Evaluating Transferability in Retrieval Tasks: An Approach Using MMD and Kernel Methods

Mengyu Dai[1]* Amir Hossein Raffiee[2] Aashish Jain[2] Joshua Correa[2]
[1]Microsoft [2]Salesforce

Abstract

Retrieval tasks play central roles in real-world machine learning systems such as search engine, recommender system, and retrieval-augmented generation (RAG). Achieving decent performance in these tasks often requires fine-tuning various pretrained models on specific datasets and selecting the best candidate, a process that can be both time and resource consuming. To tackle the problem, we introduce a novel and efficient method, called RetMMD, that leverages Maximum Mean Discrepancy (MMD) and kernel methods to assess the transferability of pretrained models in retrieval tasks. RetMMD is calculated on pretrained model and target dataset without any fine-tuning involved. Specifically, given some query, we quantify the distribution discrepancy between relevant and irrelevant document embeddings, by estimating the similarities within their mappings in the fine-tuned embedding space through kernel method. This discrepancy is averaged over multiple queries, taking into account the distribution characteristics of the target dataset. Experiments suggest that the proposed metric calculated on pretrained models closely aligns with retrieval performance post fine-tuning. The observation holds across a variety of datasets, including image, text, and multi-modal domains, indicating the potential of using MMD and kernel methods for transfer learning evaluation in retrieval scenarios. In addition, we also design a way of evaluating dataset transferability for retrieval tasks, with experimental results demonstrating the effectiveness of the proposed approach.

1. Introduction

Developing transfer learning evaluation metrics for retrieval tasks is of great importance in machine learning and information retrieval. Reliable metrics enable assessing the effectiveness of transfer learning models in retrieval-based applications, such as search engines, recommendation systems, and Retrieval Augmented Generation (RAG) with Large Language Models (LLMs) [1, 24, 39, 53]. For example, integrating retrieval mechanisms into LLMs allows them to access and leverage external knowledge sources, significantly enhancing their ability to provide accurate, up-to-date and contextually relevant responses. In these contexts, the goal is often to retrieve the most relevant items from a large dataset given a specific query or user profile. A well-designed transfer learning evaluation metric can help assess whether the learned representations from a pretrained model are effectively capturing the underlying semantics of the data and improving retrieval performance. Additionally, such a metric can facilitate comparison between different models or different fine-tuning strategies, thus guiding researchers and practitioners in model selection and further optimization.

Existing transfer learning evaluation metrics primarily focus on classification tasks [3, 37, 40, 50, 54]. Although classification and retrieval tasks are often used together and can be treated as complementary tasks, there exists fundamental differences between them. For instance, classification assigns data items to predefined categories or classes, while retrieval is the process of searching relevant information from a collection of items, often using similarity measures calculated from their embeddings. Another distinguishing factor is the asymmetric distribution of item embeddings in retrieval tasks. This asymmetric distribution can often result in a complex and nuanced performance landscape for retrieval models. Furthermore, the discrepancy between distributions of relevant and irrelevant documents in retrieval tasks is often dependent on the specific query. This means that any transfer learning evaluation metric must take into account this query-dependent variability to provide an accurate measure of model performance. In addition, one also needs to take account into the importance of each examined query, to be able to provide less biased predictions regarding overall retrieval performance in downstream tasks.

Given these unique challenges, we propose a method designed specifically to assess model transferability in retrieval tasks. Our approach utilizes the Maximum Mean Discrepancy (MMD) [15] to quantify the discrepancy between distributions of relevant and irrelevant document embeddings in embedding spaces. We employ a kernel-based method to estimate these discrepancies, which are then averaged over

*Partial work done at Salesforce.

a large number of queries considering the target dataset's distribution. Note that in downstream retrieval tasks, using cosine similarity or inner product to estimate items' relevance from fine-tuned embeddings, well fits into the use of kernel method on estimating their relevance from pretrained embeddings. The proposed method allows us to effectively estimate the retrieval performance of fine-tuned models across various datasets, including image, text, and multi-modal domains. In addition, we design a way to measure transferability between datasets particularly suitable for retrieval tasks. In our experiments, we find that the transferability between datasets has a strong correlation with their statistical discrepancies in retrieval settings.

We summarize the contribution of this work as follows: (1) To our knowledge, this work is the first attempt to design a transfer learning evaluation metric in retrieval tasks. (2) Experimental results on various modalities suggest that the proposed method has capability to predict retrieval performance of fine-tuned models using pretrained embeddings. (3) We also design a straightforward way to measure dataset transferability in retrieval settings. Experiments show strong correlations between dataset distance and relative retrieval gain from fine-tuning.

2. Related Work

Retrieval applications in machine learning. Retrieval applications have become a cornerstone in the realm of machine learning. In the context of search engines, retrieval systems utilize complex algorithms to rank the relevance of web pages, documents, or other data entities to a user's search query [34, 41]. Recommendation systems represent another significant domain where retrieval applications play a vital role. These systems analyze user behavior, preferences, and interactions to suggest content, products, or services that users are likely to find appealing [56]. Furthermore, rencently the integration of retrieval mechanisms with LLMs in the context of RAG systems has opened new era for leveraging external knowledge sources. This fusion empowers LLMs to provide responses that are both contextually relevant and up-to-date [24]. Recent advances in retrieval applications also extend to specialized fields such as medical information retrieval.The development of the domain-specific retrieval systems, enhanced by deep learning techniques, aims to provide healthcare professionals with timely access to medical literature, patient records, and other critical information [55].

Transfer learning evaluation. Here we discuss existing transfer learning evaluation methods that are relevant to our paper, such as LEEP [37], H-score[3], LogME [54], GBC [40] and NCE [50]. NCE considers the conditional entropy between the label assignments of the two tasks, which is shown to be related to the loss of the transferred model. LEEP identifies transferable components using linear mapping, whereas H-score measures transferability through the entropy of a model's feature importance. LogME employs logistic regression to assess the correlation between two datasets. GBC leverages the probability distributions of source and target domains, calculating the Bhattacharyya coefficient to capture the dissimilarities between them. Estimating distribution difference using GBC requires assuming per-class Gaussian distribution.

Kernel methods in deep learning. Kernel methods [4, 7] have significantly influenced the field of deep learning, evolving beyond their traditional roles in pattern recognition and classification [8]. Today they are being increasingly integrated into deep neural network architectures to augment their capabilities in a variety of tasks [25, 30, 45]. The use of kernel methods enables effective handling of high-dimensional data, facilitating non-linear transformations that aid in complex decision boundary formations in neural architectures. Notably, in [16] authors show that existing contrastive learning methods can be reinterpreted as learning kernel functions that approximate some fixed positive-pair kernel, which serves as a support evidence to our work. On the other hand, MMD [15] as a technique for measuring distances between probability distributions has gained prominence in deep learning, especially in areas like generative modeling and domain adaptation. Its utility in Generative Adversarial Networks (GANs) [14] and transfer learning [33] is noteworthy, as it allows for effective comparison of distributions without presuming their underlying forms.

3. Methodology

3.1. Problem Statement

The main purpose of this work is to propose a transfer learning evaluation metric used in retrieval scenarios. Let's consider q denotes query with its embedding vector v_q gained by mapping model f_θ. The document embeddings given q follow the distribution denoted by D_q. Furthermore, let $D(v_q^+|q)$ (denoted as D_q^+) represent the conditional distribution of the relevant documents with corresponding embedding vectors $\{v_q^+\}$. Similarly, let $D(v_q^-|q)$ (denoted as D_q^-) be the conditional distribution of irrelevant documents with its embedding vectors $\{v_q^-\}$, where $v_q, v_q^+, v_q^- \in \mathcal{X}$ meaning that their embeddings lie in a shared space. One can then estimate the discrepancy between the two conditional distributions by quantifying the differences:

$$H_q(D_q^+, D_q^-) \approx Z_q(\{v_q^+\}, \{v_q^-\}) \tag{1}$$

The exact form of the function Z depends on the method one uses to quantify the difference. Note that the conditional distributions D_q^+, D_q^- and their discrepancy depends on q. The final evaluation metric score S can be calculated as:

$$S = \int Z_q(\{v_q^+\}, \{v_q^-\}) dq \tag{2}$$

We assume that the proposed transferability score S can measure the capability of the pretrained embedding model f_θ in separating relevant and irrelevant documents after fine-tuning on a given dataset. Here we provide some intuitions on indicating the significance of the difference between D_q^+ and D_q^-.

Lemma 1. *Given a metric space $(\mathcal{X}, d)$, where d is any valid distance metric satisfying the triangle inequality, and three points $v_q, v_1, v_2 \in \mathcal{X}$ such that $d(v_q, v_1) < c$, $d(v_q, v_2) < c$, and $d(v_1, v_2) = c'$, then $c' < 2c$.*

The proof is straightforward by using triangle inequality. This simple fact indicates that given some query q, when relevant (or positive) document embeddings are defined based on bound c from its query embedding using any valid distance metric, the distances between any relevant document embeddings in $\mathcal{X}$ are bounded as well. On the other hand, negative document embeddings $\{v_q^-\}$ are expected to be further away from v_q. As there is no constraint that these negative embeddings should be close to each other, they can be spread out over the entire embedding space, leading to a potentially different and wider distribution compared to D_q^+. This observed difference in distributions between positive and negative embeddings can serve as an effective criterion when devising a metric for transfer learning evaluation for retrieval tasks.

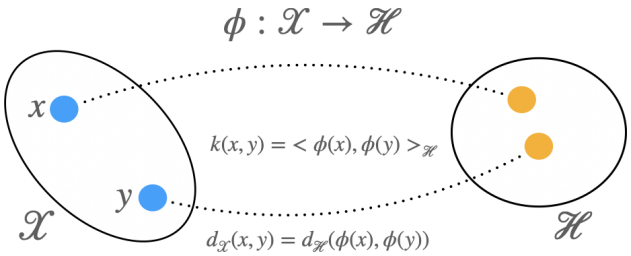


Figure 1. Illustration of pretrained embedding space $\mathcal{X}$ and fine-tuned embedding space $\mathcal{H}$.

3.2. Fine-Tuning as Kernel Learning

With the above assumption, we treat the fine-tuning neural networks as functions mapping from the space of pretrained embeddings as $\mathcal{X}$, to a space of finetuned embeddings $\mathcal{H}$ (as shown in Fig 1). Formally, we can represent such a network as $\phi : \mathcal{X} \to \mathcal{H}$, where $\phi(x)$ signifies the embedding of input x. This function maps the original embeddings into the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$. The finetuned embeddings in $\mathcal{H}$ are then represented as $\phi(x)$ and $\phi(y)$. Given two embeddings in the original feature space $\mathcal{X}$, denoted as x and y, the distance between these embeddings in this space is defined as:

$$d_{\mathcal{X}}(x, y) \approx d_{\mathcal{H}}(\phi(x), \phi(y)) = \|\phi(x) - \phi(y)\|_{\mathcal{H}} \quad (3)$$

In the following, we denote $d_{\mathcal{X}}(x, y)$ as $d(x, y)$ for simplicity. The function ϕ can be thought of as pulling back the metric from the RKHS $\mathcal{H}$ to the original feature space $\mathcal{X}$. The underlying principle is that the fine-tuning process, through the function ϕ, provides a more appropriate representation of the distances between embeddings for the retrieval task. Thus, we can relate the distances in the original feature space $\mathcal{X}$ and the RKHS $\mathcal{H}$. This distance can be expanded using the properties of the inner product in the RKHS:

$$\begin{aligned} d(x, y)^2 &= \|\phi(x) - \phi(y)\|_{\mathcal{H}}^2 \\ &= \langle \phi(x) - \phi(y), \phi(x) - \phi(y) \rangle_{\mathcal{H}} \\ &= \langle \phi(x), \phi(x) \rangle_{\mathcal{H}} - 2\langle \phi(x), \phi(y) \rangle_{\mathcal{H}} + \langle \phi(y), \phi(y) \rangle_{\mathcal{H}} \end{aligned}$$

Recall that for a kernel associated with an RKHS, the kernel function $k(x, y)$ is related to the mappings $\phi(x)$ and $\phi(y)$ in the space $\mathcal{H}$ by:

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$$

The squared distance can then be represented using the kernel function as:

$$d(x, y)^2 = k(x, x) - 2k(x, y) + k(y, y) \quad (4)$$

For d to be a valid metric over space $\mathcal{X}$, and for Lemma 1 to be applicable, it is essential for the kernel k to be positive definite. This property ensures that d induces a valid inner product in the associated RKHS. Embeddings from neural networks that align with such kernels inherit the RKHS's structure, making MMD an appropriate measure for the distance between distributions of these embeddings.

3.3. MMD for Evaluation

Distinguishing two distributions with finite samples is known as the *Two-Sample Test*. One approach to conduct this test is using the MMD [15] as previously mentioned. MMD offers a principled approach to quantify the discrepancy between two distributions based on their mean embeddings in a RKHS. Given our context, the MMD between distributions D_q^+ and D_q^-, using a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, is defined as:

$$\begin{aligned} \text{MMD}^2(D_q^+, D_q^-) &= \|\mu_{D_q^+} - \mu_{D_q^-}\|_{\mathcal{H}}^2 \\ &= \mathbb{E}_{D_q^+}[k(v^+, v^+)] \\ &\quad - 2\mathbb{E}_{D_q^+, D_q^-}[k(v^+, v^-)] \\ &\quad + \mathbb{E}_{D_q^-}[k(v^-, v^-)] \end{aligned} \quad (5)$$

where $v^+ \sim D_q^+$, $v^- \sim D_q^-$, and μ represents the mean embedding of the corresponding distribution in RKHS. In practice we use finite samples from distributions to estimate

the MMD distance:

$$\hat{M}_{k,q}(\{v^+\},\{v^-\}) = \frac{1}{n(n-1)}\sum_{i\neq j} k(v_i^+, v_j^+) - \frac{2}{nm}\sum_{i,j=1}^{n,m} k(v_i^+, v_j^-) + \frac{1}{m(m-1)}\sum_{i\neq j} k(v_i^-, v_j^-) \quad (6)$$

Here n and m are the numbers of samples from D_q^+ and D_q^-, respectively. We denote the quantity as $\hat{M}_{k,q}$ for simplicity. Using the above quantity, the discrepancy function Z_q from our initial problem statement can be represented as:

$$Z_q(\{v_q^+\},\{v_q^-\}) \approx \hat{M}_{k,q} \quad (7)$$

Following the above discussion, another way to judge the separability between positive and negative embeddings is to look at the statistical significance of MMD via two-sample testing. In implementation, a small p-value from the permutation test can serve as an indicator of strong separability between positive and negative embeddings, where

$$I_{k,q} = \begin{cases} 1 & \text{if } p\text{-value} < 0.05 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

3.4. Kernel Selection

A follow-up question is: how to choose the appropriate kernel in MMD for metric calculation? To address the question we consider a few approaches in the following. We observe that although the following approaches tackle the problem from various directions, the experimental results suggest similar patterns.

Two-Sample Testing: Given a set of queries, one can compute the p-value associated with each query as discussed above. The efficacy of kernels can then be compared based on the count of significant p-values. A kernel that consistently provides a higher number of significant p-values is indicative of its superior capability in discerning distributional differences for the given embeddings. Specifically, the count of significant p-values across a set of queries is defined as $C_k = \sum_q I_{k,q}$.

The above equation is also useful when comparison is needed across kernels. While $\hat{M}_{k,q}$ values from different kernels are not directly comparable, the magnitude of C_k can serve as a reference for the separability between positive and negative embeddings using the underlying kernel.

SVM Classifier: One can also utilize SVM classifier to find a kernel that gives the best classification performance on positive and negative embeddings. The underlying assumption is that the kernel that helps distinguish well between positive and negative embeddings can provide good separability between their distributions by using MMD as well. In implementation, one can record the classification metric such as F1 score or Recall for each query, and calculate the mean of such metrics across queries. A better overall classification performance indicates better separability using the corresponding kernel.

Empirical Evidence: Another viewpoint for kernel selection is based on empirical evidence. For example, a higher curvature in embeddings may suggest the presence of more complex, non-linear relationships within the embeddings. Consequently, these embeddings might be matched with kernels that capture an appropriate order of moments. On the other hand, low curvature values can indicate simpler relationships within embeddings. Note that curvature can only serve as support evidence to intuitively understand embedding space. One cannot merely depend on it for kernel selection. Another empirical way for kernel selection is to look at the performance of calculated correlations as an indicator of using the underlying kernel given ground truth. For example, a small standard deviation of correlations from multiple runs suggests that the kernel provides consistent evaluations of MMD across different queries, which indicates the stability of the kernel.

3.5. Dealing with Long-Tail Distribution

Once we are able to estimate distribution difference per query, we need to consider into account the distribution of datasets. In retrieval tasks, we often encounter scenarios where certain queries, referred to as *head queries*, appear much more frequently than others. This long-tail distribution can introduce biases in both training and evaluation. To ensure that each query is adequately represented, especially when there's a bias towards the *head queries*, we can weight each query q by its inverse frequency:

$$w(q) = \frac{1}{freq(q)} \quad (9)$$

A similar idea can be extended to the common case where queries formulate different classes and the number of queries in each class is imbalanced. One can also define the frequency of a query as the number of queries in the corresponding class. However, whether to assign equal weight to each query class as part of the calculation of the proposed metric is a user-defined choice. To obtain consistent prediction power, in retrieval evaluation of fine-tuned models one needs to make corresponding adjustment in the measurement set as well. For datasets without explicitly specifying the frequency of queries, one can simply let $w(q) = 1$.

3.6. Proposed Metric – RetMMD

The final metric score is computed by averaging the metric values over a large number of queries. To compare embeddings using a single kernel, we use Eqn (6) to define the

metric score S_{M_k} as:

$$S_{M_k} = \sum_q w_q \hat{M}_{k,q} \tag{10}$$

The above equation considers models using a consistent kernel setting thus the calculated MMD distances are comparable. Yet in the case where one needs to compare models with different kernel settings, the above equation will lose its power, since directly comparing MMD distances between two distinct kernel settings is not meaningful. In this case, one way to compare across different kernels is to utilize the counting of significant p-values in MMD two-sample test given a fixed set of queries, where one can pick the kernel k^* that leads to the largest counting number as the default kernel choice for the corresponding model. Specifically, k^* can be searched within a set of kernel choices as $k^* = \arg\max_k \left(\sum_q w_q I_{k,q}\right)$, and the metric score S_{C_k} is defined as:

$$S_{C_k} = \max_k \left(\sum_q w_q I_{k,q}\right) \tag{11}$$

In the following, we denote S_{M_k} as RetMMD-M, and S_{C_k} as RetMMD-C for reference. In practice, we use Principle Component Analysis (PCA) [17] to reduce the dimension of embeddings before calculating MMD. It aims to remove redundant information from high-dimensional embeddings and mitigate the effect of the curse of dimensionality. In the implementation, one can automate the selection of the number of principal components, by using the intrinsic dimension of embeddings estimated from various methods [9, 12, 23], or set up a threshold of the explained variance in PCA, such as 90%.

3.7. Transferability between Datasets

Here we also study the transferability between datasets in retrieval tasks. We first define transferability between source (pre-training) dataset D_S and target (fine-tuning) dataset D_T following the approach in [2]. In [2] authors use a relative drop in classification error to quantify transferability between datasets in classification scenario. Similarly, in the retrieval case we define transferability between datasets based on the relative difference between retrieval performance, such as recall, where

$$\tau(D_S \to D_T) = \frac{R_{D_S \to D_T} - R_{D_T}}{1 - R_{D_T}} \tag{12}$$

Here R_{D_T} represents the retrieval performance from a model that is trained on D_T from scratch, and $R_{D_S \to D_T}$ is the retrieval performance of a model that is first pretrained on D_S then fine-tuned on D_T.

On the other hand, we calculate the distribution distance between datasets, in a specifically designed way suitable for retrieval tasks: given some model, we treat relevant documents associated with each query as a class and calculate the mean embedding within the class. Thus we will be able to get a set of embeddings, where each embedding carries information from all relevant documents for a query. Let μ_{q_S} be the mean embedding of relevant documents for some query $q_S \in D_S$. Similarly, let μ_{q_T} be the mean embedding of relevant documents for a query $q_T \in D_T$. The distribution difference $d_{S,T}$ between D_S and D_T can then be calculated from two collections of mean embeddings:

$$d_{S,T} = Z(\{\mu_{q_S}\}, \{\mu_{q_T}\}) \tag{13}$$

Here Z can be any valid approach to calculate the distance between distributions, such as using MMD. If one chooses to adopt MMD distance, calculating dataset distance is then similar to comparing RetMMD scores obtained from source and target datasets respectively. Note that one document can serve as a relevant document for multiple queries and be calculated separately. The above equation focuses on the implementation for retrieval tasks, as opposed to the default way to calculate dataset distance, where each sample is treated as an individual data point to contribute to distribution distance.

4. Experiments

In this section we conducted extensive experiments on various datasets and modalities.

4.1. Datasets and Implementation Details

Image: We mainly experimented on a few image datasets: Caltech-UCSD Birds-200-2011 (CUB200) [51], CARS196 [20] and Stanford Online Products (SOP) [48] which are commonly used in image retrieval studies. We also include the Stanford Dogs dataset [19], SVHN [36], CIFAR-10 and CIFAR-100 [21] for evaluating dataset transferability. For fine-tuning, we took a common approach where in-batch contrastive learning was used. Here we mainly experiment on vision transformer (ViT) architectures [11] with different sizes and pretraining methods, including models pretrained on ImageNet-1K [46] and ImageNet-21K [44].
Text: We implement text experiments on NFCorpus [6], FiQA-2018 [10] and FEVER [49]. We use six pretrained models, including DistilBERT (D-BERT) [47](distilbert-base-uncased), DistilBERT trained on Natural Questions dataset (D-BERT-NQ) [22] (nq-distilbert-base-v1), DistilBERT trained on MS MARCO passage dataset (D-BERT-MSMARCO) [38] (msmarco-distilbert-dot-v5), distilled Roberta (D-RoBERTa) [32] (all-distilroberta-v1) and two variants of MiniLM [52] (all-MiniLM-L12-v1 and all-MiniLM-L6-v1) provided by Sentence-Transformers library [43]. These models are fine-tuned in a similar procedure, where we save the model that has the best $MAP@1$ performance on the validation set. In cases where there are

few amounts of relevant documents per query, we use data augmentation techniques to augment the set of relevant documents, for example using back-translation and summarization if needed.

Multimodal: We also test the performance of RetMMD on the image-to-text retrieval task on the MS-COCO dataset [31]. In this study, we compare CLIP [42], ALBEF [27], BLIP [28] and BLIP-2 [29] models provided by the LAVIS library [26]. We use R@1 in [29] as a reference for fine-tuned results. Note that except for CLIP, in the other papers authors first use text embedding to broadly select a set of candidates and then use multimodal embedding to perform retrieval. For simplicity in our experiments, we use multimodal embedding for the evaluation of consistency.

In implementation, without further notice, for each model and dataset combination, we randomly pick 100 queries, and for each query, we randomly select 5 positive documents and 50 negative ones. When it comes to the process of fine-tuning, we acknowledge that there is a wealth of sophisticated methods available. However, it is neither practical nor required to test every existing fine-tuning technique. Thus, to facilitate comparison we aim to conduct our experiments under a broadly consistent setting. More experimental details are provided in the supplementary material.

4.2. Analysis

Two-sample test with MMD: We first attempt to explore choosing the optimal kernel by using two-sample test with MMD. We start experimenting with the CARS196 dataset using pretrained ViT-B model. Figure 2 shows the percentage of significant p-values from 5 runs. The result shows linear kernel significantly outperforms other kernels in this case, indicating its better capability of separating between positive and negative embeddings. In other cases where the amount of significant p-values is almost 100% and there is no strong indicator of choosing kernel using the other approaches, we use RBF kernel as the default setting.

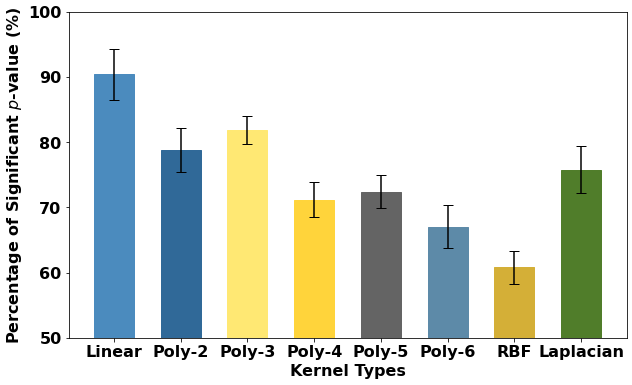


Figure 2. Percentage of significant p-values from a two-sample test using MMD from different kernels.

SVM classifier for kernel selection: Next we use SVM classifier with various kernel choices to classify positive and negative embeddings generated from text models and plot the corresponding F1 scores. In experiments we randomly select 50 positive documents (after augmentation) and 50 negative ones. The F1 scores are averaged across 100 randomly selected queries and each with 5-fold cross-validation. Training and test data were split into a consistent ratio of 4 to 1. Figure 3 shows the averaged F1 scores from different pretrained text models with regard to kernel choices for SVM classification. One can see RBF kernel outperforms other ones in most cases, suggesting its better separability on the NFCorpus dataset.

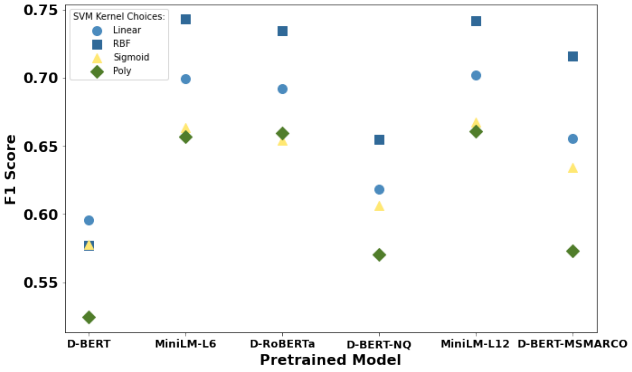


Figure 3. Averaged F1 scores from SVM classifiers using different kernels. Dataset: NFCorpus [5].

Effect of moment matching: The above analysis provides insights into the separability of positive and negative embeddings with different kernel choices. Here we would also like to see the effect of moment matching reflected by using a polynomial kernel on the performance of the proposed metric. Figure 4 shows weighted Kendall rank correlations [18] using a polynomial kernel with different degrees. As suggested from previous analysis, matching lower order moments provides better prediction performance on CARS196. For comparison, matching higher order moments provides better results on the COCO dataset, which also aligns with the intuition provided in curvature analysis.

RetMMD vs retrieval performance during fine-tuning: We fine-tune ViT-S on CUB200 dataset and save checkpoints during training. For each saved checkpoint, we evaluate R@1 as retrieval metric, and compute RetMMD-M scores using cosine kernel. In Fig 5 one can see higher RetMMD values with cosine kernel correspond to higher R@1. Intuitively, in a common retrieval setting, a better fine-tuned model would have better separability between positive and negative documents using cosine similarity between the query and their embeddings. After fine-tuning, this can be approximated as using MMD with a cosine kernel. The pretrained model without any fine-tuning results in very low R@1 and metric

Table 1. Weighted Kendall rank correlations from different methods evaluated on various models and datasets.

	CUB200 [51]	CARS196 [20]	SOP [48]	NFCorpus [6]	FEVER [49]	FiQA-2018 [10]	COCO [31]
LogME [54]	0.52 ± 0.00	0.58 ± 0.07	0.70 ± 0.05	-0.60 ± 0.00	-0.12 ± 0.00	-0.35 ± 0.00	-0.09 ± 0.14
GBC [40]	0.77 ± 0.10	0.75 ± 0.13	0.71 ± 0.10	0.70 ± 0.13	$\mathbf{0.67 \pm 0.09}$	0.33 ± 0.00	-0.72 ± 0.00
RetMMD-S	$\mathbf{0.90 \pm 0.00}$	0.57 ± 0.18	0.73 ± 0.07	-	-	-	0.79 ± 0.14
RetMMD-M	$\mathbf{0.90 \pm 0.08}$	$\mathbf{0.89 \pm 0.08}$	$\mathbf{0.80 \pm 0.13}$	$\mathbf{0.94 \pm 0.00}$	0.52 ± 0.05	$\mathbf{0.80 \pm 0.02}$	$\mathbf{1 \pm 0.00}$

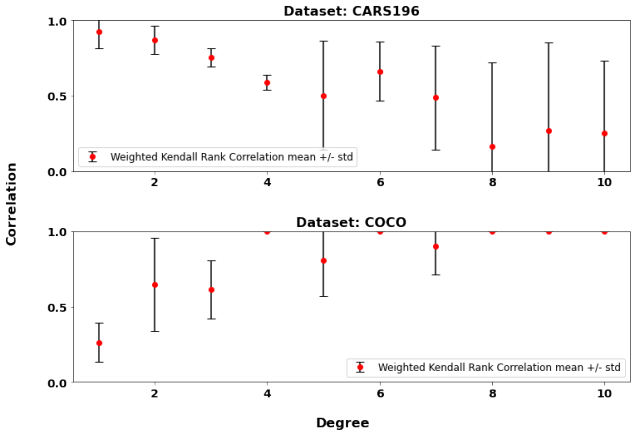


Figure 4. Weighed Kendall rank correlation vs degree of polynomial kernel in calculation RetMMD.

value $S_{M_k} = 0.04$ which is not shown in the figure.

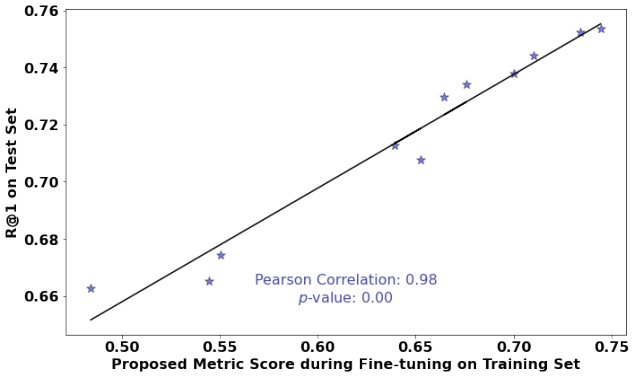


Figure 5. R@1 vs RetMMD-M values calculated from fine-tuned models at different fine-tuning stages on CUB200 [51].

4.3. Quantitative Evaluations

Comparison with other transfer learning evaluation methods: We compare the proposed method RetMMD with GBC and LogME where corresponding metric scores can be calculated using document embeddings. For calculating GBC, we first perform PCA on the embeddings following the steps in [40]. Other transfer learning evaluation metrics such as LEEP, NCE and H-Score are designed for classification problems that require pseudo labels and do not directly fit into the settings of retrieval tasks. We finally evaluate the effectiveness of transfer learning evaluation metrics by computing the weighted Kendall Rank Correlation [18] between the metric scores and the corresponding retrieval performance after fine-tuning. Weithed Kendall rank correlation extends the classic Kendall tau correlation by accounting for the varying significance of different pairs in the ranking, where higher ranks carry more weights. In Table 1 we see that the proposed method RetMMD shows stronger correlations between metric score and fine-tuning performance on most of the datasets. For implementing RetMMD-S on text datasets, we find proportions of significant p-values are nearly 100% on different kernels, where we only reported scores using RetMMD-M.

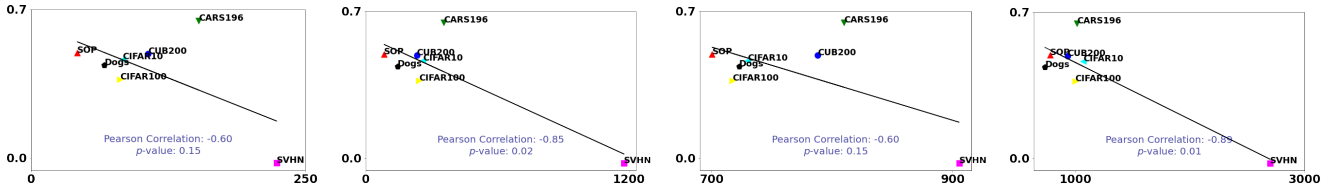
Table 2. F1 score and number of significant p-values using different kernels applied on OpenAI embeddings on NFCorpus [5] dataset.

Kernel	F1 score	Significant p-values
Linear	0.50 ± 0.01	47.6 ± 0.47
Poly, degree=3	0.51 ± 0.01	48.6 ± 0.47
RBF	0.43 ± 0.01	34.3 ± 1.69

Table 3. Comparisons of RetMMD, zero-shot and fine-tuned retrieval performance against different settings on NFCorpus [6].

Model	Zero-shot MAP@1	RetMMD (linear)	RetMMD (RBF)	Fine-tuned MAP@1
D-BERT [47]	0.04	114.97	0.10	0.39
MiniLM-L6 [52]	0.27	88.43	0.69	0.47
D-RoBERTa [32]	0.37	189.49	0.26	0.43
D-BERT-NQ [22]	0.40	124.62	0.10	0.41
MiniLM-L12 [52]	0.27	94.33	0.96	0.48
D-B-MSMARCO [38]	0.40	132.95	0.15	0.42
OpenAI [35]	0.47	**316.86**	-	-

Study on OpenAI embeddings: We also attempt to predict the performance of OpenAI embeddings using the proposed method. In this regard, we used the OpenAI embedding service "text-embedding-ada-002" [35] to generate the embedding vectors of the queries, positive and negative text samples on NFCorpus [5] dataset as an example, Since we did not have access to fine-tune the OpenAI model, we de-



(a) MMD (b) MMD + proposed (c) Wasserstein (d) Wasserstein + proposed

Figure 6. Pearson correlation between dataset distance (x-axis) and $\tau(D_S \to D_T)$ (y-axis) on image datasets. Each subfigure corresponds to a different setting. In each setting, dataset distance is calculated between source dataset ImageNet [46] and different target datasets.

cided to present the experiment as an exploratory section. In order to find the best kernel to calculate MMD score, we used both SVM classifier and kernal two-sample test. Table 2 shows that linear and polynomial kernels reach better performance in terms of classification and number of significant p-values. Here we use linear kernel for calculating RetMMD scores for reference. Results in Table 3 show the zero-shot performance of the OpenAI embedding is already as good as the best fine-tuned model from other candidates, indicating its better retrieval performance after fine-tuning. This behavior is suggested by its significant larger RetMMD score reported in the table.

Note that whether RetMMD is sensitive to zero-shot performance depends on the choice and definition of the kernel. Here we display RetMMD for both linear kernel and RBF kernel in Table 3. With the proposed kernel selection method (i.e. two-sample test or SVM classifier), one can see that with the selected RBF kernel for the six models other than OpenAI's, RetMMD is not sensitive to zero-shot performance. On the other hand, it is reasonable that with linear kernel, MMD distance has higher correlation with zero-shot performance, by the definition of linear kernel itself. More specifically, with linear kernel $\langle \phi(x), \phi(y) \rangle = k(x, y) = x^T y$, ϕ (fine-tuned model) is simply a linear transformation of the pretrained embeddings. It is not surprising that zero-shot performance sensitive to predicted scores with linear kernel.

Dataset transferability in retrieval tasks: We further explore dataset transferability following our earlier discussion in Section 3.7. For the proposed approach we use mean embedding of positive documents per query to calculate dataset distance in retrieval tasks. For default setting we simply take embeddings from each single image in the dataset. Figure 6 shows the distance between ImageNet and various datasets for downstream fine-tuning. One can see that using the proposed way for evaluation achieves significant Pearson correlations [13] between relative gains from fine-tuning and dataset distance. The significance was revealed using both MMD and 1-Wasserstein distance as shown in the figure, which implies the universality of the proposed approach. One can utilize the approach to explore patterns of dataset transferability in other scenarios.

Evaluation time: In experiments we recorded the time and resources needed for evaluating and fine-tuning model candidates. We experimented with the proposed evaluation method using the SOP dataset and the five pretrained image models on one NVIDIA A10 GPU and record its running time. We also recorded the time needed to get the best retrieval performance for these models with fine-tuning. Figure 7 shows the proposed method is much faster than directly fine-tuning these models, which greatly saves time and computational cost for model selection.

Figure 7. Time needed for evaluating using RetMMD and directly fine-tuning on SOP dataset [48].

5. Conclusion and Discussion

In this paper we discussed on evaluating model and data transferability in retrieval tasks. We designed a metric based on MMD and kernel method, utilizing the characteristic that calculating cosine similarity or inner product to estimate relevance from fine-tuned document embeddings, well fits into the use of kernel method on estimating relevance from their pretrained embeddings. It is worth mentioning that in the current setting relevant documents are treated in equal positions. An interesting direction to explore would be designing algorithms that take account into the importance or the rank of each document for more sophisticated retrieval settings. Another direction is to study scenarios where retrieval does not merely rely on embedding similarities. The challenge of building such metrics will then depend on the underlying techniques used for retrieval.

6. Acknowledgement

The first author would like to thank Joshua Correa and Linsey Pang for their advice on this work.

References

[1] A review on recent research in information retrieval. *Procedia Computer Science*, 201:777–782, 2022. The 13th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 5th International Conference on Emerging Data and Industry 4.0 (EDI40). 1

[2] David Alvarez-Melis and Nicolo Fusi. Geometric dataset distances via optimal transport. In *Advances in Neural Information Processing Systems*, pages 21428–21439. Curran Associates, Inc., 2020. 5

[3] Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *2019 IEEE international conference on image processing (ICIP)*, pages 2309–2313. IEEE, 2019. 1, 2

[4] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 541–549. PMLR, 2018. 2

[5] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. 2016. 6, 7

[6] Vera Boteva, Demian Gholipour, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. 2016. 5, 7

[7] Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2009. 2

[8] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995. 2

[9] Mengyu Dai, Haibin Hang, and Xiaoyang Guo. Adaptive feature interpolation for low-shot image generation. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XV*, page 254–270, 2022. 5

[10] Dayan de França Costa and Nadia Felix Felipe da Silva. Inf-ufg at fiqa 2018 task 1: Predicting sentiments and aspects on financial tweets and news headlines. In *Companion Proceedings of the The Web Conference 2018*, page 1967–1971. International World Wide Web Conferences Steering Committee, 2018. 5, 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 5

[12] Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7, 2017. 5

[13] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007. 8

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 2

[15] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. 1, 2, 3

[16] Daniel D. Johnson, Ayoub El Hanchi, and Chris J. Maddison. Contrastive learning can find an optimal basis for approximately view-invariant functions. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[17] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986. 5

[18] M. G. Kendall. A new measure of rank correlation. 30(1/2): 81–93, 1938. 6, 7

[19] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, 2013. 5, 7

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009. 5

[22] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur 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. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. 5, 7

[23] Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems*. MIT Press, 2004. 5

[24] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv:2005.11401*, 2020. 1, 2

[25] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017. 2

[26] Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C. H. Hoi. Lavis: A library for language-vision intelligence, 2022. 6

[27] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems*, 2021. 6

[28] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 6

[29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 6

[30] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1718–1727. PMLR, 2015. 2

[31] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 6, 7

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 5, 7

[33] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, page 2208–2217. JMLR.org, 2017. 2

[34] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. 2

[35] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022. 7

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[37] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pages 7294–7305. PMLR, 2020. 1, 2

[38] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660, 2016. 5, 7

[39] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 1

[40] Michal Pándy, Andrea Agostinelli, Jasper Uijlings, Vittorio Ferrari, and Thomas Mensink. Transferability estimation using bhattacharyya class separability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9172–9182, 2022. 1, 2, 7

[41] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, page 257–266. Association for Computing Machinery, 2017. 2

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 6

[43] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2020. 5

[44] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 5

[45] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4393–4402. PMLR, 2018. 2

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 5, 8

[47] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 5, 7

[48] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 7, 8

[49] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018. 5, 7

[50] A. Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1395–1405, 2019. 1, 2

[51] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5, 7

[52] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020. 5, 7

[53] Haolun Wu, Yansen Zhang, Chen Ma, Fuyuan Lyu, Fernando Diaz, and Xue Liu. A survey of diversification techniques in search and recommendation, 2023. 1

[54] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR, 2021. 1, 2, 7

[55] Cyril Zakka, Rohan Shad, Akash Chaurasia, Alex R. Dalal, Jennifer L. Kim, Michael Moor, Robyn Fong, Curran Phillips,

Kevin Alexander, Euan Ashley, Jack Boyd, Kathleen Boyd, Karen Hirsch, Curt Langlotz, Rita Lee, Joanna Melia, Joanna Nelson, Karim Sallam, Stacey Tullis, Melissa Ann Vogelsong, John Patrick Cunningham, and William Hiesinger. Almanac — retrieval-augmented language models for clinical medicine. *NEJM AI*, 1(2):AIoa2300068, 2024. 2

[56] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1), 2019. 2