Getting off the DIME: Dimension Pruning via Dimension Importance Estimation for Dense Information Retrieval

GUGLIELMO FAGGIOLI, University of Padua, Italy
NICOLA FERRO, University of Padua, Italy
RAFFALE PEREGO, National Research Council (CNR), Italy
NICOLA TONELLOTTO, University of Pisa, Italy

Dense Information Retrieval (IR) systems rely on neural networks to embed documents and queries within a latent low-dimensional space. Among the Dense IR approaches, bi-encoders are particularly popular, as they achieve state-of-the-art performance and allow for efficient encoding of documents and queries. Nevertheless, using this class of systems, by construction, all the documents and queries are represented using the same set of dimensions. In this paper, we introduce the Manifold Clustering (MC) hypothesis which states that, for each query, there exists a query-dependent manifold of the original embedding space where the query and documents relevant to it cluster more effectively. We empirically validate the MC hypothesis showing that it is possible to find a query-dependent linear subspace of the original embedding space where high retrieval effectiveness is achieved. To find such subspaces, we propose the Dimension IMportance Estimators (DIMEs), a class of models that associate an importance score with each dimension of an embedding and can be used to project the dense representations only on the most important dimensions. We first demonstrate the effectiveness of the DIMEs by proposing an oracle DIME which employs annotated documents and induces performance improvements as big as +184% in terms of AP. To demonstrate the practical applicability of the DIMEs beyond the oracle, we also propose a set of DIMEs based on pseudo-relevance and active feedback that induce improvement as big as +49.6% in terms of AP and +55.9% in terms of nDCG@10. The effectiveness of such DIMEs not only empirically supports the MC hypothesis, but illustrates an actual strategy to outperform the state-of-the-art that does not require any form of retraining, fine-tuning or re-indexing and can be efficiently implemented at retrieval time.

CCS Concepts: • **Information systems** → **Document representation**; **Query representation**; **Retrieval models and ranking**.

Additional Key Words and Phrases: Dimension Importance Estimation, Dense Retrieval, Feature selection

ACM Reference Format:
Guglielmo Faggioli, Nicola Ferro, Raffale Perego, and Nicola Tonellotto. 2018. Getting off the DIME: Dimension Pruning via Dimension Importance Estimation for Dense Information Retrieval. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 34 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Information Retrieval (IR) systems have benefited from the emergence of pretrained *Large Language Models (LLMs)*, leading to the development of new systems with improved retrieval effectiveness over the previous state-of-the-art IR systems [15]. These new IR systems leverage neural networks to represent and match queries and documents [40]. Among them, dense IR systems rely on learned semantic representations for queries and documents, called contextualized embeddings. Queries and document embeddings are characterized by a lower dimensionality yet denser encoding than

Manuscript submitted to ACM

traditional sparse IR systems. Dense IR systems differ significantly from IR approaches based on traditional methods like BM25 and query language models, which rely on lexical matching – where query terms in a document indicate relevance. Instead, they use signals derived from semantic similarities in the latent space. This departure allows them to effectively address challenges related to synonymity and polysemy thanks to the underlying pre-trained encoder-only language model used to build their representation [27, 28, 64, 68]. In a dense IR system, queries and documents are encoded as multidimensional vectors whose dimensions represent features that the model has learned to be important for representing the textual content. Each vector dimension may correspond to a specific aspect learned from the data – for example, it might capture the semantic meaning, the syntactic structure, or other linguistic features of the encoded text. The values along those dimensions measure the presence and the importance of those features in a given query or document. Ad hoc retrieval in this setting involves identifying the document embeddings that are nearest to the query representation in the latent space and subsequently ranking them according to the specified similarity measure, typically the dot product. These approaches operate according to the *clustering hypothesis* [62], which posits that documents with similar meanings — and thus representations — tend to be relevant to the same queries. Another well-known hypothesis concerning representation spaces was formulated by Bengio et al. [5] and is the *manifold hypothesis*. The manifold hypothesis was originally postulated for images' latent representation spaces and states that high-dimensional data of interest often lives in an unknown lower-dimensional manifold embedded in the representation space. There is strong evidence supporting this hypothesis in image representations [50]. Recently, several works in the natural language processing and computational linguistics areas have found that contextualized embeddings from LLMs lie in low-dimensional linear subspaces [26, 44] or nonlinear manifolds [9, 10].

We conjecture that both the *clustering* and the *manifold hypotheses* hold at the same time for IR and that it is possible to find a subspace of the original latent space that best represents the query and the associated relevant documents. However, when it comes to the manifold hypothesis, instead of assuming a *single* low-dimensional subspace for all the queries and documents, we hypothesize that each query has its own low-dimensional subspace, i.e., we have *multiple* low-dimensional subspaces, one per query, where documents can be projected as well. This also aligns with the clustering hypothesis since we speculate that the subspace that best represents a query topic and its relevant documents depends on the query itself. Putting everything together, we formulate the following *Manifold Clustering hypothesis (MC hypothesis)* for dense IR systems:

> High-dimensional representations of queries and documents relevant to them lie in a query-dependent lower-dimensional manifold of the representation space.

If the MC hypothesis holds, there is a query-dependent low-dimensional manifold in the latent space where retrieval is more effective, as the query and its relevant documents are clustered better than in the original latent space.

While the optimal manifold can have an arbitrary shape, finding it is a computationally challenging problem. Therefore, we reduce the search space of the optimal manifold only to *linear subspaces* of the original latent space, one for each query. In other words, we assume it is possible to devise a subset of the dimensions of the latent space, optimal to represent the query and the documents and discard the others. As we will see in the following, this assumption, despite being a first approximation, is highly effective and allows us to formulate several efficient heuristics to determine such linear subspaces. Such an assumption implies that only a subset of dimensions of the latent space is needed to optimize retrieval for a given query, and the other dimensions encode some "noise" which is detrimental to retrieval performance. This makes sense if we consider that the number of dimensions is fixed ahead, independently of the specific queries and documents. During training, learning algorithms globally optimize the disposal of the embeddings

in the latent space by exploiting the relevance relations hidden in the training dataset. Therefore, they try to exploit the full dimensionality anyway, making it extremely unlikely to completely zero out some dimensions, which, instead, would produce the best linear subspace for a given query.

To provide the first evidence in support of our hypothesis, we put ourselves in an ideal case, and we assume that relevant documents are known beforehand. In this context, we focus on different state-of-the-art dense IR systems [27, 28, 37, 38, 64] and, by relying on several TREC collections (Deep Learning 2019, 2020, DL HARD 2021, and Robust 2004), we show that there exist query-dependent linear subspaces, i.e., a specific type of manifold where dimensions are zeroed, where dense IR system performance considerably improve, moving from 0.123 to 0.351 (+184%) in terms of AP and from 0.334 to 0.649 (+94.2%) in terms of nDCG@10. This made us confident that our hypothesis offers ample room for improving performance.

Then, since known relevant documents are rarely available ahead in operational settings, we design a set of heuristics to estimate which dimensions to retain and which ones to discard, and we call them *Dimension IMportance Estimators (DIMEs)*. Thorough experimentation on the proposed DIMEs with state-of-the-art dense IR systems on various TREC collections show impressive performance improvements: up to +0.117 (+49.6%, moving from 0.236 to 0.353) in AP and +0.210 (+55.9%, moving from 0.376 to 0.586) in nDCG@10.

This work builds upon a previous conference paper [20], extending it in several directions. The extensions include the validation of the generality and robustness of our approach across two further dense representation models: TCT-ColBERT and Dragon. While TCT-ColBERT adopts a standard bi-encoder, Dragon uses asymmetric encoders for documents and queries to which we adapt our DIMEs. Another extension regards the introduction of three novel DIMEs. The first is a basic DIME that randomly selects the dimensions to retain, allowing us to characterize the solution space for our heuristic DIMEs. The second one is based on *query variations*, e.g., rewritings of the original query that should retrieve the same set of documents. The last novel DIME exploits document relevance assessments returned by an LLM. Moreover, we investigate the impact of the specific generative model on the performance of all our LLM-based DIMEs. Specifically, we experiment with DIMEs using two 7B models (Gemma and Mixtral), a 70B model (LLama), and a commercial model (GPT-4). Finally, in this extended paper, we include a novel analysis to determine the optimal fraction of dimensions to retain depending on the retrieval model, collection, and measure considered, and to what extent the optimal dimensions to retain are shared across different queries.

The paper is organised as follows: Section 2 summarizes the related work; Section 3 formalizes our methodology and introduces the different DIMEs used in this paper; Section 4 reports the results of our extensive and reproducible evaluation; finally, Section 5 draws some conclusions and outlines future work.

2 RELATED WORK

Classical IR systems primarily rely on matching query and document terms: the presence of a query term within a document is considered an indicator of relevance. This approach is particularly affected by the semantic gap: a concept can be expressed using different synonyms, and the same term might be polysemous, impairing the effectiveness of matching. With the advent of Neural IR and LLMs, the focus shifted from term matching to semantic matching. The systems based on this novel paradigm take a piece of text, i.e., a query or a document, and project it into a latent space using a neural network. This novel representation can be either sparse, i.e., it contains as many dimensions as the terms in the vocabulary, as in the case of SPLADE [23], or dense. In this paper, we focus on IR systems relying on dense representations. These representations are typically much smaller than the vocabulary size, e.g., a few hundred dimensions, and denser than sparse ones. Dense IR approaches first project documents into a latent space using a

projection function called an "encoder.. Such documents are stored efficiently in a specialized metric index, such as the one offered by the FAISS toolkit [29]. At query time, the query is projected into the latent space as well. The encoder used for the query can either be the same as the one used for the documents or a different one. In this paper, we consider four state-of-the-art dense IR models that encode the query and the documents in the same embedding space (ANCE [64], Contriever [28], TAS-B [27], and tct-ColBERT [38]) and a state-of-the-art approach that relies on two asymmetric encoders for queries and documents, namely Dragon [37]. ANCE is a seminal approach that uses contrastive learning with hard negatives: given a training query, the model is trained by asking it to guess the relevant document between two documents, a relevant one and another document chosen among the top-ranked documents scored by the model itself trained up to that point, i.e., a hard negative. Contriever is also based on contrastive learning and differs from ANCE mainly in how positive and negative examples are chosen. Dense Retriever trained with diverse AuGmentatiON (Dragon) [37] relies on distillation, i.e., using other IR models to devise soft labels. It achieves state-of-the-art performance by operating on two aspects: the order in which teachers are employed and data augmentation. Indeed, Lin et al. [37] observe that a proper ordering of the teachers improves performance. Similarly, Lin et al. [37] highlight that expanding the dataset with additional automatically generated queries is beneficial. Topic Aware Sampling Balanced (TAS-B) is a distillation method based on dual-teacher supervision, where teacher models are BERT Cross-encoder [46] and ColBERT [31]. Furthermore, when constructing batches, it relies on Topic Aware Sampling so that batches contain queries on similar topics. Tightly-Coupled Teacher ColBERT (tct-ColBERT) [38] distils information from ColBERT [31] to devise a dense encoder. This enables performance similar to ColBERT while reducing inference time by allowing precomputation of document embeddings. In particular, the loss function used to train tct-ColBERT considers both hard labels—i.e., whether the document is relevant to the query within the training set—and soft labels produced by ColBERT—i.e., the retrieval score of the document in response to the query.

A closely related area to our proposal is feature selection for machine learning [24, 30, 35, 54]. The objective of feature selection is to isolate a subset of all available features to improve a model's effectiveness while reducing the computational cost. There are several approaches to the feature selection task. Such approaches include the usage of ANalisys Of the VAriance (ANOVA) or chi-squared statistics to determine the importance of each feature [7, 16, 58] and approaches based on correlation or mutual information to determine if some features overlap in terms of provided information [48, 60, 67].

Regarding IR specifically, feature selection approaches have been successfully applied to the Learning-to-Rank task [14, 25, 51, 52]. While our DIMEs can be categorized as feature selection algorithms—if dimensions are treated as features—the key difference is that, in our case, the selected features vary on a query-by-query basis. Major feature selection approaches instead identify a set of features, regardless of the instance on which to apply the machine learning model [54]. A second difference is that, in the classical Learning-to-Rank task, features often have an explicit semantic meaning, i.e., they represent and quantify real-world properties of the data that are interpretable by humans, and such meaning can be exploited to drive the selection procedure. In our case, no dimension has an explicit meaning: the latent semantic meaning is learned and not directly interpretable. Deciding which dimensions to preserve or remove depends on the underlying representation model and can only be done at test time—i.e., when the query is available. Determining if and how current feature selection approaches can be applied for the dimension importance estimation task is left as future work.

Another line of research relevant to our work is related to *Pseudo-Relevance Feedback (PRF)* models. These methods are supported by a long-established and rich body of literature, starting with the Rocchio approach [53]. As a general pattern, PRF approaches operate by introducing additional terms to the query. Such terms can be chosen either by

considering statistics of the terms in pseudo-relevant documents and the corpus [1, 2, 53], or by considering the similarity between the query and the terms in a non-contextualized word-embedding space [18, 33, 55, 56, 65]. Most PRF approaches can be interpreted under a geometric framework. Introducing new words into the query implicitly applies a linear transformation to its representation, shifting it closer to where relevant documents are likely to be. On the other hand, our DIMEs apply a spatial transformation, i.e., a projection to a linear subspace where relevant documents might be closer to the query. Indeed, there are two major differences: i) PRF relies on linear combinations of vectors, i.e., scaling and translations in a representation space, while the MC hypothesis conjectures that projections are the most effective transformations; ii) PRF operates only on the query representation; MC hypothesis is designed to operate the projection both on queries and documents. At the same time, as discussed in Section 4, PRF and the MC hypothesis can also synergize. For example, pseudo-relevant documents can be used to instantiate a DIME that operates under the MC hypothesis. Recent work has begun investigating how to use LLMs to generate PRF documents. In this regard, given a query, Mackie et al. [42] employs several prompts corresponding to different "tasks" to generate different pseudo-relevant outputs. Such pseudo-relevant documents are used to identify the most frequent terms and use them to expand the query. It is important to note that Mackie et al. use their strategy to expand queries that are then processed with lexical IR models, i.e., BM25, thus focusing on a different class of systems compared to the dense IR systems that we consider here.

On a different line, some effort has also been devoted to developing PRF-based approaches specifically tailored for dense IR models. One of the most prominent approaches in this regard is *Vector PRF (VPRF)*, proposed by Li et al. [34]. VPRF, inspired by Rocchio, combines the dense representations of query and pseudo-relevant documents. More in detail, VPRF computes a weighted centroid of the query and document representations that can be used as an expanded query. Assuming ϕ is a dense encoder, the expanded query representation is computed as $\mathbf{q}^* = \beta_1 \phi(\mathbf{q}) + \frac{\beta_2}{\mathbf{k}} \sum_{\mathbf{i=1}}^{\mathbf{k}} \phi(\mathbf{d_i})$, where β_1 and β_2 are two parameters that regulate the contribution of the original query and pseudo-relevant feedback respectively, while k is the number of pseudo-relevant documents considered. Given the similarity in setting, we compare the proposed DIMEs with VPRF in our experimental analysis. Later on, Zhuang et al. [69] extended this approach to employ implicit feedback instead of pseudo-relevance feedback. More in detail, Zhuang et al. develop a counterfactual-based approach, called CoRocchio, to de-bias the click frequencies derived from query logs and use such click frequencies to construct an informed centroid representation to serve as the query. Although the use of implicit feedback makes direct comparison between CoRocchio and DIMEs difficult, we plan to investigate the integration of implicit feedback and DIME in future work.

On a different note, the novel Matryoshka Learning framework [32, 36] focuses explicitly on learning representations that allow for easy and efficient dimensionality reduction. During the learning phase, Matryoshka embeddings are optimised to maximise the similarity between the query representation and the relevant documents, while minimising the similarity between the query and non-relevant documents. Unlike other approaches, Matryoshka Learning employs a loss function that optimizes both the full query-document representation and nested subsets of dimensions. Consequently, using Matryoshka embeddings, it is possible to use only the first dimensions of the representation for a more efficient computation of the query-document similarity. After the first-stage retrieval, the list of retrieved documents can be further refined using the entire representation. According to Kusupati et al. [32] and Li et al. [36], just a few hundred dimensions of the Matryoshka representation achieve performance comparable to the full model. This aligns with the MC hypothesis underlying DIMEs: the full representation can be reduced to a lower-dimensional manifold. Nevertheless, two major differences must be highlighted. First, Matryoshka embeddings are, by construction, such that the most important dimensions are the same for every query, allowing for efficient retrieval by retaining only the first dimensions.

In contrast, DIME operates by "searching" such optimal dimensions on a pre-existing query representation. Secondly, Matryoshka embeddings achieve optimal effectiveness when the full representation is used. DIMEs, on the other hand, achieve optimal effectiveness when only a subset of dimensions is considered.
We hypothesize that Matryoshka Learning and MC hypothesis rely on the assumption that the full representation contains some noise: Matryoshka Learning replaces this noise with redundancy to improve efficiency, whereas DIMEs remove it to enhance effectiveness. The major advantage of DIME over Matryoshka Learning is that the former does not require any training and can be applied at inference time.

3 METHODOLOGY

Relying on our MC hypothesis and the assumption that identifying linear subspaces is effective, in Section 3.1 we formalize the dimension importance estimation framework and introduce our *Dimension IMportance Estimators (DIMEs)*, i.e., efficient methods for assigning query-dependent importance scores to dimensions in the latent representation. These DIMEs allow us to sort the dimensions in decreasing order of estimated importance and to select the most important ones, identifying the query-dependent linear subspace at the basis of our assumption. Specifically, in Section 3.2, we define an oracle DIME to provide experimental evidence in support of the MC hypothesis in an ideal scenario where relevance judgments are known. In Section 3.3, we discuss instead several DIME methods for practical use, i.e., when relevance judgments are not known ahead.

3.1 The Dimension Importance Estimation Framework

Let $\mathbf{q}$ denote a query Q and $\{\mathbf{d}, ...\}$ a corpus of documents $\{D, ...\}$ represented in the latent space $\mathbb{R}^d$ by an encoder ϕ of a dense neural model. Notice that ϕ can either be symmetrical (i.e., the same for the queries and documents) or asymmetrical (i.e., two encoders ϕ_Q and ϕ_D for the queries and the documents, respectively). The IR system takes as input the representations of the query and the documents and produces a ranked list of documents $\langle \mathbf{q}, \{\mathbf{d}, ...\}\rangle$ as output. Let $\mathcal{M}\big(\langle \mathbf{q}, \{\mathbf{d}, ...\}\rangle\big)$, be an evaluation measure to quantify the performance of the IR system for the query $\mathbf{q}$.

Let W denote a subspace of $\mathbb{R}^d$. Furthermore, let π_W be the projection operator that projects a vector from $\mathbb{R}^d$ to W. Our MC hypothesis implies that there exists a query-dependent subspace W that induces higher retrieval effectiveness than the one observed in the original space:

$$\exists W \subset \mathbb{R}^d s.t., \mathcal{M}\big(\langle \pi_W(\mathbf{q}), \{\pi_W(\mathbf{d}), ...\}\rangle\big) > \mathcal{M}\big(\langle \mathbf{q}, \{\mathbf{d}, ...\}\rangle\big), \tag{1}$$

where $\mathcal{M}\big(\langle \pi_W(\mathbf{q}), \{\pi_W(\mathbf{d}), ...\}\rangle\big)$ denotes the evaluation measure when both the query $\mathbf{q}$ and the documents are projected from $\mathbb{R}^d$ onto the subspace W by the corresponding projection operator π_W. If there exists a space W satisfying inequality (1), then MC hypothesis holds (at least for the query represented by $\mathbf{q}$). More in detail, we are interested in finding the optimal subspace W where retrieval performance improves over the full latent space in $\mathbb{R}^d$. Thus, our objective can be formalized as finding the subspace W s.t.:

$$\underset{W \subset \mathbb{R}^d}{\operatorname{argmax}}\, \mathcal{M}\big(\langle \pi_W(\mathbf{q}), \{\pi_W(\mathbf{d}), ...\}\rangle\big), \tag{2}$$

and verify whether retrieval in W is more effective than in $\mathbb{R}^d$.

Exploring any possible linear or nonlinear subspace W is not feasible, since the solution space would be infinite; thus, we need a constructive method to determine an appropriate subspace. Therefore, we assume that a linear subspace is a suitable simplification and that, among all the linear subspaces, we can restrict ourselves to those obtained by zeroing

out one or more dimensions of the representations in $\mathbb{R}^d$. As discussed in Section 1, this assumption is a reasonable first approximation, which might lead to a suboptimal solution, but at the great benefit of a clear and straightforward method to construct W, as we will discuss in the following sections.

Therefore, we specialise the projection operator π_W to π_δ, which removes the components of a vector in $\mathbb{R}^d$ not included in a set of dimensions $\delta \subseteq \{1, \ldots, d\}$, and we rewrite eq. (2) as

$$\underset{\delta \subseteq \{1,\ldots,d\}}{\operatorname{argmax}} \mathcal{M}\big(\langle \pi_\delta(\mathbf{q}), \{\pi_\delta(\mathbf{d}), ...\}\rangle\big). \tag{3}$$

Although Eq. (3) restricts the infinite solution space of Eq. (2) to the finite solution space of finding the best subset of dimensions δ which maximizes $\mathcal{M}$, this is still a huge solution space, corresponding to the power set of the d dimensions, which has cardinality 2^d.

To make the problem computationally tractable, we introduce a "bag-of-dimensions" assumption by considering the contribution of each dimension to retrieval effectiveness independent of the others. Such an assumption allows us to independently choose the dimensions in δ based on *Dimension IMportance Estimators (DIMEs)*, i.e., functions $u_q : \{1, ..., d\}; \theta \rightarrow \mathbb{R}$ that associate to each dimension of $\mathbf{q}$ a score estimating its *importance*. In other words, we assume that if $u_q(i) > u_q(j)$ for two dimensions i and j, then for two sets δ_i and δ_j differing only for the presence of dimension i in δ_i and j in δ_j, $\mathcal{M}\big(\langle \pi_{\delta_i}(\mathbf{q}), \{\pi_{\delta_i}(\mathbf{d}), ..., \}\rangle\big) > \mathcal{M}\big(\langle \pi_{\delta_j}(\mathbf{q}), \{\pi_{\delta_j}(\mathbf{d}), ..., \}\rangle\big)$. The DIMEs can exploit some additional information θ to devise their estimation, e.g., the top retrieved documents, a relevant document, and query variations.

Given the previous assumption, to address the problem in Eq. 3, we can rely on a DIME to score the d dimensions for query $\mathbf{q}$ and simply search for the solution among the prefixes of the list of dimensions ordered by decreasing DIME score. To formulate its importance estimation, a DIME can rely on several possible sources of information, including the query and document representations.

Using DIMEs has two significant advantages: i) it relaxes the task, making it practical; ii) it lets us explore the behavior of the proposed approaches for a varying number of subspace dimensions. It is worth noting that our DIMEs are query-dependent. Our objective is not to find a global ordering of the dimensions that optimizes the effectiveness performance on all queries, but to find a query-dependent ordering in line with the MC hypothesis.

3.2 Oracle DIME

To assess the impact of the MC hypothesis, we propose an estimator that shows the superior retrieval effectiveness achievable by removing some of the dimensions. This oracle DIME uses all relevance-annotated documents, making it unsuitable for operational scenarios. Nevertheless, it is useful to demonstrate that: i) different dimensions have a diverse degree of importance, i.e., the MC hypothesis is well grounded; and ii) there is a large margin of improvement for effectiveness, achievable by properly selecting the correct dimensions in dense IR representations.

Let $\mathcal{R}$ be the list of annotated documents for a given query q. A relevance label r, represented as an integer, is associated with each document in $\mathcal{R}$. Depending on the collection, the integer label can be a binary value or a graded relevance assessment. Without loss of generality, we assume that $|\mathcal{R}| > 2$ and the annotated documents belong to at least two distinct classes (e.g., relevant and non-relevant). Moreover, let $R \in \mathbb{R}^{d \times |\mathcal{R}|}$ be a matrix s.t. $R_{i,j} = \mathbf{q}_i \cdot \mathcal{R}_i^j$. In other terms, the element in the i-th row and j-th column of R is the product of the i-th component of the query representation $\mathbf{q}$ and the i-th component of the representation of the j-th document in $\mathcal{R}$. To assess if a dimension "correlates" positively with the relevance of documents in $\mathcal{R}$, we build the *relevance vector* $\mathbf{r} \in \mathbb{R}^{|\mathcal{R}|}$, where the j-th element is the relevance label of the j-th document in $\mathcal{R}$. For each dimension i, our oracle estimator u_q^{or} measures the

Pearson correlation ρ between the i-th column of R, $R_{:,i}$, and the relevance vector:

$$u_q^{or}(i) = \rho(\mathbf{r}, R_{:,i}). \tag{4}$$

The oracle DIME associates the maximum importance to the dimension whose corresponding column in R correlates the most with the relevance labels. Thus, the better a dimension ranks the documents according to their relevance, the more important it is.

3.3 DIMEs in Practice

Since relevance annotations are unavailable in practice, we now introduce practical DIMEs, which, unlike the oracle DIME of Eq. (4), do not rely on such information. In particular, we identify two main families of DIMEs: those that rely on internal information and those that rely on external information. With "internal information" we refer to what can be derived exclusively by considering the representations of the query and the documents in the corpus. On the other hand, "external information" includes additional sources of information, such as query variations, pseudo-relevant documents not present in the corpus, or the active feedback of the user.

3.3.1 DIME Relying on Internal Information. DIMEs relying on internal information use exclusively the representations of the query and/or the documents to compute the importance of each dimension. Within this family, we include a DIME that correlates the importance with the magnitude of the dimensions of the query representation and a DIME that employs the representation of the top documents retrieved in a pseudo-relevance fashion.

Magnitude DIME. In this case, we assume that the information that allows us to determine the importance of each dimension is already available from the query representation $\mathbf{q}$ itself. Specifically, we hypothesize that the magnitude of each dimension of the query describes how important the dimension is for producing a good ranking. If a dimension is particularly large, it is likely associated with a latent facet that is of great importance to understanding the query. On the other hand, dimensions with small magnitudes are likely to be associated with noise and irrelevant aspects for the query and, therefore, can be neglected. Our magnitude-based DIME heuristic u_q^{mag} for dimension i is thus simply defined as:

$$u_q^{mag}(i) = |\mathbf{q}_i|, \tag{5}$$

where $\mathbf{q}_i$ denotes the i-th component of $\mathbf{q}$. Notice that we consider the absolute value of each element: stating that a query is particularly skewed toward a specific dimension – even in negative terms – should be of great importance to describe the query. A filter based on this heuristic will retain particularly large-magnitude dimensions and discard small-magnitude dimensions.

PRF DIME. This DIME assumes that the top τ retrieved documents are relevant, and the interaction between such documents and the query can provide effective insights on how to identify the most effective dimensions.

More in detail, given the representations $\mathbf{d}@1, \ldots, \mathbf{d}@\tau$ of the top τ documents retrieved for the query q, which are assumed to be pseudo-relevant, we construct the representation $\mathbf{p}$ of a generic pseudo-relevant document as the centroid of the representations of the retrieved documents such that $\mathbf{p}_i = \frac{\sum_{j=1}^{\tau} \mathbf{d}@j_i}{\tau}$. This allows us to instantiate our PRF DIME heuristic u_q^{PRF} for the importance of dimension i as follows:

$$u_q^{PRF}@\tau(i) = \mathbf{q}_i \cdot \mathbf{p}_i. \tag{6}$$

PRF DIME approximates the importance of dimension i as the product between the i-th dimensions of the query and the centroid of the representations of the top τ pseudo-relevant documents. We assume that if the alignment between the query and the archetypal pseudo-relevant document is particularly prominent on a certain dimension, then it is more likely that such a dimension is effective for retrieval and, therefore, should be retained.

Bi-encoder models can be divided into two different families: symmetric bi-encoders that use the same encoder ϕ to encode both the query and the documents (e.g., ANCE, Contriever, TAS-B, tct-ColBERT), and asymmetric bi-encoders that use two different encoders, ϕ_Q and ϕ_D, for the query and the documents, respectively (e.g., Dragon). Our definition of the PRF DIME assumes that the default representation is used: if the bi-encoder is symmetric, then ϕ is used to encode both the query and the documents. Conversely, if the bi-encoder is asymmetric, representations $\mathbf{q}$ are constructed as $\phi_Q(q) = \mathbf{q}$, while documents representations $\mathbf{d}@1, ..., \mathbf{d}@\tau$ are such that $\phi_D(D@1) = \mathbf{d}@1, ..., \phi_D(D@\tau) = \mathbf{d}@\tau$.

However, in the asymmetric case, the use of two different encoders, the queries and the documents live in two different spaces, which might or might not be characterized by the same topological properties. For example, documents might have more uniform representations while queries might have more diversified ones. This might impact which dimensions are considered the most important, potentially leading to detrimental effects. Therefore, we define a second PRF DIME, which we refer to as "symmetrical PRF DIME". To compute it we first obtain the pseudo-relevant documents using the asymmetric encoding—the standard one—then we compute their representation using the query encoder: $\phi_Q(D@1) = \mathbf{d}@1^S, ..., \phi_Q(D@\tau) = \mathbf{d}@\tau^S$. Similar to the standard PRF DIME, we compute $\mathbf{p}^S$, the centroid of the embeddings $\mathbf{d}@1^S, ..., \mathbf{d}@\tau^S$. Finally, the symmetric PRF DIME is defined as:

$$u_q^{PRF\text{-}S}@\tau(i) = \mathbf{q}_i \cdot \mathbf{p}_i^S. \tag{7}$$

A major aspect we wish to stress is that we do not change how documents are retrieved—for that, we still use the asymmetric representation for queries and documents. The change regards only how we estimate the importance of the dimensions. Notice that it makes sense to use the symmetrical PRF DIME—as well as the other symmetrical DIMEs we will define later on—only in the case of asymmetric bi-encoders (Dragon, in our experimental section). Indeed, we would observe no changes if the encoder ϕ is already symmetric.

3.3.2 DIME With External Information. DIMEs based on external information employ external information sources to estimate the importance of each dimension. More in detail, we devise four external information DIMEs: i) a DIME based on query variations; ii) a DIME that uses an LLM-generated pseudo-relevant document; iii) a DIME that employs an LLM as a weak relevance annotator; iv) a DIME that utilizes a relevant document as a form of active feedback.

DIMEs Based on Query Variations. Query variations can provide a useful signal for identifying the most relevant dimensions. Query variations are queries that correspond to the same information need and thus aim at retrieving the same set of documents. Such variations might be specializations or generalizations of the query, contain synonyms or hyperonyms of the query terms, or additional terms. Even though query variations express the same information need, in the most common scenario, they tend to interact with the system and the corpus [3, 4, 13], leading to different results in practice. For example, a certain query might perform poorly, while a reformulation might be extremely effective. This behavior is commonly observed in everyday interactions with IR systems: by simply rewriting the query, we could end up with a better or worse set of retrieved documents.

Query variations have been employed in several tasks meant to improve the system performance, such as query rewriting [8], relevance modeling [39], rank fusion [6], to predict the performance of a system [17, 21, 22, 66], or

measure its stability [13, 49]. We hypothesise that they can also be effectively used to determine which dimensions are the most important in a dense representation. Indeed, if we assume that the dimensions have an underlying latent semantic meaning, all the query variations that correspond to the same information need will insist more on those dimensions that better describe the meaning of the query, while they will contain "noise" on the other dimensions. Therefore, we expect important dimensions to be important for all the variations.

Let $\mathcal{V}_q$ denote the set of query variations that represent the same information need as the query q. We call $\mathbf{v}$ the embedding of a query variation in the latent space. The first estimator is based on computing the *interaction* between the query and a randomly picked variant. We call such a heuristic *Random Query Variation DIME* and refer to it as u_q^{var}. In this case, the weight of each dimension is defined as follows:

$$u_q^{var}(i) = \mathbf{q}_i \cdot \mathbf{v}_i, \tag{8}$$

where $\mathbf{v}$ is the embedding of a query variation v uniformly sampled from $\mathcal{V}$. One of the limitations intrinsic to this approach is that, depending on which query variation is sampled, the results might differ, inducing different performances.

Therefore, to address this limitation, we propose the *Query Variations Centroid DIME*, identified with u_q^{Cvar}. Instead of using a single query variation, this weighting mechanism weights the query representation by the centroid of the representations of its query variations. More formally:

$$u_q^{Cvar}(i) = \mathbf{q}_i \cdot \frac{\sum_{v \in \mathcal{V}_q} \mathbf{v}_i}{|\mathcal{V}_q|} \tag{9}$$

Notice that, mathematically speaking, either employing the centroid of the query variations or using each query variation to compute the importance and then averaging the performance across the query variations is identical.

A limitation of the Query Variations Centroid DIME is that it weights more the query representation than any other variations by a factor of $|\mathcal{V}_q|$. This behaviour might not be desirable: as mentioned before, it is common for query variations to perform better or worse than the original query – by weighting the original query we do not exploit this characteristic.

To avoid weighing more the original formulation, we propose a final DIME within this family that treats the original query as an arbitrary variation. We refer to this estimator as *Query and Query Variations Centroid DIME* and indicate it with u_q^{CQvar}. The weight associated with each dimension according to this DIME is computed as follows:

$$u_q^{CQvar}(i) = \left| \frac{\mathbf{q}_i + \sum_{v \in \mathcal{V}_q} \mathbf{v}_i}{1 + |\mathcal{V}_q|} \right| \tag{10}$$

More in detail, the weight is the absolute value of the i-th dimension of the centroid of all query variations, including the query itself.

Notice that in this case, we employ the absolute value of the weight. Indeed, similarly to the magnitude-based DIME, if one of the dimensions is particularly big in negative terms, it is an important piece of information: the query is opposed to the latent concept underlying that dimension. If we do not employ the absolute value, this piece of information is lost, and a large negative dimension is considered not important.

A major advantage linked to these DIMEs is that it is an easy task to obtain the query variations by simply looking at the query log of a search engine. Following, for example, the procedure adopted to build the UQV 100 dataset [3], it is possible to associate a set of queries with a large set of query variations.

LLM DIME. LLMs are the current state of the art for generating text. Therefore, given a query q, we harness their power to generate an artificial pseudo-relevant document that can be used to determine which dimensions of $\mathbf{q}$ are the most important. More in detail, we employ an LLM to generate an answer A in response to the query, which acts as a pseudo-relevant document. We are not interested in investigating if the answer returned is correct, as it will not be presented to the user but used only for computing the DIME. To avoid introducing any form of bias, we do not perform any prompt engineering: we directly input the verbatim query to the LLM, without any form of preprocessing. Once the text in response to the query has been generated by the LLM, we compute its representation $\mathbf{a}$ in the latent space. Then, the DIME based on LLM feedback u_q^{LLM} is defined as follows:

$$u_q^{LLM}(i) = \mathbf{q}_i \cdot \mathbf{a}_i. \tag{11}$$

The dimension importance is given by the product of the i-th dimension of the representations of the query and the LLM-generated answer.

Also in this case, if the bi-encoder is asymmetric, we need to decide which encoder to use to represent the answer A. Being A a pseudo-relevant document, the most intuitive solution is to use ϕ_D s.t. $\phi_D(A) = \mathbf{a}_i$. Nevertheless, we could consider representing both the query and the pseudo-relevant document using the same encoder ϕ_Q. In this case, we refer to the symmetric representation as $\phi_Q(A) = \mathbf{a}_i^S$. Plugging this representation into Eq. 11, we obtain the "symmetric LLM DIME", which we refer to as $u^{LLM\text{-}S}$. As for the PRF DIME, this change affects only how we estimate the dimensions' importance, but leaves unaltered how we represent documents during the retrieval phase.

LLM Assessor DIME. Both the PRF DIME and LLM DIME present some limitations. The PRF DIME operates under the assumption that the top-τ documents retrieved are relevant and provide a useful signal in determining what are the characteristics of relevant documents. Nevertheless, if the top-τ retrieved documents are not relevant, there is a high risk of pushing up non-relevant content, possibly damaging the quality of the ranked list. On the other hand, the LLM DIME employs a LLM-generated document as an approximation of the relevance signal. Nevertheless, the document will likely have a different structure and term distribution compared to the actual documents contained in the collection. This might represent a limitation, as the representation of the generated pseudo-relevant document might differ from that of actual documents in the collections. Inspired by recent efforts [19, 41, 59] that employ LLMs to devise relevance judgments, we use an LLM to combine the effectiveness of the two approaches. In more detail, we use an LLM to associate each document with a relevance label. Formally, we employ a prompt that combines a query and a document and conditions the probability distribution of the LLM so that it will output one among `not relevant`, `partially relevant`, `relevant`, `highly relevant`, according to the likelihood that the document is relevant to the query. To simulate a real-world scenario while reducing the number of interactions with the LLM—typically expensive both in terms of time and cost—we scan sequentially the list of the top-τ documents retrieved by the IR system. The sequential scan continues until either the LLM generates the `highly relevant` label to describe the relationship between a document and a query, or until all the top-τ documents have been considered. We then consider as the pseudo-relevant document the first document whose LLM-generated label corresponds to the maximum relevance among those assessed by the LLM. Let l be such a document, and let $\mathbf{l}$ be its representation in the latent embedding space. We can define the following DIME:

$$u^{LLMRJ}(i) = \mathbf{q}_i \cdot \mathbf{l}_i$$

This approach operates under the assumption that, by using a more expensive model, the LLM, we can better estimate the relevance of a small subset of documents. Therefore, this might improve over the PRF DIME. Furthermore, unlike

the PRF DIME, this approach has the advantage of not requiring tuning of the parameter τ: thanks to our sequential scan, we can continue until we have found a `highly relevant` document or reached some predefined budget. On the other hand, we rely on in-domain documents as pseudo-relevant in this case, compared to the LLM DIME.

As for both the PRF DIME and LLM DIME, we use the document encoder to encode l into $\mathbf{l}$. Therefore, in line with such approaches, we define a second version of this DIME, the "Symmetric LLM Assessor DIME" ($u^{LLMRJ\text{-}S}$), in which we use the query encoder to encode the document l into $\phi_Q(l) = \mathbf{l}^S$.

Active-Feedback DIME. This DIME constructs upon the LLM DIME, by replacing the document generated by the LLM with an actual, human-assessed, relevant document. This estimator may not be suitable in offline scenarios, as it requires access to a relevant document for each query. Nevertheless, it can be particularly effective for specific use cases or in online situations. Consider, for example, the case in which the user has issued a query to a search engine and has retrieved a set of documents in the form of a *Search Engine Result Page (SERP)*. After inspecting it, the user clicks on a link corresponding to a document they may consider relevant. Such a document can then be used to instantiate a DIME, reorganizing the SERP according to the active feedback provided by the user. Other scenarios where this DIME can be effective include, for example, legal IR, where some relevant previous cases are known, or systematic review.

Let us thus assume to have access to a relevant document in response to a query and let $\mathbf{s}$ be its representation in the latent space. The DIME based on Active-Feedback is defined as follows:

$$u_q^{rel}(i) = \mathbf{q}_i \cdot \mathbf{s}_i. \tag{12}$$

In other terms, the weight of each dimension is the product of the i-th dimension of the relevant document representation and the i-th dimension of the query representation.

While this DIME has a specific area of application, e.g., online retrieval, it is also effective in showing the power of DIMEs in identifying the optimal dimensions. In turn, it represents a sort of middle solution between the superior performance of the oracle DIME and the performance of the other, more practical DIMEs.

As for other DIMEs that use document encoding, u^{rel} assumes we use the default document representation in case of asymmetric bi-encoders. Thus, we can define a second variant $u^{rel\text{-}S}$ which employs the same encoder to represent both the query and the relevant document.

4 EXPERIMENTAL RESULTS

4.1 Operationalizing DIMEs

Our DIMEs produce a score for each dimension, estimating its expected relevance. Therefore, the higher the DIME score, the more likely the dimension is to be relevant and effective in producing a well-ranked list of documents. We thus use each DIME to rank dimensions and perform retrieval using a selected subset, analyzing how this dimensionality reduction impacts performance. In this paper, we are interested in showing that reducing the number of dimensions improves retrieval performance; we leave the task of determining the optimal number of dimensions to retain as future work.

Given a generic DIME u, we project the query onto the top k dimensions. In practice, this means setting the remaining $d - k$ dimensions (i.e., those not in the top k) to zero based on the DIME scores. We then use this modified query representation to rank the documents, while keeping their original representations unchanged. By zeroing out the non-selected dimensions in the query, we ensure that only the retained dimensions contribute to the final query–document similarity score. This use of DIMEs enables seamless integration into existing retrieval pipelines: there is no need to

re-index the collection, as operating solely on the query representations is sufficient. Future operationalizations could skip computations on ignored dimensions, increasing retrieval efficiency.

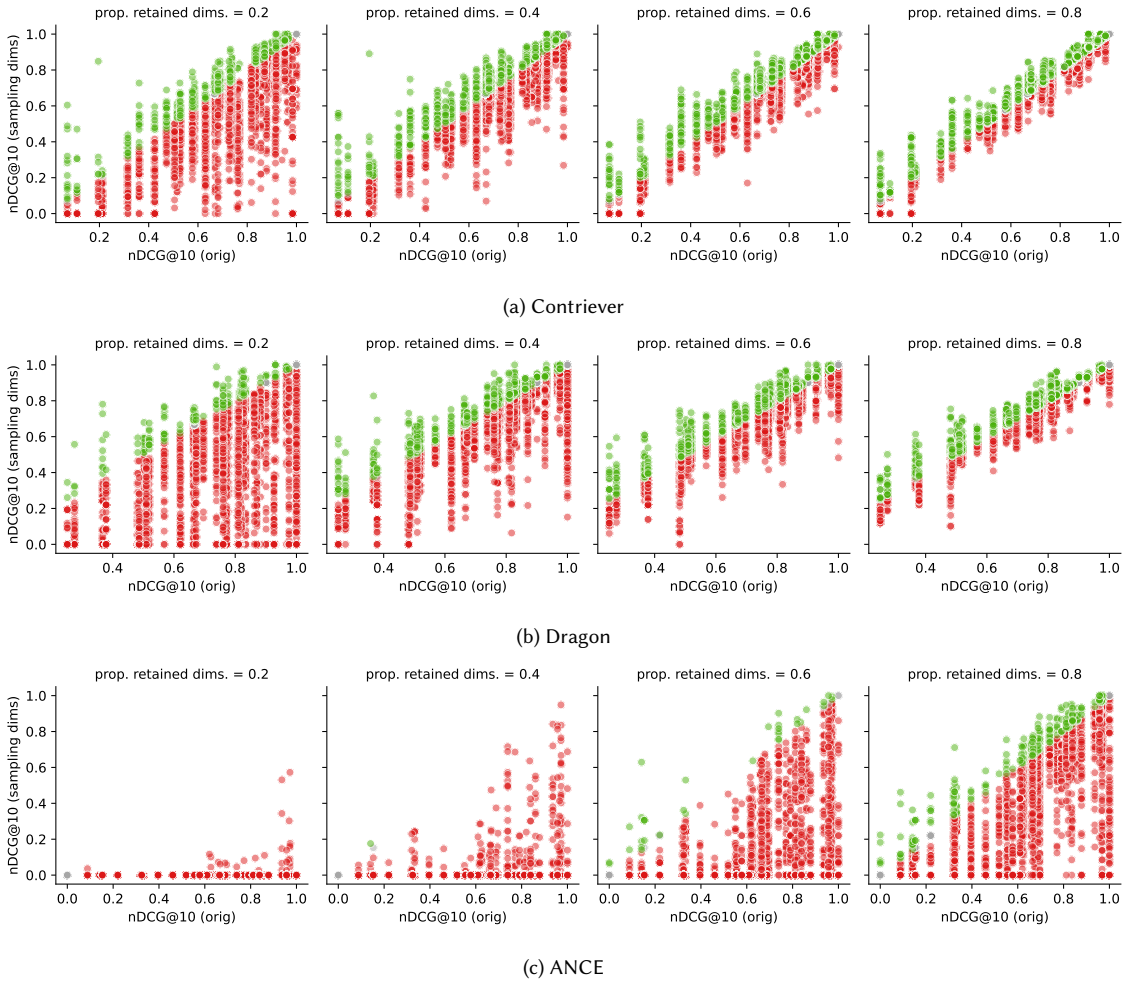
4.2 Experimental Setup

In our experimental analysis[1], we examine five dense retrieval models: ANCE[2] [64], Contriever[3] [28], Dragon[4] [37], TAS-B[5] [27], and tct-ColBERT[6] [38]. This allows us to study the effectiveness of the DIMEs on dense IR models that employ symmetric (ANCE, Contriever, TAS-B, tct-ColBERT) and asymmetric (Dragon) query and document encoders. We use model weights that were fine-tuned on the MS-MARCO collection and are publicly accessible from the Huggingface repository. All models operate in a 768-dimensional latent space. In terms of datasets, we consider four experimental collections: TREC Deep Learning '19 (DL '19) [12], TREC Deep Learning '20 (DL '20) [11], Deep Learning Hard (DL HD) [43], and TREC Robust '04 (RB '04) [63]. The first three datasets focus on ad-hoc passage retrieval, containing 43, 54, and 50 annotated queries, respectively. All are based on the MS MARCO passages collection [45]. RB '04 contains 249 queries and is based on the TIPSTER corpus (disks 4 and 5), excluding the congressional records. All dense IR systems were fine-tuned on the MS MARCO passages collection, making them in-domain for DL '19, DL '20, and DL HD. In contrast, applying them to RB '04 constitutes a zero-shot setting, as its queries and documents differ from the training distribution. Additionally, the RB '04 queries will be considered an "out-of-domain" scenario, as test queries and documents come from a different distribution from the training ones. To instantiate the DIME based on LLMs, we used GPT-4 [47], Gemma and Mixtral with 7B parameters, and LLaMA with 70B parameters. To assess whether improvements over the baseline are statistically significant, we use *ANalysis Of the VAriance (ANOVA)* [57] and Tukey's *Honestly Significant Differences (HSD)* post-hoc test [61] with a significance level of $\alpha = 0.05$.

4.3 Determining if Using Fewer Dimensions is Beneficial for Ranking

4.3.1 Random DIME. To provide initial empirical evidence supporting our MC hypothesis, we evaluate how performance changes when using a random subset of dimensions. Figure 1 presents this analysis for three IR systems: Contriever (Fig.1a), Dragon (Fig.1b), and ANCE (Fig. 1c). We vary the size of the sampled subset of dimensions in each case. The sampling is repeated 100 times. Each dot represents the performance of a random subset of dimensions on a single query. The x-axis shows the original nDCG@10 score, and the y-axis shows the score when using only the sampled subset. For readability, we do not distinguish between individual queries in the figure; hence, vertical bands represent multiple samples for the same query. By construction, points above the diagonal line (shown in green) indicate an improvement over the original result, while points below the diagonal (shown in red) represent a performance drop. A key observation is that, for every query, many random subsets result in improved performance over the original. For Contriever and Dragon, some of these subsets include as few as 20% of the dimensions, whereas for ANCE, at least 80% are required to achieve improvements. Another pattern is that the maximum improvement tends to be higher for queries with low initial performance, as shown by the higher green dots on the left side of the plots. Using only 20% of the dimensions significantly increases the risk of performance degradation, as indicated by the lower red dots. As the

[1] source code available at: https://github.com/guglielmof/DIME-SIGIR-2024
[2] https://huggingface.co/sentence-transformers/msmarco-roberta-base-ance-firstp
[3] https://huggingface.co/facebook/contriever-msmarco
[4] https://huggingface.co/facebook/dragon-plus-context-encoder
[5] https://huggingface.co/sentence-transformers/msmarco-distilbert-base-tas-b
[6] https://huggingface.co/castorini/tct_colbert-v2-hnp-msmarco



(a) Contriever

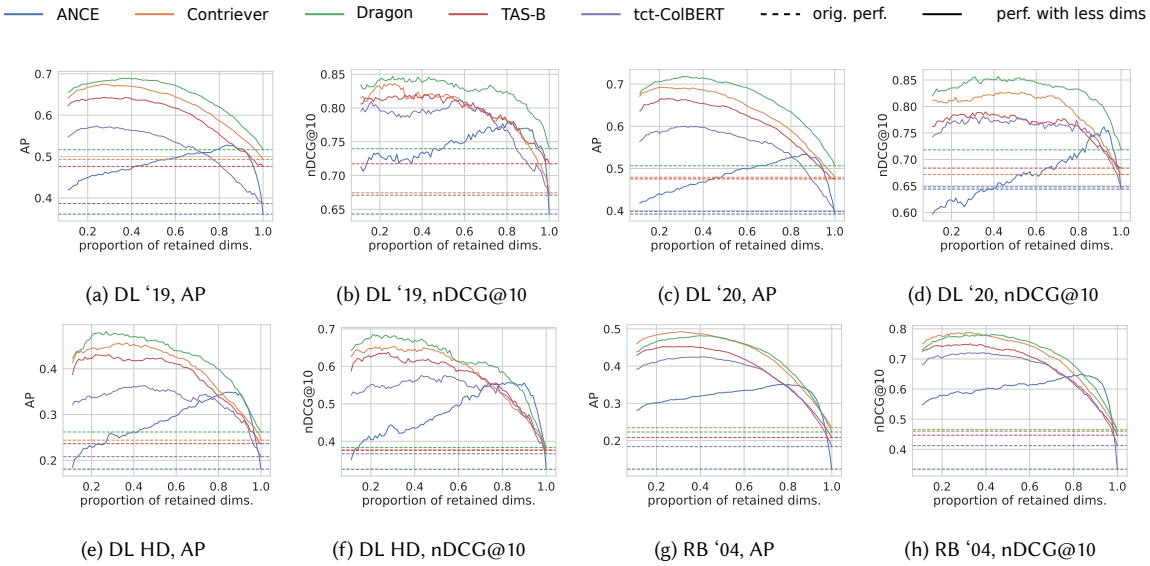
(b) Dragon

(c) ANCE

Fig. 1. Effect of selecting a random sample of the dimensions when varying the sample size. Each dot describes the effect of sampling the dimensions for a single query — vertically aligned dots represent the same query (i.e., the same original performance). Values above the diagonal (green) indicate an improvement, below a decrement (red). Randomly sampling a subset of dimensions in several cases induces an improvement in the performance.

number of used dimensions increases, this risk decreases, and the spread of the dots narrows. When using 100% of the dimensions, all points would lie exactly on the diagonal.

This experiment provides preliminary empirical validation for our MC hypothesis. If we were able to sample the optimal 20% of dimensions for Contriever and Dragon, and 80% for ANCE, performance would improve for every query. Two challenges now arise: i) identifying these optimal dimensions and ii) developing a practical method to select them during retrieval. Interestingly, these results support the hypothesis that not all dimensions contribute equally to ranking quality. If every dimension were equally important, removing any of them would consistently degrade performance. The observed improvements from removing random dimensions suggest that the current representation is suboptimal.



(a) DL '19, AP (b) DL '19, nDCG@10 (c) DL '20, AP (d) DL '20, nDCG@10

(e) DL HD, AP (f) DL HD, nDCG@10 (g) RB '04, AP (h) RB '04, nDCG@10

Fig. 2. Retrieval performance using our oracle DIME when varying the fraction of retained dimensions. Horizontal dashed lines correspond to the performance of baseline models that use all representation dimensions.

Therefore, we aim to guide it toward a more effective form, motivating the development of the DIMEs described in the remainder of this work.

4.3.2 Oracle Dimension Identification. To estimate the upper bound of potential performance improvement achievable through optimal dimension selection, Figure 2 show the performance of the oracle DIME, presented in Subsection 3.2. For each possible configuration of collection/measure, we compute the performance of the dense IR system when considering only the first k dimensions sorted according to the DIME, ranging from 10% to 100% of the total dimensions, in 1% increments (i.e., 90 different cutoffs). For example, given a representation in $\mathbb{R}^{768}$, we start with the top 77 dimensions identified by the DIME and continue adding 1% of the dimensions (~8) at each step. Using 100% of the dimensions corresponds to employing the model without any dimension importance estimation. We notice that Contriever, Dragon, tct-ColBERT, and TAS-B exhibit similar behaviour, while ANCE displays a distinct pattern across scenarios.

Contriever, Dragon, tct-ColBERT and TAS-B. For all these systems, the oracle DIME exhibits impressive performance improvement even if only 10% of the dimensions are retained in all scenarios (i.e., curves are much higher than the dashed baseline). Retaining more than 10% of the dimensions leads to further improvement in almost all cases. The performance follows a convex trend: it starts low, peaks as more dimensions are added, and then declines. This indicates that the subsequent dimensions provide the IR system with additional relevance signals useful for increasing ranking quality.

Patterns are generally more stable for *Average Precision (AP)*, as small changes in the ranking are less likely to cause significant shifts in the AP value—the only exception to this is DL HD. Due to its shallow pooling, DL HD is more sensitive to small changes in the ranking list. Similarly, the behavior is more stable for RB '04: this might occur because i) it has more queries; ii) this is an out-of-domain collection (we will provide more details on this later on). In contrast, the behavior tends to be less stable with nDCG@10: even small changes in the top 10 ranked documents might impact the

performance. Notice that, generally speaking, fluctuations between two consecutive points in the curves are typically small and unlikely to be statistically significant. As for AP, in almost all scenarios, for Contriever, Dragon, tct-ColBERT, and TAS-B, the performance peaks when only 20-40% of the dimensions are considered. After that, the performance steadily decreases, reaching its lowest point when 100% of the dimensions are considered: the worst performance occurs under the current standard usage of these models, without dimension importance estimation. For nDCG@10 (and AP on DL HD), we notice a small initial increase in almost all cases, then a plateau, more evident for DL '19 and DL '20, that continues until 60-80% of the dimensions are considered. As for AP, after that, we observe a monotonic decrease until we reach the full-size representation.

We now present quantitative results highlighting the significant performance improvements enabled by the oracle DIME. The maximum absolute improvement in AP is up to +0.258 (+116%) for Dragon on RB '04 with 40% dimensions. Similarly, for nDCG@10, dimension pruning yields a maximum absolute improvement of +0.322 (+69.2%) when using Contriever for RB '04 queries with 31% dimensions. In terms of maximum relative improvement, AP increases by +131% when using 42% of the dimensions for tct-ColBERT on RB '04 and +78.7% nDCG@10 when using 21% of the dimensions for Dragon on DL HD.

ANCE. ANCE exhibits a different pattern.In most cases, retaining only 10% of the dimensions does not yield significant improvements. For instance, in the case of nDCG@10 on DL '20, it even leads to a performance drop. In all cases, the performance continues to grow from 10% of the dimensions up to 90% of the dimensions. This indicates that, in general, for ANCE, the information about relevance is distributed across multiple dimensions. Then, approximately the last 10% of the dimensions are extremely harmful to ANCE, with a severe drop in performance. Without dimension importance estimation, these harmful dimensions remain mixed in with the rest, introducing noise that impairs model quality. Despite the different behavior, the oracle DIME still yields substantial performance improvements with ANCE: +0.228 (+184%) in AP on DL HD; +0.315 (+94.2%) in nDCG@10 on RB '04.

DIME on out-of-domain collections. An interesting pattern that can be observed is the difference in the behaviour of the models under dimension importance pruning when applied to in-domain and out-of-domain collections. On the out-of-domain collection, RB '04, we observe larger performance improvements. In contrast, improvements on the in-domain collections—DL '19, DL '20, and DL HD—are more variable. When a dense IR model is applied in a zero-shot fashion on an out-of-domain collection, the least important dimensions are extremely harmful. This is reasonable: being an out-of-domain collection, the documents highly differ from those typically used to train the representation. This suggests that when such documents are fed to the encoder, their unfamiliar characteristics are encoded as noise within the representations: by choosing properly the dimensions, we can remove such noise.

The fact that using 100% of the dimensions results in the worst performance suggests not only that dimensions are not equally informative for ranking, but also that many may actively degrade performance. Identifying and removing such dimensions could lead to significant improvements in ranking performance.

4.4 DIME Relying on Internal Information

4.4.1 Magnitude DIME. We report AP and nDCG@10 results for varying numbers of retained dimensions, ranging from 20% to 100% (i.e., the full representation), in 20% increments. This DIME does not yield any notable improvement over the baselines. When there is an improvement, it occurs at the third decimal place and is not statistically significant. These results suggest that a dimension's magnitude is not a reliable indicator of its importance. In other words, dimensions might have been weighted highly by the representation model, but without being particularly relevant for document

Table 1. Retrieval performance of the considered IR models, when u^{mag} is used to identify the most informative dimensions. While in some cases we observe a slight improvement over the baseline, such an improvement is never statistically significant.

	AP					nDCG@10					AP					nDCG@10				
Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
	DL '19										DL '20									
ANCE	0.047	0.240	0.319	0.354	**0.361**	0.144	0.533	0.620	**0.645**	0.643	0.106	0.281	0.357	0.389	**0.392**	0.217	0.52	0.604	0.638	**0.644**
Contriever	0.463	0.486	0.492	**0.494**	0.493	0.648	0.672	0.676	**0.677**	0.674	0.463	0.478	**0.479**	**0.479**	**0.479**	0.66	0.663	0.664	0.671	**0.672**
Dragon	0.467	0.506	0.512	**0.517**	**0.517**	0.709	0.740	0.740	**0.745**	0.740	0.466	0.497	0.502	0.505	**0.506**	0.687	0.708	0.716	**0.718**	**0.718**
TAS-B	0.45	0.467	0.472	**0.476**	**0.476**	0.692	0.711	0.715	**0.719**	0.717	0.440	0.465	0.472	**0.475**	**0.475**	0.658	0.678	**0.688**	0.685	0.684
tct-ColBERT	0.353	0.383	0.386	**0.387**	**0.387**	0.611	0.648	**0.674**	0.671	0.671	0.332	0.38	0.392	**0.399**	0.398	0.574	0.630	0.642	**0.648**	**0.648**
	DL HD										RB '04									
ANCE	0.023	0.126	0.164	0.180	**0.181**	0.072	0.284	0.321	**0.326**	0.325	0.020	0.078	0.111	0.120	**0.124**	0.08	0.240	0.307	0.325	**0.334**
Contriever	0.222	0.238	0.239	**0.246**	0.244	0.362	0.373	0.376	**0.379**	0.377	0.218	0.23	0.232	0.234	**0.235**	0.446	0.456	0.462	0.462	**0.465**
Dragon	0.237	0.260	0.257	0.260	**0.262**	0.363	**0.394**	0.383	0.379	0.384	0.191	0.215	0.221	**0.223**	**0.223**	0.417	0.449	0.459	0.460	**0.461**
TAS-B	0.211	0.228	0.232	**0.236**	**0.236**	0.335	**0.377**	0.375	0.374	0.376	0.186	0.2	0.206	**0.208**	**0.208**	0.410	0.431	0.444	0.446	**0.447**
tct-ColBERT	0.173	0.197	0.206	0.207	**0.208**	0.321	0.346	0.364	**0.368**	0.367	0.150	0.177	0.182	**0.184**	**0.184**	0.364	0.398	0.410	**0.414**	0.412

ranking. Conversely, some dimensions may be assigned a low weight by the encoder, yet still play an important role in ranking. However, even with this basic DIME, we obtain effectiveness figures comparable with the baseline by using about 40-60% of the representation dimensions.

This DIME relies solely on the query representation, which may not provide enough information to identify the most relevant dimensions. Therefore, it is likely that additional information is needed beyond the representation of the query to estimate dimension importance.

4.4.2 PRF DIME. Table 2 reports the results in terms of AP and nDCG@10 when filtering the dimensions using the PRF DIME described in Subsection 3.3. First, we observe that regardless of the setup — i.e., the IR model, collection, and measure considered — it is almost always possible to find at least one DIMEthat outperforms the baseline (i.e., using 100% of dimensions) for some fraction of retained dimensions. However, such improvements are not always statistically significant. The magnitude of improvement depends on various factors, including the collection, evaluation measure, and IR system used. Contriever and TAS-B consistently show improvements, whereas results for tct-ColBERT are variable. For ANCE, any improvements only occur when at least 80% of the dimensions are retained, and these are never statistically significant. Finally, for Dragon, this DIME consistently fails to improve performance.

Dragon and the Symmetric PRF DIME. Using the standard PRF DIME on Dragon does not yield any improvement, unlike the other systems. Nevertheless, both the random and oracle experiments show that Dragon benefits from the dimensionality pruning as much as any other system. As previously mentioned, Dragon differs from the other systems as it employs asymmetric query-document encoders. Therefore, we can test the effectiveness of the symmetric PRF DIME (Eq. 7), which uses the query encoder also to encode the top-τ pseudo-relevant documents. The performance achieved by such DIME on Dragon is reported in Table 3. Interestingly, this change in encoding strategy leads to a substantial improvement in the effectiveness of DIME, aligning the results with those observed also for the other dense IR models. In particular: i) in almost all scenarios, we notice an improvement compared to the baseline that uses 100% of the dimensions; ii) this improvement often occurs for large numbers of retained dimensions (60-80%); iii) the improvement is significant only in half of the cases.

Understanding why the approach based on the symmetric encoder is more effective than the approach based on the standard document encoder requires investigating the distribution of the vectors within the embedding space. In particular, we observe that, for Dragon, the average cosine similarity between pairs of queries when encoded using

Table 2. Performance of the PRF DIME at various τ. In bold, the best performance observed for each triple IR system, test collection, and evaluation measure. Values marked with * are a statistically significant improvement over the baseline using all the dimensions (corresponding to Retained = 1). In almost all cases this heuristic allows for improving the performance. The exception is Dragon where results are always worse.

		AP					nDCG@10					AP					nDCG@10				
	Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
		DL '19										DL '20									
ANCE	u^{PRF}@1	0.033	0.253	0.340	**0.370**		0.082	0.553	0.636	**0.650**		0.083	0.285	0.365	0.389		0.175	0.539	0.609	**0.644**	
	u^{PRF}@2	0.036	0.257	0.341	**0.370**	0.361	0.095	0.565	0.637	0.649	0.643	0.083	0.288	0.366	0.388	**0.392**	0.168	0.543	0.613	0.643	**0.644**
	u^{PRF}@5	0.034	0.257	0.344	**0.370**		0.087	0.563	0.635	0.648		0.077	0.289	0.364	0.391		0.147	0.542	0.613	**0.644**	
Contriever	u^{PRF}@1	0.483	0.503	0.507	0.507		0.676	0.683	0.687	**0.689**		0.488	**0.497***	**0.497***	0.494		**0.711***	0.704*	0.703*	0.693	
	u^{PRF}@2	0.493	0.503	0.508	0.507	0.493	0.672	0.675	0.679	0.685	0.674	0.479	0.488	0.494	0.494	0.479	0.681	0.685	0.687	0.685	0.672
	u^{PRF}@5	0.491	0.503	**0.511**	0.509		0.647	0.664	0.679	0.681		0.488	0.494*	0.496*	0.494*		0.698*	0.686	0.690	0.685	
Dragon	u^{PRF}@1	0.407	0.460	0.478	0.497		0.699	0.713	0.716	0.736		0.428	0.465	0.476	0.487		0.676	0.704	0.708	0.708	
	u^{PRF}@2	0.411	0.458	0.476	0.497	**0.517**	0.699	0.716	0.717	0.735	**0.740**	0.426	0.467	0.477	0.488	**0.506**	0.676	0.702	0.702	0.711	**0.718**
	u^{PRF}@5	0.410	0.460	0.477	0.497		0.701	0.712	0.722	0.732		0.420	0.466	0.470	0.485		0.664	0.691	0.689	0.705	
TAS-B	u^{PRF}@1	0.487	0.506*	0.505*	0.503*		0.720	0.732	**0.733**	0.729		0.465	0.484	**0.490**	0.489		0.698	0.702	**0.712***	0.705	
	u^{PRF}@2	0.492	0.507*	**0.508***	0.503*	0.476	0.719	0.731	0.731	0.725	0.717	0.466	0.481	0.488	0.487	0.475	0.684	0.697	0.709*	0.707*	0.684
	u^{PRF}@5	0.499*	0.505*	0.507*	0.505*		0.712	0.725	0.722	0.724		0.468	0.479	0.488	0.488		0.685	0.685	0.694	0.697	
tct-ColBERT	u^{PRF}@1	0.390	**0.413***	**0.413***	0.408*		0.630	0.643	0.659	0.658		0.381	0.418	0.428*	0.425		0.632	0.671	0.677	0.675	
	u^{PRF}@2	0.387	0.407*	0.410*	0.406	0.387	0.628	0.638	0.651	0.655	**0.671**	0.387	0.422	0.432*	0.433*	0.398	0.636	0.680	**0.684**	0.680	0.648
	u^{PRF}@5	0.390	0.411*	0.412*	0.406		0.634	0.659	0.662	0.661		0.385	0.423	**0.435***	0.433*		0.635	0.676	0.676	0.681	
		DL HD										RB '04									
ANCE	u^{PRF}@1	0.021	0.126	0.175	0.180		0.059	0.272	0.331	0.329		0.017	0.085	0.119	**0.126**		0.068	0.263	0.319	**0.338**	
	u^{PRF}@2	0.018	0.129	0.170	0.182	0.181	0.048	0.268	0.327	0.331	0.325	0.016	0.082	0.120	0.125	0.124	0.062	0.251	0.321	0.335	0.334
	u^{PRF}@5	0.017	0.126	0.173	**0.184**		0.046	0.275	**0.334**	0.332		0.016	0.081	0.119	0.125		0.059	0.247	0.321	0.333	
Contriever	u^{PRF}@1	0.241	0.250	0.253	**0.255**		0.387	0.386	0.388	0.390		0.243	0.256*	0.259*	0.259*		0.483*	0.493*	0.495*	0.494*	
	u^{PRF}@2	0.248	**0.255**	0.254	0.254	0.244	**0.396**	0.395	0.387	0.389	0.377	0.252*	0.262*	**0.264***	0.262*	0.235	0.489*	0.495*	**0.497***	0.494*	0.465
	u^{PRF}@5	0.248	0.248	0.253	0.253		0.379	0.378	0.389	0.387		0.249*	0.258*	0.259*	0.257*		0.476	0.479*	0.479*	0.475	
Dragon	u^{PRF}@1	0.198	0.226	0.235	0.244		0.356	0.367	0.376	0.382		0.166	0.182	0.191	0.202		0.4	0.406	0.419	0.434	
	u^{PRF}@2	0.195	0.229	0.238	0.246	**0.262**	0.349	0.369	0.377	0.382	**0.384**	0.167	0.182	0.191	0.203	**0.223**	0.395	0.409	0.421	0.436	**0.461**
	u^{PRF}@5	0.198	0.233	0.235	0.240		0.354	0.377	0.378	0.372		0.167	0.182	0.190	0.202		0.395	0.404	0.417	0.437	
TAS-B	u^{PRF}@1	0.224	0.239	0.240	0.238		0.359	0.374	0.382	0.375		0.212	0.223*	0.226*	0.226*		0.432	0.449	0.450	0.455	
	u^{PRF}@2	0.224	0.234	0.234	0.237	0.236	0.350	0.373	0.374	0.374	0.376	0.222*	0.230*	**0.231***	0.229*	0.208	0.459	0.468*	0.468*	0.467*	0.447
	u^{PRF}@5	0.236	0.243	0.250	**0.255**		0.369	0.385	0.391	**0.395**		0.224*	**0.231***	**0.231***	**0.231***		0.463*	0.467*	0.468*	**0.469***	
tct-ColBERT	u^{PRF}@1	0.214	0.222	**0.228**	0.226		0.361	0.368	**0.377**	**0.377**		0.176	0.201*	0.206*	0.205*		0.394	0.425	0.434*	**0.437***	
	u^{PRF}@2	0.203	0.223	**0.228**	0.227	0.208	0.340	0.365	0.375	0.374	0.367	0.183	0.204*	**0.208***	0.207*	0.184	0.405	0.427	0.436*	**0.437***	0.412
	u^{PRF}@5	0.205	0.215	0.221	0.219		0.339	0.360	0.361	0.360		0.187	0.206*	**0.208***	0.205*		0.412	0.431*	0.436*	0.434*	

Table 3. Performance of the symmetric PRF DIME on Dragon at various τ. Using the query encoder for both the query and the documents to decide which dimensions are relevant is more effective than using the original Dragon encoders.

		AP					nDCG@10					AP					nDCG@10				
	Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
		DL '19										DL '20									
Dragon	u^{PRF-S}@1	0.516	0.540	0.548*	**0.549***		0.746	0.745	0.757	**0.763**		0.489	0.511	0.515	0.515		0.706	0.714	0.720	0.720	
	u^{PRF-S}@2	0.510	0.545*	0.544*	0.543*	0.517	0.740	0.744	0.759	0.757	0.740	0.494	0.518	0.517	0.519	0.506	0.716	**0.727**	0.719	0.725	0.718
	u^{PRF-S}@5	0.527	0.545*	0.548*	0.547*		0.750	0.749	0.751	0.758		0.498	**0.523***	0.520	0.519		0.704	0.721	0.715	0.718	
		DL HD										RB '04									
Dragon	u^{PRF-S}@1	0.253	0.260	0.265	0.269		0.390	0.389	0.392	0.394		0.194	0.218	0.223	0.226		0.426	0.447	0.454	0.459	
	u^{PRF-S}@2	0.261	0.270	0.271	0.275	0.262	0.394	0.389	0.389	**0.399**	0.384	0.204	0.223	0.227	0.229	0.223	0.444	0.453	0.458	0.454	**0.461**
	u^{PRF-S}@5	0.275	0.278	0.277	**0.279**		**0.399**	0.397	0.392	0.398		0.209	0.228	0.232*	**0.233***		0.437	0.449	0.451	0.454	

the query encoder ϕ_Q is 0.488, indicating that, as expected, queries tend to lie on different subspaces characterized by different relevant dimensions. In contrast, when encoding a random pair of documents using the document encoder ϕ_D, the average cosine similarity is 0.989: the documents tend to live in a very narrow hyper-cone with overall narrow angles. If we consider documents annotated for the same query, both relevant and non-relevant, this similarity grows up to 0.992. This is a relatively small change, considering the shift from completely uncorrelated documents to ones on the same topic—even though possibly not relevant. Finally, when considering only relevant documents, the average similarity is 0.995—again, a small change given that now the documents should be very highly correlated. For comparison, consider that for Contriever, the average similarity between query representations is 0.235, the average similarity between

Table 4. Performance of the query variation-based DIMEs on the RB '04 collection. Except for ANCE, where the improvement is present but not significant, we always observe at least one DIME that induces a statistically significant improvement.

		AP					nDCG@10				
	Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
ANCE	u^{var}	0.023	0.089	0.125	0.129		0.081	0.269	0.340	0.347	
	u^{Cvar}	0.021	0.088	0.125	0.129	0.124	0.080	0.262	0.339	0.344	0.334
	u^{CQvar}	0.021	0.096	**0.137**	0.136		0.077	0.289	**0.364**	0.362	
Contriever	u^{var}	0.241	0.251*	0.252*	0.250*		0.488*	0.500*	0.497*	0.490*	
	u^{Cvar}	0.251*	0.256*	0.256*	0.252*	0.235	0.505*	0.501*	0.498*	0.494*	0.465
	u^{CQvar}	0.264*	**0.267***	0.261*	0.249*		**0.531***	**0.531***	0.509*	0.489*	
Dragon	u^{var}	0.212	0.235*	0.241*	0.239*		0.456	0.482*	0.486*	0.486*	
	u^{Cvar}	0.223	0.244*	0.243*	0.240*	0.223	0.475	0.500*	0.495*	0.489*	0.461
	u^{CQvar}	0.235*	**0.255***	0.250*	0.238*		0.506*	**0.521***	0.502*	0.483*	
TAS-B	u^{var}	0.203	0.221*	0.224*	0.223*		0.443	0.470*	0.477*	0.472*	
	u^{Cvar}	0.213	0.225*	0.226*	0.222*	0.208	0.460	0.475*	0.479*	0.471*	0.447
	u^{CQvar}	0.220*	**0.236***	0.232*	0.221*		0.481*	**0.496***	0.482*	0.469*	
tct-ColBERT	u^{var}	0.166	0.196*	0.200*	0.197*		0.401	0.445*	0.451*	0.442*	
	u^{Cvar}	0.177	0.203*	0.203*	0.199*	0.184	0.409	0.449*	0.447*	0.443*	0.412
	u^{CQvar}	0.183	**0.215***	0.211*	0.198*		0.430	**0.473***	0.465*	0.442*	

random documents is 0.237 (almost identical to the average similarity between queries), while the average similarity between documents relevant to the same query is 0.723. The single encoder used by Contriever effectively captures the subspaces where closely related documents reside. This suggests that the query encoder is optimized to diversify the dimensions on which the represented entities lie. For the same reason, it is also the most effective encoder to instantiate the DIMEs, as it gives more importance to the different dimensions. Moreover, the high similarity between entities encoded by the document encoder suggests that it is ineffective for identifying the most relevant dimensions. If documents appear nearly identical across dimensions, then even having a pseudo-relevant—or truly relevant—document provides little insight into the subspace where other relevant documents may lie.

4.5 DIME With External Knowledge

4.5.1 DIMEs Based on Query Variations. To assess the performance of the query variations-based DIMEs, we use the UQV 100 collection [3]. We selected this collection because it contains human-authored reformulations of the RB '04's queries. Our objective is to determine whether real query variations have a positive impact on identifying the optimal dimensions; the analysis of the role of automatic query variation generation approaches is left as future work. This limitation confines our analysis to the RB '04 collection, as, at the current time, no human-made query variations are available for the DL '19, DL '20, and DL HD collections.

Table 4 reports the results of our three query variations-based DIMEs. For each retrieval model, rows represent the Random Query Variation (eq. 8, u^{var}), Query Variations Centroid (eq. 9, u^{Cvar}), and Query and Query Variations Centroid (eq. 10, u^{CQvar}) DIMEs. These DIMEs yield substantial improvement over the PRF DIMEs. The magnitude of the improvement and its significance depend on the experimental setting considered. In particular, if we consider Contriever, Dragon, TAS-B, and tct-ColBERT, when more than 20% of dimensions are retained, we always have a

statistically significant improvement regardless of the DIME considered. Nevertheless, the optimal improvement is given by the u^{CQvar} estimator. This improvement is approximately 0.03 (ranging from +13.4% to +16.8%) AP points for all IR models and 0.05-0.06 (ranging from +11.0% to +14.8%) nDCG@10 points.

ANCE. If we consider ANCE, the best performance is achieved by the Query and Query Variations Centroid (u^{CQvar}). When 60% of the most important dimensions according to this DIME are retained, both for AP and nDCG@10, the performance is above the baseline (i.e., when 100% of the dimensions are retained), but the improvement is not significant. An identical pattern is also observed for the other two DIMEs, although in this case, the improvement is smaller. It is also important to notice that ANCE is the least performing approach: it is likely that, regardless of which dimensions are considered, the representations likely contain too much noise to allow this model to perform on a par with the others.

Other IR models. Moving beyond ANCE, we notice that, on all other scenarios, these DIMEs consistently yield statistically significant improvements over the 100% dimensions baselines. As a general pattern, the Random Query Variation DIME is overcome by the Query Variations Centroid DIME, which is, in turn, surpassed by the Query and Query Variations Centroid DIME. This supports our hypothesis that individual query variations are less informative than the combined set of variations. It also supports our hypothesis that the query should be treated as any arbitrary query variation. Nevertheless, even a single query variation can lead to significant improvements when using the Random Query Variation DIME by retaining at least 40% of the dimensions.

In terms of the number of dimensions to be retained, we notice that the improvement is always significant when 40% of the dimensions are retained. Indeed, using 40% of the dimension is also the optimal strategy for both the Query Variations Centroid and for the Query and Query Variations Centroid DIME. Conversely, for the Random Query Variations DIME, the optimal dimension is 60% of the dimensions.

In absolute terms, the largest improvement is observed when using the Query and Query Variations Centroid with 40% to pick the dimensions of Contriever evaluated using nDCG@10, with an increase of +0.066 (+14.2%). In percentage, the improvement ranges from 11.0% (+0.049 nDCG points) for TAS-B evaluated using nDCG@10, to 16.8% (+0.031 nDCG) in the case of tct-ColBERT using AP as evaluation measure.

4.5.2 DIME Based on Large Language Models. Table 5 reports the performance of the LLM DIME for the different experimental settings considered in this paper. We notice that, in line with what we observed for the PRF DIME, excluding Dragon, there is always a dimension cutoff that yields improved performance over the baseline, where all the dimensions are taken into consideration. Notably, when LLaMA or GPT-4 is used, improvements are often observed even with just 20% of the dimensions retained. The optimal cutoff depends on several factors, such as the considered system, LLM, dataset, and measure.

While all the LLMs tend to provide an improvement over the baseline, whether such improvement is significant or not depends on several factors. As a general trend, the 7B models, Gemma and Mixtral, tend to be less effective, with the latter being slightly better than the former in most cases. This is reasonable: being smaller models, it is likely that the quality of the generated pseudo-relevant document is lower. For both models, the greatest improvement is achieved when 80% of the dimensions are retained. This suggests that while these models are effective in recognizing harmful dimensions, i.e., the worst 20% of dimensions that should not be used, they are not equally effective in recognizing good dimensions. In contrast, LLaMA and GPT-4 are significantly more effective. In particular, for Contriever, TAS-B, and tct-ColBERT, LLaMA and GPT-4 provide improvements even when only 20% of the dimensions are considered. In

Table 5. Performance achieved by the LLM DIME, using different LLMs. 7B models (Gemma and Mixtral) tend to perform worse, while the 70B model (LLama) and GPT4 are more effective with relatively similar effectiveness.

		AP					nDCG@10					AP					nDCG@10				
	Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
		DL '19										DL '20									
ANCE	u^{LLM}-Gemma	0.025	0.236	0.338	0.370	0.361	0.062	0.517	0.636	0.647	0.643	0.074	0.281	0.370	**0.398**	0.392	0.128	0.534	0.629	0.650	0.644
	u^{LLM}-Mixtral	0.032	0.261	0.355	0.373		0.075	0.537	0.655	0.655		0.066	0.296	0.377	0.397		0.130	0.559	0.630	0.652	
	u^{LLM}-LLama	0.026	0.261	0.355	**0.374**		0.065	0.570	**0.670**	0.656		0.070	0.296	0.377	0.397		0.141	0.550	0.628	0.654	
	u^{LLM}-GPT	0.031	0.260	0.351	0.370		0.081	0.568	0.651	0.664		0.084	0.284	0.374	0.397		0.171	0.537	0.629	**0.655**	
Contriever	u^{LLM}-Gemma	0.473	0.500	0.507	0.509	0.493	0.647	0.672	0.686	0.687	0.674	0.488	0.492	0.498*	0.498*	0.479	0.697	0.687	0.692	0.691	0.672
	u^{LLM}-Mixtral	0.514	0.525*	0.525*	0.521		0.663	0.693	0.705	0.708		0.501*	0.515*	0.512*	0.507*		0.705*	0.712*	0.714*	0.718*	
	u^{LLM}-LLama	0.524	**0.537***	0.534*	0.530*		0.724*	0.746*	0.745*	0.742*		0.506*	**0.518***	0.517*	0.508*		0.714*	0.722*	0.715*	0.712*	
	u^{LLM}-GPT	0.516	0.528*	0.534*	0.527		0.720	0.742*	**0.752***	0.750*		0.503*	0.512*	0.511*	0.504*		0.719*	0.722*	**0.725***	0.710*	
Dragon	u^{LLM}-Gemma	0.393	0.452	0.473	0.492	**0.517**	0.667	0.705	0.721	0.728	**0.740**	0.415	0.469	0.477	0.486	**0.506**	0.662	0.694	0.695	0.708	**0.718**
	u^{LLM}-Mixtral	0.399	0.457	0.476	0.496		0.687	0.702	0.723	0.738		0.427	0.467	0.473	0.490		0.662	0.689	0.694	0.713	
	u^{LLM}-Llama	0.394	0.458	0.476	0.498		0.673	0.705	0.720	0.736		0.431	0.466	0.479	0.487		0.677	0.692	0.697	0.707	
	u^{LLM}-GPT	0.401	0.455	0.475	0.497		0.688	0.708	0.720	0.732		0.432	0.469	0.481	0.487		0.679	0.694	0.700	0.710	
TAS-B	u^{LLM}-Gemma	0.477	0.493	0.503	0.503	0.476	0.679	0.710	0.723	0.725	0.717	0.470	0.494	0.497*	0.496*	0.475	0.697	0.704	0.710	0.709	0.684
	u^{LLM}-Mixtral	0.515*	0.526*	0.521*	0.519*		0.716	0.727	0.733	0.733		0.480	0.489	0.497*	0.496*		0.690	0.691	0.705	0.702	
	u^{LLM}-LLama	0.532*	**0.537***	0.532*	0.525*		0.754*	**0.761***	**0.761***	0.751		0.482	0.505*	0.505*	**0.506***		0.691	0.710	0.706	0.708	
	u^{LLM}-GPT	0.511*	0.529*	0.527*	0.521*		0.747	0.749	0.760*	0.756*		0.483	0.498*	0.501*	0.500*		0.708	0.706	0.710	**0.712**	
tct-ColBERT	u^{LLM}-Gemma	0.368	0.387	0.400	0.399	0.387	0.614	0.638	0.654	0.657	0.671	0.361	0.407	0.418	0.419	0.398	0.596	0.656	0.671	0.683	0.648
	u^{LLM}-Mixtral	0.405	0.419*	0.423*	0.421*		0.653	0.660	0.680	0.684		0.369	0.417	0.420	0.419		0.618	0.660	0.664	0.676	
	u^{LLM}-LLama	0.415	0.434*	**0.438***	0.432*		0.673	0.690	**0.713***	0.711		0.375	0.413	0.421	0.419		0.617	0.658	0.667	0.678	
	u^{LLM}-GPT	0.393	0.421*	0.424*	0.419*		0.669	0.703	0.704*	0.694		0.380	0.423	**0.429***	0.426*		0.645	0.683	**0.689**	0.685	
		DL HD										RB '04									
ANCE	u^{LLM}-Gemma	0.009	0.116	0.169	0.183	0.181	0.027	0.245	0.327	0.326	0.325	0.015	0.080	0.119	0.125	0.124	0.056	0.249	0.332	0.338	0.334
	u^{LLM}-Mixtral	0.013	0.130	0.175	0.185		0.036	0.274	0.339	0.338		0.017	0.086	0.125	0.128		0.068	0.261	0.337	0.341	
	u^{LLM}-LLama	0.011	0.126	0.178	0.184		0.033	0.272	**0.348**	0.339		0.016	0.085	0.124	0.128		0.066	0.268	0.339	0.344	
	u^{LLM}-GPT	0.012	0.129	0.175	**0.186**		0.040	0.284	0.339	**0.348**		0.017	0.086	0.126	**0.130**		0.067	0.267	0.343	**0.347**	
Contriever	u^{LLM}-Gemma	0.228	0.239	0.249	0.254	0.244	0.358	0.366	0.378	0.386	0.377	0.237	0.250*	0.255*	0.253*	0.235	0.486*	0.502*	0.507*	0.504*	0.465
	u^{LLM}-Mixtral	0.239	0.251	0.257	0.260		0.357	0.373	0.389	0.390		0.241	0.253*	0.254*	0.253*		0.479	0.499*	0.499*	0.497*	
	u^{LLM}-LLama	0.254	0.269	0.272*	**0.274***		0.401	0.409	0.409	**0.414***		0.245*	0.259*	0.261*	0.259*		0.501*	0.515*	0.519*	0.513*	
	u^{LLM}-GPT	0.259	0.267	0.270*	0.270*		0.392	0.409	**0.414***	0.412*		0.262*	**0.270***	**0.270***	0.267*		0.528*	**0.534***	0.531*	0.525*	
Dragon	u^{LLM}-Gemma	0.182	0.222	0.232	0.237	**0.262**	0.322	0.363	0.375	0.376	0.384	0.164	0.182	0.189	0.202	**0.223**	0.386	0.400	0.414	0.434	**0.461**
	u^{LLM}-Mixtral	0.197	0.226	0.237	0.247		0.343	0.364	0.379	**0.386**		0.167	0.183	0.191	0.204		0.397	0.403	0.413	0.439	
	u^{LLM}-Llama	0.187	0.231	0.232	0.246		0.327	0.372	0.371	0.383		0.168	0.184	0.192	0.203		0.402	0.406	0.419	0.438	
	u^{LLM}-GPT	0.195	0.228	0.238	0.248		0.343	0.373	0.378	0.384		0.171	0.186	0.194	0.204		0.398	0.414	0.421	0.439	
TAS-B	u^{LLM}-Gemma	0.226	0.236	0.242	0.242	0.236	0.365	0.377	0.389	0.385	0.376	0.205	0.221*	0.222*	0.221*	0.208	0.444	0.470*	0.469*	0.465*	0.447
	u^{LLM}-Mixtral	0.235	0.246	0.246	0.251		0.364	0.373	0.384	0.393		0.213	0.227*	0.227*	0.227*		0.455	0.469*	0.469*	0.473*	
	u^{LLM}-LLama	0.257	0.261	0.261	**0.264**		0.395	0.405	**0.407**	**0.407**		0.219*	0.230*	0.231*	0.229*		0.470*	0.483*	0.485*	0.481*	
	u^{LLM}-GPT	0.243	0.254	0.258	0.250		0.385	0.397	0.401	0.397		0.224*	**0.239***	0.238*	0.236*		0.476*	**0.501***	0.495*	0.489*	
tct-ColBERT	u^{LLM}-Gemma	0.140	0.189	0.202	0.206	0.208	0.256	0.342	0.351	0.369	0.367	0.151	0.184	0.189	0.190	0.184	0.365	0.416	0.424	0.424	0.412
	u^{LLM}-Mixtral	0.168	0.205	0.210	0.207		0.280	0.336	0.341	0.351		0.160	0.191	0.194*	0.194*		0.374	0.422	0.428	0.426	
	u^{LLM}-LLama	0.191	0.221	0.227	0.226		0.317	0.362	0.382	0.377		0.156	0.190	0.193*	0.193*		0.375	0.425	0.432*	0.432*	
	u^{LLM}-GPT	0.193	0.223	0.226	**0.228**		0.339	0.381	0.382	**0.387**		0.166	0.199*	**0.206***	0.204*		0.385	0.442*	**0.455***	0.449*	

Table 6. Performance achieved by the symmetric LLM DIME for Dragon. In line with the PRF DIME, using the query encoder for both the query and the pseudo-relevant documents allows us to obtain the improvement also for Dragon.

		AP					nDCG@10					AP					nDCG@10				
	Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
		DL '19										DL '20									
Dragon	u^{LLM-S}-Gemma	0.466	0.515	0.528	0.532	0.517	0.708	0.737	0.749	0.754	0.740	0.488	0.521	0.524	0.525	0.506	0.705	0.735	0.739	0.741	0.718
	u^{LLM-S}-Mixtral	0.509	0.543	0.543	0.540		0.704	0.732	0.750	0.744		0.503	0.522	0.527*	0.525*		0.722	0.730	0.741	0.738	
	u^{LLM-S}-LLama	0.533	0.555*	**0.557***	0.553*		0.753	0.772*	0.772*	0.772*		0.502	0.525*	0.527*	0.524*		0.725	0.738	0.742	0.747	
	u^{LLM-S}-GPT	0.511	**0.557***	0.556*	0.552*		0.747	**0.775***	0.773*	0.771*		0.507	**0.534***	0.532*	0.532*		0.735	0.758*	0.755*	**0.759***	
		DL HD										RB '04									
Dragon	u^{LLM-S}-Gemma	0.245	0.262	0.271	0.272	0.262	0.379	0.394	0.400	0.397	0.384	0.200	0.224	0.230	0.231*	0.223	0.432	0.470	0.472	0.474	0.461
	u^{LLM-S}-Mixtral	0.242	0.266	0.267	0.262		0.355	0.378	0.390	0.384		0.206	0.229	0.233*	0.233*		0.441	0.468	0.475	0.475	
	u^{LLM-S}-LLama	0.279	**0.293***	0.290	0.284		0.404	**0.417***	0.416	0.411		0.205	0.228	0.233*	0.232*		0.449	0.472	0.476	0.474	
	u^{LLM-S}-GPT	0.267	0.282	0.280	0.276		0.402	0.416	0.410	0.403		0.216	0.242*	**0.245***	0.243*		0.449	**0.495***	0.492*	0.492*	

line with all the previous experiments, excluding Dragon, ANCE is the model that benefits the least from DIMEs. The optimal LLM depends on the target collection. For example, for the DL '20 and DL HD collections, the optimal strategy solution is based on using LLaMA to generate the pseudo-relevant document. Vice versa, for the RB '04 collection, GPT-4 seems to be more effective on average. In the case of the DL '20, the performance achieved by the two LLMs is

Table 7. LLM Assessor DIME (u^{LLMRJ}) and Symmetric LLM Assessor DIME ($u^{LLMRJ\text{-}S}$) performance. Overall, the results are closer to those of the PRF DIME than those of the LLM DIME, suggesting that either the LLM assessor is not sufficiently effective in recognizing highly relevant documents.

		AP					nDCG@10					AP					nDCG@10				
	Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
		DL '19										DL '20									
ANCE	u^{LLMRJ}	0.030	0.240	0.334	**0.370**	0.361	0.083	0.552	0.636	**0.656**	0.643	0.091	0.286	0.367	**0.392**	**0.392**	0.177	0.545	0.623	**0.651**	0.644
Contriever	u^{LLMRJ}	0.507	**0.512**	0.509	0.509	0.493	0.715	**0.726**	0.722	0.724	0.674	0.492	**0.508***	0.505*	0.500*	0.479	0.717*	**0.719***	0.711*	0.707*	0.672
Dragon	u^{LLMRJ}	0.410	0.460	0.478	0.497	0.517	0.700	0.715	0.723	0.737	0.740	0.428	0.465	0.478	0.489	0.506	0.681	0.700	0.699	0.708	0.718
	$u^{LLMRJ\text{-}S}$	0.505	0.527	0.533	**0.535**		0.731	0.732	0.743	**0.752**		0.496	0.518	**0.521**	0.518		0.723	0.729	**0.736**	0.734	
TAS-B	u^{LLMRJ}	0.492	**0.500**	0.497	0.492	0.476	0.715	**0.721**	0.714	0.715	0.717	0.481	0.499*	**0.502***	0.498*	0.475	0.716*	**0.721***	0.719*	0.719*	0.684
tct-ColBERT	u^{LLMRJ}	0.406	0.429*	**0.431***	0.424*	0.387	0.670	0.702	**0.706**	**0.706**	0.671	0.392	0.428	**0.438***	0.437*	0.398	0.663	0.689	**0.700***	0.698*	0.648
		DL HD										RB '04									
ANCE	u^{LLMRJ}	0.019	0.128	0.179	**0.185**	0.181	0.060	0.281	**0.344**	0.341	0.325	0.018	0.089	0.127	**0.133**	0.124	0.070	0.279	0.349	**0.356**	0.334
Contriever	u^{LLMRJ}	0.238	**0.259**	0.252	0.256	0.244	0.388	**0.408**	0.398	0.404	0.377	0.240	0.253*	0.256*	**0.257***	0.235	0.476	0.483	0.488*	**0.490***	0.465
Dragon	u^{LLMRJ}	0.199	0.227	0.240	0.246	0.262	0.358	0.366	0.376	0.387	0.384	0.169	0.184	0.192	0.204	0.223	0.404	0.417	0.424	0.439	0.461
	$u^{LLMRJ\text{-}S}$	0.250	0.262	0.265	**0.266**		0.386	0.388	**0.392**	0.390		0.199	0.223	0.229	**0.231**		0.440	0.462	0.469	**0.473**	
TAS-B	u^{LLMRJ}	0.231	0.245	**0.246**	0.241	0.236	0.383	0.397	**0.401**	0.393	0.376	0.221*	0.231*	**0.235***	0.234*	0.208	0.469*	0.475*	0.479*	**0.480***	0.447
tct-ColBERT	u^{LLMRJ}	0.201	0.218	**0.223**	**0.223**	0.208	0.330	0.349	0.362	0.361	**0.367**	0.187	0.213*	**0.217***	0.216*	0.184	0.428	0.461*	0.460*	**0.464***	0.412

very similar, without a clear winner. Overall, the two large models perform similarly, with most differences appearing only at the third decimal place, which are unlikely to reflect on the overall quality of the retrieval perceived by the user.

Dragon and the Symmetric LLM DIME. The LLM DIME, like the PRF DIME, is not effective for Dragon. Therefore, we test the symmetric LLM DIME that uses the query encoder to encode the pseudo-relevant document generated by the LLM. Table 6 reports the performance of such a DIME for Dragon. Akin to the PRF DIME, the symmetric encoding strategy renders the LLM DIME effective also in the case of Dragon. As for the other IR systems, we notice that this DIME always allows for improving the performance over the standard representation. In line with previous experiments, even though the 7B models are already effective, the best performances are achieved with LLaMA and GPT-4. Overall, the two LLMs tend to perform similarly. The highest absolute improvement is achieved on DL '20 if nDCG@10 is the evaluation measure. In this case, using GPT-4 as LLM to generate the pseudo-relevant and 80% of the dimensions, the improvement is +0.041 (+5.71%) of nDCG@10. The greatest relative improvement, on the other hand, is observed for AP on DL HD, where, with LLaMA and 40% of the dimensions, the improvement is +0.031 (+11.8%).

4.6 LLM Assessor DIME

We now describe the effectiveness of the LLM Assessor DIME that combines the positive effect of the PRF DIME, i.e., the usage of actual documents from the corpus, with those of the LLM DIME, i.e., a more expensive model to provide the feedback. Different from the LLM DIME, where the prompt was directly the query, in this case, we need a minimal prompt that allows us to condition the LLM distribution on the terms so that it outputs one among `not relevant`, `partially relevant`, `relevant`, `highly relevant`. The prompt used is the following:

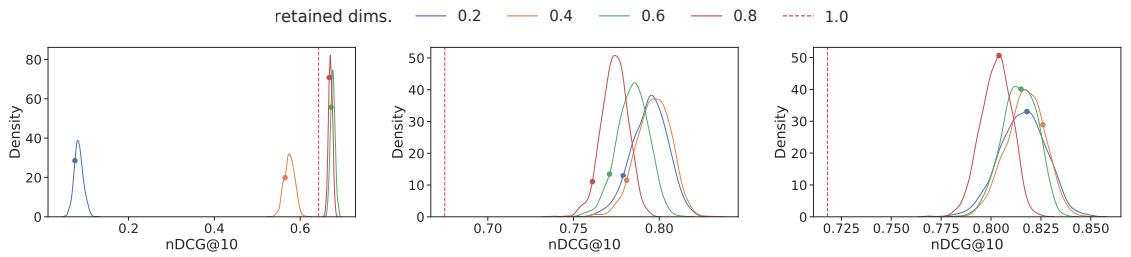
```
is document:
<document>
relevant to the query:
<query>
respond only with one among: ["non relevant", "partially relevant", "relevant", "highly relevant"]
```

As LLM, in this case, we experiment only with LLaMA: the experiments with the LLM DIME highlighted the sub-optimality of Gemma and Mixtral. At the same time, the similar performance between GPT-4 and LLaMA, as well as the fact that LLaMA is an open weights model — thus, it has stronger reproducibility — led us to choose it as the LLM relevance assessor. Table 7 reports the performance achieved by both the LLM Assessor DIME and the Symmetric LLM Assessor DIME (in the case of Dragon). Overall, we notice that the performances achieved by this DIME are comparable to, or worse than, those achieved using the PRF DIME, across almost all scenarios. One of the advantages of this approach is that it does not require setting a number of pseudo-relevant documents to be considered. Therefore, given their substantial similarity, this approach might be preferable where the number of pseudo-relevant documents considered might negatively impact the performance. Secondly, if we look across datasets, the dataset where the approach is the most effective is the RB '04: this approach might perform the best in out-of-domain scenarios, where the dense IR models are more likely to retrieve highly relevant documents in the first positions. Finally, this limited performance of the assessor DIME can be due to the overall low agreement between real and LLM-generated relevance judgements. If we consider the small pool of annotated documents through our sequential annotation process, we observe a Cohen's κ of 0.15 between real and LLM-generated labels for DL '19 (slight agreement), 0.24 for DL '20 (fair agreement), and 0.05 for DL HD, which indicates no agreement. We can expect that future generations of LLMs will behave as more effective annotators: in such a case, this DIME might become more and more relevant.

Table 8. Performance of the Active-Feedback DIME. Contriever, TAS-B, tct-ColBERT and Dragon (with symmetric encoding) show a significant improvement, regardless of the proportion of retained dimensions. ANCE improves when 60-80% dimensions are retained. Values marked with * are a statistically significant improvement over the baseline using all the dimensions (corresponding to Retained = 1).

		AP					nDCG@10					AP					nDCG@10				
	Retained	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1	0.2	0.4	0.6	0.8	1
		DL '19										DL '20									
ANCE	u^{rel}	0.034	0.271	0.369	**0.380**	0.361	0.084	0.584	**0.684**	**0.684**	0.643	0.080	0.295	0.377	**0.396**	0.392	0.155	0.568	0.644	**0.669**	0.644
Contriever	u^{rel}	0.554*	**0.561***	0.559*	0.553*	0.493	0.796*	**0.803***	0.785*	0.774*	0.674	0.507*	0.521*	**0.525***	0.520*	0.479	**0.789***	0.788*	0.786*	0.761*	0.672
Dragon	u^{rel}	0.418	0.465	0.483	0.500	0.517	0.708	0.714	0.732	**0.744**	0.740	0.440	0.470	0.479	0.494	0.506	0.705	0.715	0.711	0.719	0.718
	$u^{rel\text{-}s}$	0.551	**0.590***	0.589*	0.582*		0.821*	**0.834***	0.821*	0.822*		0.509	0.536*	**0.543***	0.542*		0.783*	0.787*	**0.788***	0.778*	
TAS-B	u^{rel}	0.563*	**0.571***	0.566*	0.557*	0.476	0.827*	**0.829***	0.820*	0.804*	0.717	0.482	0.503*	**0.511***	0.508*	0.475	0.763*	0.767*	**0.772***	0.761*	0.684
tct-ColBERT	u^{rel}	0.451*	**0.480***	0.470*	0.460*	0.387	0.770*	**0.788***	0.766*	0.759*	0.671	0.408	0.444*	**0.446***	**0.446***	0.398	0.734*	**0.775***	0.768*	0.769*	0.648
		DL HD										RB '04									
ANCE	u^{rel}	0.026	0.147	**0.202**	0.195	0.181	0.071	0.321	**0.390**	0.364	0.325	0.012	0.084	0.127	**0.130**	0.124	0.048	0.250	0.337	**0.345**	0.334
Contriever	u^{rel}	0.349*	**0.357***	0.349*	0.334*	0.244	0.579*	**0.581***	0.561*	0.529*	0.377	0.273*	0.290*	**0.298***	0.297*	0.235	0.592*	0.611*	**0.620***	0.611*	0.465
Dragon	u^{rel}	0.222	0.251	0.249	0.253	0.262	0.409	0.412	0.399	0.395	0.384	0.168	0.185	0.193	0.203	0.223	0.395	0.411	0.423	0.438	0.461
	$u^{rel\text{-}s}$	0.363*	**0.372***	**0.372***	0.353*		0.588*	**0.590***	0.575*	0.549*		0.191	0.226	0.236	**0.241***		0.456	0.499*	0.504*	**0.511***	
TAS-B	u^{rel}	**0.353***	**0.353***	0.345*	0.337*	0.236	**0.586***	**0.586***	0.570*	0.546*	0.376	0.238*	0.256*	**0.260***	0.256*	0.208	0.550*	0.569*	**0.570***	0.548*	0.447
tct-ColBERT	u^{rel}	0.291*	**0.317***	0.310*	0.300*	0.208	0.540*	**0.567***	0.551*	0.535*	0.367	0.216*	0.247*	**0.250***	0.246*	0.184	0.552*	**0.577***	0.570*	0.558*	0.412

4.6.1 Active Feedback DIME. In this section, we investigate whether we can exploit active feedback information to improve retrieval by following our DIME approach. To this end, we use the active-feedback DIME u^{rel}. A similar scenario may occur in systematic reviews, where documents are annotated iteratively. In particular, for each query, we assume that the user provides us with feedback on a single document that is highly relevant to the query. To simulate this feedback, we randomly select, for each query, the most relevant annotated document. Exploring the impact of partially relevant or non-relevant feedback is left for future work. For DL '19, DL '20, and DL HD, we randomly pick a document annotated with relevance "3"—the maximum—for queries having them, else "2". For RB '04 we sample among documents annotated with either "2" or "1", depending on the maximum relevance of the documents annotated for the query. Once we have a relevant document for each query, we instantiate u^{rel} and use the products of the weights



(a) ANCE (b) Contriever (c) TAS-B

Fig. 3. Distribution of performance on DL '19 for u^{act} when using different relevant documents. The dashed line represents the original performance (i.e., 100% dimensions retained). Contriever and TAS-B improves always, ANCE only if 60% dimensions are retained.The dot is the performance reported in Table 8.

in each dimension of the representations of the query and the relevant document to sort the dimensions in order of importance. Table 8 shows the performance achieved if, based on such active-feedback DIME, we retain a varying fraction of the representation dimensions. We also include in Table 8 the performance of the symmetric active-feedback DIME for Dragon. To simulate a real-life scenario, Table 8 reports the results when considering a single relevant document returned as feedback. First of all, it is interesting to notice that in all scenarios there is an improvement over the baseline. In particular, in the case of Contriever, TAS-B, tct-ColBERT, and Dragon (if the symmetric encoder is used), the improvement is significant (and large), regardless of the collection or evaluation measure considered. The maximum improvement is observed on the DL HD, where: Contriever achieves an impressive +0.204 (+55.6%) improvement in nDCG@10; TAS-B performance grows by +0.210 (+55.9%); Dragon increases of +0.206 (+53.6%) (using the symmetric approach); tct-ColBERT gets a +0.200 (+54.5%). ANCE, on the other hand, remains the most challenging model, with improvements that are not significant, although they are quite large in some cases (e.g., +0.065 of nDCG@10 with DL HD). Table 8 assumes a single relevant document as active feedback to obtain comparable results. Nevertheless, we can imagine that different users might click on different documents. We are thus interested in determining if, when using different documents as feedback, the user will observe widely different performances. To this end, we repeat the experiment mentioned above 1,000 times: for each query, we pick a random highly relevant document and use it to instantiate u^{rel}. In this setting, we carry out retrieval and measure the average performance over the test queries of DL '19. Figure 3 shows the results of this experiment for three systems (ANCE, Contriever, and TAS-B). Specifically, the plots report the distribution of nDCG@10 scores measured by randomly selecting 1,000 times the relevant document used to instantiate u^{rel} as a function of the fraction of dimensions retained. In line with Table 8 (but also with the oracle DIME used in Figure 2) for both Contriever and TAS-B (Figures 3b and 3c), all the fractions of retained dimensions allow improving the performance over the baseline (dashed red line). In the case of Contriever, we see that the settings using 0.4 or 0.2 of the dimensions obtain the best performance, while we achieve slightly lower nDCG@10 scores with 0.6 and 0.8, even if, also in these cases, we strongly outperform the baseline. Similarly, for TAS-B, 0.6, 0.4, and 0.2 achieve almost identical top performance, while 0.8 performs slightly lower, even if always better than the baseline. On average, the choice of the relevant document instantiating the DIME has a limited impact as the performances are distributed in an interval of ±0.025 around the mean. For ANCE (Fig. 3a), in line with previous analyses, the improvement is observed only when 60%/80% of the dimensions are retained. Even in this case, the improvement is observed independently of the relevant document considered.

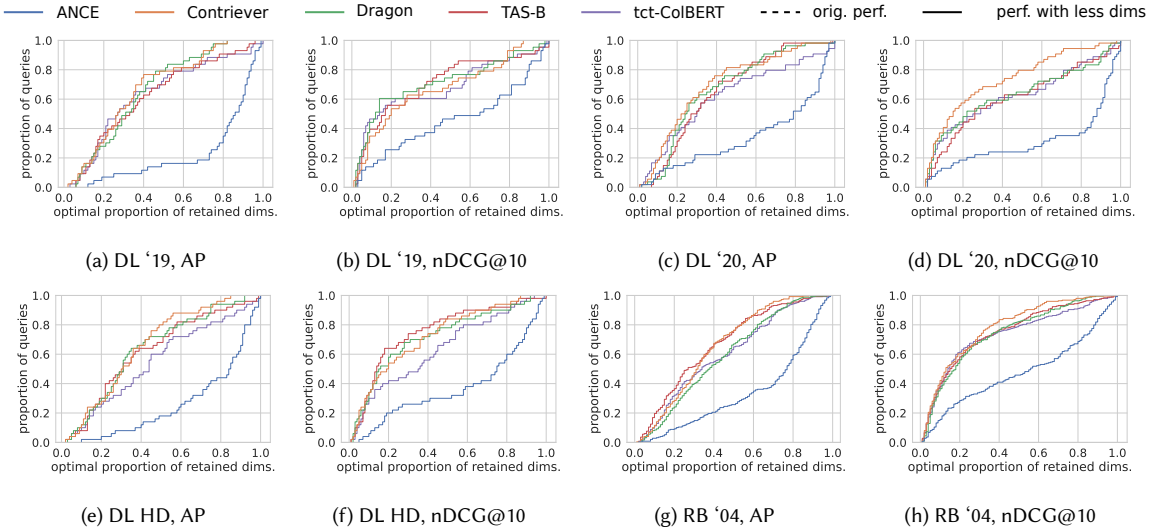
To conclude our analysis, it is worth noting that u^{mag}, u^{PRF}, and u^{LLM} are entirely automatic—therefore, they represent a full-fledged improvement over the current state of the art. On the contrary, the results of the u^{rel} DIME cannot be compared with purely automatic ranking strategies, as they require some active feedback from the user. Nevertheless, its application is simple as it requires a single relevant document—we can rely, for example, on a user's click. Thus, it can be used online to reduce the dimensions and retrieve more precise new documents or re-rank those already retrieved. Finally, it provides a clear view of the achievable improvements using proper DIME techniques.

4.7 Comparing DIMEs with the State of the Art

Table 9 reports the comparison of the most effective DIMEs proposed in this paper with VPRF [34]. In terms of DIMEs, we consider u^{PRF}@5, u^{CQvar}, u^{LLM}-GPT, u^{LLMRJ}, u^{rel}. Concerning Dragon, we use the symmetric encoder approaches to compute the importance, since they are more effective. For each DIME, we report the performance considering dimension cutoffs at 20%, 40%, 60%, and 80%. The performance of the models without any dimension pruning is reported in the column "`100% dims`". For VPRF, following Zhuang et al. [69], we set $\beta_1 = 0.4$ and $\beta_2 = 0.6$. The symbols "$*$" and "$\dagger$" represent a statistical improvement over the representation that uses all the dimensions and over VPRF. The patterns observable in Table 9 align with what was observed for the approaches in isolation. In almost all cases, we observe an optimal dimension pruning that allows us to overcome the representation based on all dimensions. Furthermore, such improvement is statistically significant for u^{CQvar}, u^{LLM}-GPT, and u^{rel}. On the contrary, u^{PRF}@5 and u^{LLMRJ} tend to be less effective, with improvements that are statistically significant only in a limited number of cases. As observed before, DIMEs tend to be more effective with precision-oriented measures (nDCG@10). If we compare the DIMEs with VPRF, we notice that the u^{PRF}@5 and u^{LLMRJ} are subpar. On the other hand, in most cases, other DIMEs overcome VPRF—all the exceptions occur when using AP as the evaluation measure. Furthermore, the improvement provided by the DIMEs over VPRF is significant in several cases, especially when considering u^{CQvar} and u^{rel}. In terms of significant improvement, VPRF improves over the baseline in a limited number of cases—almost exclusively when using AP. On the contrary, the improvement provided by the DIMEs is statistically significant in several more scenarios. Finally, as in previous scenarios, the ANCE model tends to benefit the least of DIMEs. When measuring the performance of ANCE using AP, the overall most effective approach is VPRF, although the difference is not statistically significant.

Table 9. Comparison of the proposed DIMEs with VPRF in terms of AP and nDCG@10. The column "100% dims" contains the performance of the full model without any pruning. In bold, the best result. The symbol *indicates significant improvements over the full representation, while †marks significant improvements over VPRF.

		100% dims	VPRF	u^{PRF}@5				u^{CQvar}				u^{LLM}-GPT				u^{LLMRJ}				u^{rel}			
				0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
DL '19																							
ANCE	AP	.361	**.410***	.034	.257	.344	.370	—	—	—	—	.031	.260	.351	.370	.030	.240	.334	.370	.034	.271	.369	.380
	nDCG	.643	.650	.087	.563	.635	.648	—	—	—	—	.081	.568	.651	.664	.083	.552	.636	.656	.084	.584	**.684**	.684
Contriever	AP	.493	.511	.491	.503	.511	.509	—	—	—	—	.516	.528*	.534*	.527*	.507	.512	.509	.509	.554*†	**.561***†	.559*†	.553*†
	nDCG	.674	.665	.647	.664	.679	.681	—	—	—	—	.720†	.742*†	.752*†	.750*†	.715†	.726†	.722†	.724†	.796*†	**.803***†	.785*†	.774*†
Dragon	AP	.517	.569*	.527	.545*	.548*	.547*	—	—	—	—	.511	.557*	.556*	.552*	.505	.527	.533	.535	.551	**.590***	.589*	.582*
	nDCG	.740	.751	.750	.749	.751	.758	—	—	—	—	.747*	.775*	.773*	.771*	.731	.732	.743	.752	.821*†	**.834***†	.821*†	.822*†
TAS-B	AP	.476	.524*	.499*	.505*	.507*	.505*	—	—	—	—	.511*	.529*	.527*	.521*	.492	.500	.497	.492	.563*†	**.571***†	.566*†	.557*†
	nDCG	.717	.726	.712	.725	.722	.724	—	—	—	—	.747	.749	.760*	.756*	.715	.721	.714	.715	.827*†	**.829***†	.820*†	.804*†
tct-ColBERT	AP	.387	.427*	.390	.411*	.412*	.406	—	—	—	—	.393	.421*	.424*	.419*	.406	.429*	.431*	.424*	.451*†	**.480***†	.470*†	.460*†
	nDCG	.671	.638	.634	.659	.662	.661	—	—	—	—	.669	.703†	.704*†	.694†	.670	.702†	.706†	.706†	.770*†	**.788***†	.766*†	.759*†
DL '20																							
ANCE	AP	.392	**.406**	.077	.289	.364	.391	—	—	—	—	.084	.284	.374	.397	.091	.286	.367	.392	.080	.295	.377	.396
	nDCG	.644	.649	.147	.542	.613	.644	—	—	—	—	.171	.537	.629	.655	.177	.545	.623	.651	.155	.568	.644	**.669**
Contriever	AP	.479	.492	.488	.494*	.496*	.494*	—	—	—	—	.503*	.512*†	.511*†	.504*	.492	.508*	.505*	.500*	.507*	.521*†	**.525***†	.520*†
	nDCG	.672	.693	.698*	.686	.690	.685	—	—	—	—	.719*	.722*	.725*†	.710*	.717*	.719*	.711*	.707*	**.789***†	.788*†	.786*†	.761*†
Dragon	AP	.506	.529*	.498	.523*	.520	.519	—	—	—	—	.507	.534*	.532*	.532*	.496	.518	.521	.518	.509	.536*	**.543***	.542*
	nDCG	.718	.720	.704	.721	.715	.718	—	—	—	—	.735	.758*†	.755*†	.759*†	.723	.729	.736	.734	.783*†	.787*†	**.788***†	.778*†
TAS-B	AP	.475	.485	.468	.479	.488	.488	—	—	—	—	.483	.498*	.501*	.500*	.481	.499*	.502*	.498*	.482	.503*	**.511***†	.508*†
	nDCG	.684	.694	.685	.685	.694	.697	—	—	—	—	.708	.706	.710	.712	.716*	.721*	.719*	.719*	.763*†	.767*†	**.772***†	.761*†
tct-ColBERT	AP	.398	.439*	.385	.423	.435*	.433*	—	—	—	—	.380	.423	.429*	.426*	.392	.428	.438*	.437*	.408	.444*	.446*	**.446***
	nDCG	.648	.678	.635	.676	.676	.681	—	—	—	—	.645	.683	.689	.685	.663	.689	.700*	.698*	.734*†	**.775***†	.768*†	.769*†
DL HD																							
ANCE	AP	.181	.190	.017	.126	.173	.184	—	—	—	—	.012	.129	.175	.186	.019	.128	.179	.185	.026	.147	**.202**	.195
	nDCG	.325	.334	.046	.275	.334	.332	—	—	—	—	.040	.284	.339	.348	.060	.281	.344	.341	.071	.321	**.390**†	.364
Contriever	AP	.244	.258	.248	.248	.253	.253	—	—	—	—	.259	.267	.270*	.270*	.238	.259	.252	.256	.349*†	**.357***†	.349*†	.334*†
	nDCG	.377	.403	.379	.378	.389	.387	—	—	—	—	.392	.409	.414*	.412*	.388	.408	.398	.404	.579*†	**.581***†	.561*†	.529*†
Dragon	AP	.262	.290	.275	.278	.277	.279	—	—	—	—	.267	.282	.280	.276	.250	.262	.265	.266	.363*†	**.372***†	.372*†	.353*†
	nDCG	.384	.409	.399	.397	.392	.398	—	—	—	—	.402	.416	.410	.403	.386	.388	.392	.390	.588*†	**.590***†	.575*†	.549*†
TAS-B	AP	.236	.259	.236	.243	.250	.255	—	—	—	—	.243	.254	.258	.250	.231	.245	.246	.241	.353*†	**.353***†	.345*†	.337*†
	nDCG	.376	.411	.369	.385	.391	.395	—	—	—	—	.385	.397	.401	.397	.383	.397	.401	.393	**.586***†	**.586***†	.570*†	.546*†
tct-ColBERT	AP	.208	.215	.205	.215	.221	.219	—	—	—	—	.193	.223	.226	.228	.201	.218	.223	.223	.291*†	**.317***†	.310*†	.300*†
	nDCG	.367	.354	.339	.360	.361	.360	—	—	—	—	.339	.381	.382	.387	.330	.349	.362	.361	.540*†	**.567***†	.551*†	.535*†
RB '04																							
ANCE	AP	.124	**.146***	.016	.081	.119	.125	.021	.096	.137	.136	.017	.086	.126	.130	.018	.089	.127	.133	.012	.084	.127	.130
	nDCG	.334	.348	.059	.247	.321	.333	.077	.289	**.364**	.362	.067	.267	.343	.347	.070	.279	.349	.356	.048	.250	.337	.345
Contriever	AP	.235	.256*	.249*	.258*	.259*	.257*	.264*	.267*†	.261*	.249*	.262*	.270*†	.270*†	.267*†	.240	.253*	.256*	.257*	.273*†	.290*†	**.298***†	.297*†
	nDCG	.465	.476	.476	.479*	.479*	.475	.531*†	.531*†	.509*†	.489*	.528*†	.534*†	.531*†	.525*†	.476	.483	.488*	.490*	.592*†	.611*†	**.620***†	.611*†
Dragon	AP	.223	.235*	.209	.228	.232*	.233*	.235*	**.255***†	.250*†	.238*	.216	.242*	.245*†	.243*	.199	.223	.229	.231	.191	.226	.236	.241*
	nDCG	.461	.449	.437	.449	.451	.454	.506*†	**.521***†	.502*†	.483*†	.449	.495*†	.492*†	.492*†	.440	.462	.469†	.473†	.456	.499*†	.504*†	.511*†
TAS-B	AP	.208	.234*	.224*	.231*	.231*	.231*	.220*	.236*	.232*	.221*	.224*	.239*	.238*	.236*	.221*	.231*	.235*	.234*	.238*	.256*†	**.260***†	.256*†
	nDCG	.447	.473*	.463*	.467*	.468*	.469*	.481*	.496*†	.482*	.469*	.476*	.501*†	.495*†	.489*	.469*	.475*	.479*	.480*	.550*†	.569*†	**.570***†	.548*†
tct-ColBERT	AP	.184	.211*	.187	.206*	.208*	.205*	.183	.215*	.211*	.198*	.166	.199*	.206*	.204*	.187	.213*	.217*	.216*	.216*	.247*†	**.250***†	.246*†
	nDCG	.412	.435*	.412	.431*	.436*	.434*	.430	.473*†	.465*†	.442*	.385	.442*	.455*†	.449*	.428	.461*†	.460*†	.464*†	.552*†	**.577***†	.570*†	.558*†



(a) DL '19, AP (b) DL '19, nDCG@10 (c) DL '20, AP (d) DL '20, nDCG@10

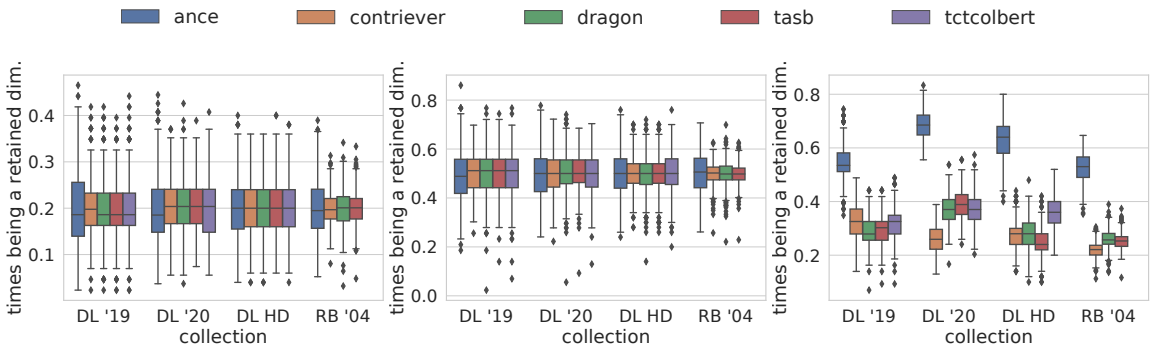
(e) DL HD, AP (f) DL HD, nDCG@10 (g) RB '04, AP (h) RB '04, nDCG@10

Fig. 4. Distribution of the number of optimal dimensions. The number of optimal dimensions is typically bigger (i.e., lower curves) for ANCE and for AP.

4.8 An Analysis on the Optimal Dimensions

We propose here a more detailed analysis of the dimensions that induce the optimal performance. The objective of this analysis is twofold: i) determining how many dimensions are optimal and if there is a pattern across queries in this regard; and ii) understanding if the optimal dimensions are common across different queries. The next experiments are based on the Oracle DIME, which allows us to better investigate the role of the different dimensions in improving performance. Since we want a more fine-grained analysis, we consider the number of retained dimensions from 1% (approximately 8 dimensions) to 100%, with a step of 1%.

4.8.1 How many dimensions are optimal. Figure 4 reports the optimal number of retained dimensions depending on the retrieval model, collection, and measure considered. For example, in Figure 4a we can observe that for approximately 40% of the queries (y-axis), the optimal number of dimensions is less than 20% (x-axis). The first pattern that we can observe concerns ANCE having a systematically lower distribution compared to the other IR models. This indicates that, to achieve optimal performance, more dimensions need to be retained on average. In most cases, approximately 50% of the queries (y-axis, upper part) need more than 60-80% of the dimensions (x-axis, right) to obtain the optimal performance. This follows our previous observations concerning this specific model needing more retained dimensions compared to the others to improve compared to the full representations. The other IR models present a more uniform behaviour: they have very similar curves. In general, we see that, to optimize approximately 60% of the queries (y-axis, lower part), less than 20-40% dimensions are sufficient. Notice that the opposite holds: if we add dimensions beyond the optimal point, performance deteriorates. Concerning the IR measures, we notice that the curves tend to be slightly shifted towards the left for nDCG@10 than AP (i.e., fewer dimensions are needed to optimize nDCG@10 than AP). This pattern is particularly evident for RB '04 and visible in DL '19 and DL HD. Using a representation with fewer dimensions, we are likely to retrieve documents concerning similar topics, as the representation is "less expressive". Being nDCG@10, a precision-oriented measure, it is sufficient to find 10—possibly similar—relevant documents to



(a) 20% dimensions retained (b) 50% dimensions retained (c) optimal number of dimensions retained

Fig. 5. Distribution of the times each dimension is retained over the queries.

optimize it. Vice versa, AP is more recall-oriented: therefore, to optimize it, it is necessary to retrieve more relevant documents, possibly concerning very different topics. Thus, slightly more expressive representations are needed. The general absence of a prolonged plateau indicates that the number of optimal dimensions is a query-dependent property. Being able to find the optimal cutoff in a query-wise manner would allow us to further improve the performance beyond what can already be achieved using DIMEs. We leave the study of determining such a cutoff as future work.

4.8.2 Which dimensions are optimal. The second question we are interested in answering concerns whether some dimensions are never among the set of optimal dimensions. This would be extremely beneficial from the efficiency performance perspective: if such dimensions existed and we could identify them, they could be removed from the representation. This would reduce the number of operations and the disk occupation. To verify this, Figure 5 describes how many times each dimension is retained, for all the queries. We report the plots assuming to retain the 20% best dimensions (Figure 5a), the 50% (Figure 5b), and the optimal number of dimensions, as described in the previous Section (Figure 5c). Consider, for example, Figure 5a: the first blue bar describes what happens with the ANCE retrieval system used on DL '19, if we were to retain only the top 20% of the dimensions. We observe that the dimension that is retained the least amount of times, is retained 2.3% of the times (1 query), on average, each dimension is retained 18.6% of the times (8 queries), while the dimension retained most times is retained 86% (retained for 37 queries). Therefore, if we retained 20% of the dimensions, each dimension would be chosen at least once. The same pattern also holds for all the other retrieval systems and collections. The RB '04, which contains approximately 5 times the queries in the other collections, exhibits narrower distributions, suggesting that, if we had more queries, we would observe that optimal dimensions are distributed uniformly across queries. Notice that, on average, each dimension is retained 20% of the time, further evidence of the underlying uniform distribution.

Retaining only 20% of the dimensions might be suboptimal: Figure 2 shows that often 20% dimensions do not provide the best results, and a similar pattern is observed also in Figure 4. If we move further and consider retaining 50% of the queries, we observe that, if we exclude a few outlier dimensions, each dimension is in the set of optimal queries for 20-40% of the queries. This highlights that no dimension can be removed without damaging 20-40% of the queries. Again, the expectation is close to 0.5, and it reduces if we consider more queries (as for RB '04)—suggesting the underlying uniform distribution of the dimensions.

As a final analysis, we consider the optimal number of dimensions for each query (based on nDCG@10), reported in Figure 5c. The dimension chosen for the least queries is chosen 7% of the time (3 queries). Excluding outliers, each dimension is considered optimal for at least 10 to 20% of the queries. The width of the distribution on the RB '04 is further reduced. Notice that the distribution for ANCE is shifted toward the top of the plot. This is in line with what is observed in Figure 4: ANCE is optimized when many dimensions are considered, and therefore each dimension is likely to be picked more times.

This empirically highlights that no dimension can be preemptively removed across all queries: each dimension is important for at least 10% of the queries. Similarly, no dimension is important to all queries, with the most important dimensions being such for approximately 50% of the queries, depending on the model.

4.9 On DIMEs Efficiency

Like other PRF or query rewriting and processing approaches, DIMEs introduces some computational overhead due to the computation of the dimensions to be preserved. The extent of this overhead depends on the available information and the specific type of DIME used. For instance, the LLM DIME requires generating a pseudo-relevant document in response to the query, which can be particularly challenging in real-time scenarios. However, LLM generation will likely become faster over time, and several commercial search engines—such as Bing[7] and Google[8]—already use generated content in their SERPs. Similarly, the Rel DIME relies on access to a relevant document in response to the query. Once such a document is retrieved, the DIME derived from it can be cached and reused for similar queries. It is worth noting that, once a representation of the relevant or pseudo-relevant information (whether a document or a set of query variations) is available, the computation of the optimal dimensions is very efficient. This step typically involves only linear operations, such as Hadamard products and vector summations.

Finally, we emphasise that DIMEs do not introduce significant efficiency overheads within vector search pipelines. Once the relevant dimensions are identified, DIMEs can be seamlessly integrated into current indexing frameworks—such as FAISS, Product Quantization (PQ), or Hierarchical Navigable Small World (HNSW)—by simply zeroing out the "unimportant" dimensions in the query representation, leaving the rest of the retrieval process unaffected.

Two key research directions could further enhance the efficiency of DIMEs: (i) The identification of query-independent DIMEs: if it becomes feasible to determine subsets of unimportant dimensions for clusters of documents, indexes could be built that exclude these dimensions altogether, thus reducing both storage and computational costs during inference. (ii) The design of new indexing pipelines: These would involve algorithmic optimisations to bypass operations (e.g., sums and products) involving unimportant dimensions. Both directions offer promising challenges for improving the efficiency of DIMEs, and their investigation is open for future research.

5 CONCLUSION AND FUTURE WORK

This paper introduces the MC hypothesis for the latent space learned by dense IR neural models: "high-dimensional representations of queries and documents relevant to them lie in a query-dependent lower-dimensional manifold of the representation space". According to this hypothesis, for a given query, there exists a subspace of the learned representation space in which the representations of relevant documents cluster closely around the query. To empirically validate our hypothesis, we restrict our investigation to linear subspaces—i.e., subspaces formed by zeroing out certain dimensions of the original representation space—under the assumption of dimension independence. To address this

[7] https://www.microsoft.com/en-us/edge/features/the-new-bing
[8] https://developers.google.com/search/docs/appearance/ai-overviews

task, we introduce the problem of Dimension Importance Estimation. Given a dense IR model and a query, the goal is to determine which dimensions in the query and document representations are most important for producing an optimal ranking. In doing so, we also introduce a novel class of *Dimension IMportance Estimators (DIMEs)*. We propose an oracle DIME that demonstrates how selecting optimal dimensions can improve retrieval performance by up to 184% (from 0.123 to 0.351 in nDCG@10). While this oracle supports the validity of the MC hypothesis, it requires access to relevant documents and is thus impractical for real-world use. While the oracle DIME effectively highlights that the MC hypothesis has ground in reality, it relies on the availability of relevant documents and cannot be used in practice. To enable practical application, we propose several DIMEs based on different heuristics: the magnitude of query representation dimensions, pseudo-relevant feedback documents, and documents generated by an LLM. These heuristics yield strong results, with improvements of up to +11.6%—increasing nDCG@10 from 0.674 to 0.752. Finally, we introduce an active-feedback DIME that uses a single relevant document to significantly boost retrieval performance. This approach achieves improvements of up to +49.6% in AP (from 0.236 to 0.353) and +55.9% in nDCG@10 (from 0.376 to 0.586). A key advantage of DIME models is their compatibility with existing dense IR pipelines, whether for ranking or re-ranking. Using a DIME model to identify the most relevant dimensions per query can lead to substantial performance gains.

In future work, we aim to automate the selection of the optimal number of dimensions to retain. We also intend to explore DIMEs informed by additional signals—such as prior utterances in conversational search or query reformulations—and to develop models that leverage linear combinations of dimensions.

ACKNOWLEDGMENTS

This work was supported, in part, by the Spoke "FutureHPC & BigData" of the ICSC – Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing, the Spoke "Human-centered AI" of the M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research", the "Extreme Food Risk Analytics" (EFRA) project, Grant no. 101093026, funded by European Union – NextGenerationEU, the FoReLab project (Departments of Excellence), the NEREO PRIN project funded by the Italian Ministry of Education and Research Grant no. 2022AEFHAZ and the CAMEO PRIN 2022 Project Grant no. 2022ZLL7MW.

REFERENCES

[1] Giambattista Amati. 2003. *Probability models for information retrieval based on divergence from randomness.* Ph. D. Dissertation. University of Glasgow, UK. http://theses.gla.ac.uk/1570/

[2] Gianni Amati and C. J. van Rijsbergen. 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.* 20, 4 (2002), 357–389. https://doi.org/10.1145/582415.582416

[3] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel (Eds.). ACM, 725–728. https://doi.org/10.1145/2911451.2914671

[4] Peter Bailey, Alistair Moffat, Falk Scholer, and Paul Thomas. 2017. Retrieval Consistency in the Presence of Query Variations. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, Noriko Kando, Tetsuya Sakai, Hideo Joho, Hang Li, Arjen P. de Vries, and Ryen W. White (Eds.). ACM, 395–404. https://doi.org/10.1145/3077136.3080839

[5] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (2013), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[6] Rodger Benham, Joel M. Mackenzie, Alistair Moffat, and J. Shane Culpepper. 2019. Boosting Search Performance Using Query Variations. *ACM Trans. Inf. Syst.* 37, 4 (2019), 41:1–41:25. https://doi.org/10.1145/3345001

[7] Trevor J. Bihl, Kenneth W. Bauer Jr., and Michael A. Temple. 2016. Feature Selection for RF Fingerprinting With Multiple Discriminant Analysis and Using ZigBee Device Emissions. *IEEE Trans. Inf. Forensics Secur.* 11, 8 (2016), 1862–1874. https://doi.org/10.1109/TIFS.2016.2561902

[8] Bodo Billerbeck, Falk Scholer, Hugh E. Williams, and Justin Zobel. 2003. Query expansion using associated queries. In *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, 2003*. ACM, 2–9. https://doi.org/10.1145/956863.956866

[9] Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the Contextual Embedding Space: Clusters and Manifolds. In *International Conference on Learning Representations*. https://openreview.net/forum?id=xYGNO86OWDH

[10] Emily Cheng, Corentin Kervadec, and Marco Baroni. 2023. Bridging Information-Theoretic and Geometric Compression in Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 12397–12420. https://doi.org/10.18653/v1/2023.emnlp-main.762

[11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. *CoRR* abs/2102.07662 (2021). arXiv:2102.07662 https://arxiv.org/abs/2102.07662

[12] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *CoRR* abs/2003.07820 (2020). arXiv:2003.07820 https://arxiv.org/abs/2003.07820

[13] J. Shane Culpepper, Guglielmo Faggioli, Nicola Ferro, and Oren Kurland. 2022. Topic Difficulty: Collection and Query Formulation Effects. *ACM Trans. Inf. Syst.* 40, 1 (2022), 19:1–19:36. https://doi.org/10.1145/3470563

[14] Maurizio Ferrari Dacrema, Fabio Moroni, Riccardo Nembrini, Nicola Ferro, Guglielmo Faggioli, and Paolo Cremonesi. 2022. Towards Feature Selection for Ranking and Classification Exploiting Quantum Annealers. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 2814–2824. https://doi.org/10.1145/3477495.3531755

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. ACL, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[16] R Dhanya, Irene Rose Paul, Sai Sindhu Akula, Madhumathi Sivakumar, and Jyothisha J Nair. 2020. F-test feature selection in Stacking ensemble model for breast cancer prediction. *Procedia Computer Science* 171 (2020), 1561–1570. https://doi.org/10.1016/j.procs.2020.04.167 Third International Conference on Computing and Network Communications (CoCoNet'19).

[17] Giorgio Maria Di Nunzio and Guglielmo Faggioli. 2021. A Study of a Gain Based Approach for Query Aspects in Recall Oriented Tasks. *Applied Sciences* 11, 19 (2021). https://www.mdpi.com/2076-3417/11/19/9075

[18] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query Expansion with Locally-Trained Word Embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics. https://doi.org/10.18653/V1/P16-1035

[19] Guglielmo Faggioli, Laura Dietz, Charles Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelo Kanoulas, Martin Potthast, Benno Stein, and Henning Wachsmuth. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Advances in Information Retrieval Theory, 9th International Conference on the Theory of Information Retrieval, ICTIR 2023, Taipei, Taiwan, July 23, 2023*. ACM. https://doi.org/10.1145/3578337.3605136

[20] Guglielmo Faggioli, Nicola Ferro, Raffaele Perego, and Nicola Tonellotto. 2024. Dimension Importance Estimation for Dense Information Retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024*. ACM, Washington D.C., USA.

[21] Guglielmo Faggioli, Oleg Zendel, J. Shane Culpepper, Nicola Ferro, and Falk Scholer. 2021. An Enhanced Evaluation Framework for Query Performance Prediction. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12656)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 115–129. https://doi.org/10.1007/978-3-030-72113-8_8

[22] Guglielmo Faggioli, Oleg Zendel, J. Shane Culpepper, Nicola Ferro, and Falk Scholer. 2022. sMARE: a new paradigm to evaluate and understand query performance prediction methods. *Inf. Retr. J.* 25, 2 (2022), 94–122. https://doi.org/10.1007/S10791-022-09407-W

[23] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse Lexical and Expansion Model for First Stage Ranking. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2288–2292. https://doi.org/10.1145/3404835.3463098

[24] Andrea Gigli, Claudio Lucchese, Franco Maria Nardini, and Raffaele Perego. 2016. Fast Feature Selection for Learning to Rank. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval* (Newark, Delaware, USA) *(ICTIR '16)*. Association for Computing Machinery, New York, NY, USA, 167–170. https://doi.org/10.1145/2970398.2970433

[25] Andrea Gigli, Claudio Lucchese, Franco Maria Nardini, and Raffaele Perego. 2016. Fast Feature Selection for Learning to Rank. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, Ben Carterette, Hui Fang, Mounia Lalmas, and Jian-Yun Nie (Eds.). ACM, 167–170. https://doi.org/10.1145/2970398.2970433

[26] Evan Hernandez and Jacob Andreas. 2021. The Low-Dimensional Linear Geometry of Contextualized Word Representations. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, Arianna Bisazza and Omri Abend (Eds.). Association for Computational Linguistics, Online, 82–93. https://doi.org/10.18653/v1/2021.conll-1.7

[27] Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently Teaching an Effective Dense Retriever with Balanced Topic Aware Sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 113–122. https://doi.org/10.1145/3404835.3462891

[28] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards Unsupervised Dense Information Retrieval with Contrastive Learning. *CoRR* abs/2112.09118 (2021). arXiv:2112.09118 https://arxiv.org/abs/2112.09118

[29] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data* 7, 3 (2021), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

[30] Alan Jovic, Karla Brkic, and Nikola Bogunovic. 2015. A review of feature selection methods with applications. In *38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015, Opatija, Croatia, May 25-29, 2015*, Petar Biljanovic, Zeljko Butkovic, Karolj Skala, Branko Mikac, Marina Cicin-Sain, Vlado Sruk, Slobodan Ribaric, Stjepan Gros, Boris Vrdoljak, Mladen Mauher, and Andrej Sokolic (Eds.). IEEE, 1200–1205. https://doi.org/10.1109/MIPRO.2015.7160458

[31] Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided Supervision for OpenQA with ColBERT. *Trans. Assoc. Comput. Linguistics* 9 (2021), 929–944. https://doi.org/10.1162/TACL_A_00405

[32] Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham M. Kakade, Prateek Jain, and Ali Farhadi. 2022. Matryoshka Representation Learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (Eds.). http://papers.nips.cc/paper_files/paper/2022/hash/c32319f4868da7613d78af9993100e42-Abstract-Conference.html

[33] Saar Kuzi, Anna Shtok, and Oren Kurland. 2016. Query Expansion Using Word Embeddings. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, Snehasis Mukhopadhyay, ChengXiang Zhai, Elisa Bertino, Fabio Crestani, Javed Mostafa, Jie Tang, Luo Si, Xiaofang Zhou, Yi Chang, Yunyao Li, and Parikshit Sondhi (Eds.). ACM, 1929–1932. https://doi.org/10.1145/2983323.2983876

[34] Hang Li, Ahmed Mourad, Shengyao Zhuang, Bevan Koopman, and Guido Zuccon. 2023. Pseudo Relevance Feedback with Deep Language Models and Dense Retrievers: Successes and Pitfalls. *ACM Trans. Inf. Syst.* 41, 3 (2023), 62:1–62:40. https://doi.org/10.1145/3570724

[35] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. 2018. Feature Selection: A Data Perspective. *ACM Comput. Surv.* 50, 6 (2018), 94:1–94:45. https://doi.org/10.1145/3136625

[36] Xianming Li, Zongxi Li, Jing Li, Haoran Xie, and Qing Li. 2024. 2D Matryoshka Sentence Embeddings. *CoRR* abs/2402.14776 (2024). https://doi.org/10.48550/ARXIV.2402.14776 arXiv:2402.14776

[37] Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to Train Your Dragon: Diverse Augmentation Towards Generalizable Dense Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 6385–6400. https://doi.org/10.18653/V1/2023.FINDINGS-EMNLP.423

[38] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2020. Distilling Dense Representations for Ranking using Tightly-Coupled Teachers. *CoRR* abs/2010.11386 (2020). arXiv:2010.11386 https://arxiv.org/abs/2010.11386

[39] Xiaolu Lu, Oren Kurland, J. Shane Culpepper, Nick Craswell, and Ofri Rom. 2019. Relevance Modeling with Multiple Query Variations. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2019, Santa Clara, CA, USA, October 2-5, 2019*, Yi Fang, Yi Zhang, James Allan, Krisztian Balog, Ben Carterette, and Jiafeng Guo (Eds.). ACM, 27–34. https://doi.org/10.1145/3341981.3344224

[40] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, Dense, and Attentional Representations for Text Retrieval. *Trans. Assoc. Comput. Linguistics* 9 (2021), 329–345. https://doi.org/10.1162/tacl_a_00369

[41] Sean MacAvaney and Luca Soldaini. 2023. One-Shot Labeling for Automatic Relevance Estimation. *CoRR* abs/2302.11266 (2023). https://doi.org/10.48550/arXiv.2302.11266 arXiv:2302.11266

[42] Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. 2023. Generative Relevance Feedback with Large Language Models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, Hsin-Hsi Chen, Wei-Jou (Edward) Duh, Hen-Hsen Huang, Makoto P. Kato, Josiane Mothe, and Barbara Poblete (Eds.). ACM, 2026–2031. https://doi.org/10.1145/3539618.3591992

[43] Iain Mackie, Jeffrey Dalton, and Andrew Yates. 2021. How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 2335–2341. https://doi.org/10.1145/3404835.3463262

[44] Jonathan Mamou, Hang Le, Miguel A Del Rio, Cory Stephenson, Hanlin Tang, Yoon Kim, and SueYeon Chung. 2020. Emergence of Separable Manifolds in Deep Language Representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 623, 11 pages.

[45] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016 (CEUR Workshop Proceedings, Vol. 1773)*, Tarek Richard Besold, Antoine Bordes, Artur S. d'Avila Garcez, and Greg Wayne (Eds.). CEUR-WS.org. https:

//ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf

[46] Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). arXiv:1901.04085 http://arxiv.org/abs/1901.04085

[47] OpenAI. 2023. ChatGPT [Large language model]; Accessed on December 2023.

[48] Hanchuan Peng, Fuhui Long, and Chris H. Q. Ding. 2005. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 8 (2005), 1226–1238. https://doi.org/10.1109/TPAMI.2005.159

[49] Gustavo Penha, Arthur Câmara, and Claudia Hauff. 2022. Evaluating the Robustness of Retrieval Pipelines with Query Variation Generators. In *Advances in Information Retrieval - 44th European Conference on IR Research, ECIR 2022, Stavanger, Norway, April 10-14, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13185)*, Matthias Hagen, Suzan Verberne, Craig Macdonald, Christin Seifert, Krisztian Balog, Kjetil Nørvåg, and Vinay Setty (Eds.). Springer, 397–412. https://doi.org/10.1007/978-3-030-99736-6_27

[50] Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. 2021. The Intrinsic Dimension of Images and Its Impact on Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=XJk19XzGq2J

[51] Alberto Purpura, Karolina Buchner, Gianmaria Silvello, and Gian Antonio Susto. 2021. Neural Feature Selection for Learning to Rank. In *Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12657)*, Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani (Eds.). Springer, 342–349. https://doi.org/10.1007/978-3-030-72240-1_34

[52] Ashwini Rahangdale and Shital A. Raut. 2019. Deep Neural Network Regularization for Feature Selection in Learning-to-Rank. *IEEE Access* 7 (2019), 53988–54006. https://doi.org/10.1109/ACCESS.2019.2902640

[53] Joseph John Rocchio Jr. 1971. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing* (1971).

[54] Irene Rodríguez-Luján, Ramón Huerta, Charles Elkan, and Carlos Santa Cruz. 2010. Quadratic Programming Feature Selection. *J. Mach. Learn. Res.* 11 (2010), 1491–1516. https://doi.org/10.5555/1756006.1859900

[55] Dwaipayan Roy, Debasis Ganguly, Sumit Bhatia, Srikanta Bedathur, and Mandar Mitra. 2018. Using Word Embeddings for Information Retrieval: How Collection and Term Normalization Choices Affect Performance. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 1835–1838. https://doi.org/10.1145/3269206.3269277

[56] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. 2016. Using Word Embeddings for Automatic Query Expansion. *CoRR* abs/1606.07608 (2016). arXiv:1606.07608 http://arxiv.org/abs/1606.07608

[57] Andrew Rutherford. 2011. *ANOVA and ANCOVA: a GLM approach.* John Wiley & Sons.

[58] Noelia Sánchez-Maroño, María Caamaño-Fernández, Enrique F. Castillo, and Amparo Alonso-Betanzos. 2006. Functional Networks and Analysis of Variance for Feature Selection. In *Intelligent Data Engineering and Automated Learning - IDEAL 2006, 7th International Conference, Burgos, Spain, September 20-23, 2006, Proceedings (Lecture Notes in Computer Science, Vol. 4224)*, Emilio Corchado, Hujun Yin, Vicente J. Botti, and Colin Fyfe (Eds.). Springer, 1031–1038. https://doi.org/10.1007/11875581_123

[59] Paul Thomas, Seth Spielman, Nick Craswell, and Bhaskar Mitra. 2024. Large Language Models can Accurately Predict Searcher Preferences. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 1930–1940. https://doi.org/10.1145/3626772.3657707

[60] Kari Torkkola. 2003. Feature Extraction by Non-Parametric Mutual Information Maximization. *J. Mach. Learn. Res.* 3 (2003), 1415–1438. http://jmlr.org/papers/v3/torkkola03a.html

[61] John W. Tukey. 1949. Comparing Individual Means in the Analysis of Variance. *Biometrics* 5, 2 (1949), 99–114. http://www.jstor.org/stable/3001913

[62] C. J. van Rijsbergen. 1979. *Information Retrieval.* Butterworth.

[63] Ellen Voorhees. 2005. Overview of the TREC 2004 Robust Retrieval Track. https://doi.org/10.6028/NIST.SP.500-261

[64] 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 *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=zeFrfgyZln

[65] Hamed Zamani and W. Bruce Croft. 2016. Embedding-based Query Language Models. In *Proceedings of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016, Newark, DE, USA, September 12- 6, 2016*, Ben Carterette, Hui Fang, Mounia Lalmas, and Jian-Yun Nie (Eds.). ACM, 147–156. https://doi.org/10.1145/2970398.2970405

[66] Oleg Zendel, J. Shane Culpepper, and Falk Scholer. 2021. Is Query Performance Prediction With Multiple Query Variations Harder Than Topic Performance Prediction?. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 1713–1717. https://doi.org/10.1145/3404835.3463039

[67] Zilin Zeng, Hongjun Zhang, Rui Zhang, and Chengxiang Yin. 2015. A novel feature selection method considering feature interaction. *Pattern Recognit.* 48, 8 (2015), 2656–2666. https://doi.org/10.1016/j.patcog.2015.02.025

[68] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing Dense Retrieval Model Training with Hard Negatives. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 1503–1512. https://doi.org/10.1145/3404835.3462880

[69] Shengyao Zhuang, Hang Li, and Guido Zuccon. 2022. Implicit Feedback for Dense Passage Retrieval: A Counterfactual Approach. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai (Eds.). ACM, 18–28. https://doi.org/10.1145/3477495.3531994