.2 points higher than Gemma 7B and notably surpasses Llama 2-7B's performance. On the BIG-Bench Hard 3-shot, CoT benchmark, Meta-Llama 3-8B scored 61.1, showing a substantial improvement of 23.1 points over Llama 2-13B and outperforming Mistral 7B by 5.1 points. In the ARC-Challenge 25-shot test, Meta-Llama 3-8B achieved a score of 78.6, outperforming Gemma 7B by 25.4 points and Llama 2-13B by 11.0 points. Lastly, on the DROP 3-shot, F1 benchmark, Meta-Llama 3-8B scored 58.4, which is 4.0 points higher than Gemma 7B and significantly outperforms Llama 2-7B by 20.5 points. These results underscore Meta-Llama 3-8B's advanced capabilities and superior performance in natural language processing tasks compared to its predecessors and similar models.

Table 9: The example of model's meta-information.