

# Are We There Yet? A Decision Framework for Replacing Term-Based Retrieval with Dense Retrieval Systems

SEBASTIAN HOFSTÄTTER, TU Wien

NICK CRASWELL, Microsoft

BHASKAR MITRA, Microsoft

HAMED ZAMANI, University of Massachusetts Amherst

ALLAN HANBURY, TU Wien

Recently, several dense retrieval (DR) models have demonstrated competitive performance to term-based retrieval that are ubiquitous in search systems. In contrast to term-based matching, DR projects queries and documents into a dense vector space and retrieves results via (approximate) nearest neighbor search. Deploying a new system, such as DR, inevitably involves tradeoffs in aspects of its performance. Established retrieval systems running at scale are usually well understood in terms of effectiveness and costs, such as query latency, indexing throughput, or storage requirements. In this work, we propose a framework with a set of criteria that go beyond simple effectiveness measures to thoroughly compare two retrieval systems with the explicit goal of assessing the readiness of one system to replace the other. This includes careful tradeoff considerations between effectiveness and various cost factors. Furthermore, we describe *guardrail* criteria, since even a system that is better on average may have systematic failures on a minority of queries. The guardrails check for failures on certain query characteristics and novel failure types that are only possible in dense retrieval systems.

We demonstrate our decision framework on a Web ranking scenario. In that scenario, state-of-the-art DR models have surprisingly strong results, not only on average performance but passing an extensive set of guardrail tests, showing robustness on different query characteristics, lexical matching, generalization, and number of regressions. DR with approximate nearest neighbor search has comparable low query latency to term-based systems. The main reason to reject current DR models in this scenario is the cost of vectorization, which is much higher than the cost of building a traditional index. It is impossible to predict whether DR will become ubiquitous in the future, but one way this is possible is through repeated applications of decision processes such as the one presented here.

Additional Key Words and Phrases: Information Retrieval, Machine Learning, Neural Ranking

## 1 INTRODUCTION

Term-based indexes comprise of a list of terms and their locations in a text collection. The idea of term-based indexing predates modern computers and their application can be traced back to the Bible concordances of the 13<sup>th</sup> century [21] which were verbal indexes to the Bible. Subsequently, the oldest printed indexes appeared in the mid-15<sup>th</sup> century [65]. Today, term-based indexes in the form of inverted-indexes [70] are employed in most computational search systems, including commercial web search engines.

Boytssov et al. [5] have argued that another approach involving nearest neighbour lookup should be revisited in the context of search. In light of emerging neural representation learning models for retrieval [49], such an approach combined with learned dense vector representations—sometimes referred to as dense retrieval (DR)—have renewed interests in the research community. Initial attempts at dense retrieval—both in the pre-deep learning era [25] and in

---

Authors' addresses: Sebastian Hofstätter, TU Wien, s.hofstatter@tuwien.ac.at; Nick Craswell, Microsoft, nickcr@microsoft.com; Bhaskar Mitra, Microsoft, bmitra@microsoft.com; Hamed Zamani, University of Massachusetts Amherst, zamani@cs.umass.edu; Allan Hanbury, TU Wien, allan.hanbury@tuwien.ac.at.

the early days of the neural information retrieval [1, 22]—suffered heavily from off-topic false positive matches under the full retrieval setting which typically then necessitated that dense retrieval be combined with term-based retrieval to achieve reasonable result quality. However, recently there have been several significant improvements [18, 26, 31, 67] in training methodologies for dense retrieval models leading to many top ranking runs on public benchmarks like MS MARCO [13] and various open-domain question answering (QA) datasets [31]. In spite of achieving competitive performance with traditional term-based indexing on different research benchmarks, the question remains whether these dense retrieval systems are ready to replace existing term-based retrieval in practical search systems. The answer to that question requires more than simply comparing these systems based on mean effectiveness metrics on a query workload, and involves careful consideration of tradeoffs between several different costs and effectiveness, as well as the result quality on subsets of the query distribution.

A critical dimension that is often overlooked or under-studied in TREC-style [11, 12] and leaderboard [13] evaluations, is the cost of deploying these models. In production retrieval systems, cost typically can be a combination of several factors—*e.g.*, indexing cost, query processing cost, and even environmental impact of training deep models [4]—and should be measured under the exact conditions in which the system will be deployed.

After testing two or more systems, we may have many observations in terms of cost factors and effectiveness measures. A rational choice for deployment of a system assumes that all choices are alongside the Pareto optimum of cost vs. effectiveness. If a candidate system is not alongside the known Pareto optimum, it would naturally fall out of consideration immediately, because there are better options in all dimensions. However, as soon as we compare systems along the Pareto optimum, we face tradeoff decisions and constraints.

When evaluating effectiveness of ranking models, be it at TREC or on a leaderboard, it is also common practice to compare and rank systems based on performance metrics averaged over a sample set of queries. However, the mean value of the metric may unwittingly hide critical systemic failures and mislead us on the model’s readiness for deployment. For example, previous work [50, 66] has argued for the need to incorporate lexical matching features in neural models to deal with rare terms, especially under the full retrieval setting [34, 51]. It is therefore reasonable to question if dense retrieval methods on their own may systemically fail to retrieve relevant documents when the query contains rare terms. The dense retriever may show other systemic weaknesses, such as under-performing on longer queries or failing to retrieve longer documents, given theoretical constraints on learning fixed sized embeddings for long text [36] or demonstrated proclivity of neural retrieval models towards over-retrieving shorter documents [28] when inspecting only the beginning of documents for efficiency reasons. Dense retrieval models may fail on queries that are out of distribution [9, 13, 63] with respect to the training data. Performance on out-of-distribution queries can be critical for real world search systems where data distributions can change over time, or vary across locations and user demographics. It may even be important to distinguish between different types of failures—*e.g.*, retrieving non-relevant but related documents as opposed to retrieving results that are completely off-topic. Such embarrassing failures can have disproportionate impact on the system’s brand and long term user engagement [16].

In this work, we design a decision framework to systematically compare dense retrieval to term-based indexes to determine their suitability of deployment in real production search systems as a replacement for the latter. Unlike previous work, *e.g.*, [31], we are not concerned about whether dense retrieval outperforms term-based retrieval across different benchmarks. Instead, we are interested to study if dense retrieval can be deployed in the context of a given benchmark or application, and the various detailed considerations that influence that specific decision. This paper has one full case study of applying the decision framework. To almost our surprise, in our chosen test case, carefully-designed dense retrieval models outperform traditional term-based retrieval in almost every dimension, including

analysis designed to highlight problems such as handling of rare terms. However, this is only one scenario and even here, the ultimate shipping decision is constrained by available hardware budget increases and the individual importance of different cost factors.

Designing a decision framework to compare two retrieval systems with the explicit purpose of determining whether one system should replace another is inherently challenging. We note that our proposed framework is general and appropriate for comparing arbitrary search systems. So, in addition to highlighting the efficacy of state-of-the-art dense retrieval methods over traditional search using inverted-indexes, an important contribution of this work is the framework itself. We posit that the proposed framework can guide rational and informed choices between different search architectures and models, albeit, individual cases require individual weighting decisions of which cost is more important.

Our main contributions and findings are summarized as follows:

- We propose a set of criteria to compare retrieval systems with cost-effectiveness tradeoff considerations and guardrails to ensure robustness against failures.
- We propose a general decision framework to guide a decision maker in applying our criteria.
- We implement our framework on one scenario, finding state-of-the-art DR models are ready to replace term-based systems, unless cost of vectorization is the overriding concern.

## 2 RELATED WORK

**Neural Ranking and Dense Retrieval.** Neural ranking models (NRMs) have been widely studied in the past few years and have led to substantial and significant improvements in terms of retrieval effectiveness [24, 49]. Following learning to rank models and their applications in multi-stage cascaded retrieval systems, NRMs are mostly designed to re-rank a small number of documents retrieved by one or more efficient early-stage retrieval models, such as BM25 [57]. Notable models of this category include DRMM [23], Duet [50], KNRM [66], TK [27], and BERT re-ranking [53]. However, the performance of these models is bounded by the recall of the early stage retrieval models. To address this shortcoming, Zamani et al. [68] introduced SNRM, the first standalone neural ranking model that can retrieve documents directly from a large-scale collection. Recently, models use dense representations obtained from pre-trained contextual language models, e.g., BERT [17]. They are often called *dense retrieval* models. ColBERT [32], ANCE [67], RocketQA [18] and TAS-Balanced [26] are among notable dense retrieval models. These dense retrieval models have been recently used in the literature as state-of-the-art NRMs, however, we believe that they have never been appropriately evaluated and studied. Prior work on dense retrieval [26, 55, 64, 67] mostly report the average retrieval performance on a query set. Therefore, it is still an open question to what extent these dense retrieval models can replace the established and robust term matching models that use inverted indexes for efficient retrieval. This paper introduces a comprehensive evaluation framework, which can be applied to make the replacement decision in a given scenario.

**Axiomatic Analysis.** The proposed evaluation framework is slightly related to the axiomatic analysis literature. Axiomatic analysis is a study of retrieval models using a set of well-defined and easy-to-measure constraints (called axioms) and the intuition is that every model should satisfy those axioms. Therefore, axiomatic analysis can provide guidelines for further development of models by highlighting the unsatisfied axioms. Fang et al. [20] introduced axiomatic analysis to information retrieval with the goal of improving term-based retrieval models, such as BM25. Their approach has been further extended to a wide range of models, such as pseudo-relevance feedback [8, 52] and query performance prediction [42]. It has also been employed for improving neural ranking models [56, 58]. Moreover,

Busin and Mizzaro [6] went beyond retrieval models and brought axiomatic analysis to study evaluation metrics, called axiometrics. They suggest a set of axioms that a metric should satisfy. Similar to this body of work, our evaluation framework also consists of a number of criteria or axioms. We believe that making decisions on the use of dense retrieval in search engines requires analysis of multiple proposed criteria. The criteria we introduce in this paper are novel and unlike prior work on axiomatic analysis, some of the criteria we propose may introduce a tradeoff and system designers should make decision based on their needs.

**Efficiency-Effectiveness Tradeoff in IR.** IR systems mainly aim at *retrieving relevant documents* from large-scale collections, *efficiently*. Even though published research mostly focuses on either effectiveness or efficiency considerations, taking both of them into account is at the heart of IR systems. There have been numerous efforts for developing and evaluating IR systems from both of these perspectives [14, 48, 70]. For instance, Asadi and Lin [2] studied efficiency-effectiveness tradeoffs in candidate generation for multi-stage cascaded retrieval systems. Later on, Clarke et al. [7] extended this work to end-to-end multi-stage systems and developed a model that does not require relevance judgement information. Refer to the tutorial by Lucchese and Nardini [37] on the tradeoffs of multi-stage cascaded systems for more detail. Optimizing retrieval systems by considering both effectiveness and efficiency measures is a multi-objective optimization problem. A natural approach to address this optimization problem is to use the Pareto frontier [29, 39, 40].

Unlike previous work in this area which mainly focused on multi-stage cascaded systems, we study dense retrieval as a standalone retrieval model. The efficiency-effectiveness tradeoff in dense retrieval models is relatively unknown and this work will provide suggestions on the practical use of dense retrieval in search engines.

### 3 DECISION CRITERIA

Our goal is to decide whether to replace an old system with a new system. Our case study in this paper considers dense retrieval as the new system with traditional indexing as the old system. This would mean replacing a well-studied and proven retrieval system with a promising, but less understood retrieval system. Dense retrieval has some known potential ‘deal breakers’, which could make it unusable in practice. The cost of vectorization and nearest-neighbor search may be too high. Despite good average effectiveness, there may be subsets of queries where results are extremely bad, perhaps when dealing with rare query terms that were not seen during training. If there is data drift after deployment, the improvements in performance may disappear. To handle these concerns, as in any given application, we suggest identifying a set of application-specific decision criteria.

#### 3.1 Overall Criteria

The starting point when comparing retrieval systems is the mean effectiveness [18, 26, 31, 67] on a dataset that matches the target application. We apply statistical tests to determine if there is a significant difference in means, adopting the following notation.

The operator  $\gg_{sig.T}$  means statistically significantly larger, as tested with a test  $T$ , for two results  $X$  &  $Y$ :

$$X \gg_{sig.T} Y \tag{1}$$

**C-Effective** Using a metric (f.e.  $NDCG@10$ ) the new system  $B$  should be significantly better than system  $A$ :  $B \gg_{sig.T} A$ .

We can also require a margin of improvement, to eliminate tiny but significant improvements, which may be possible in some cases.

Among our overall decision criteria, some could require a significance test, as in Eq. 1. However, other quantities are less suitable for significance testing, such as the size of the index. If the new system requires a much larger index, this can be captured by an absolute threshold (the index size must not exceed 20GB in production) or ratio (the index shall not grow by a factor of five). Such limits, defined by the decision maker, may be of great practical importance, depending on the scenario. It is also possible to define a criterion that combines statistical significance and practical significance, to discount small-but-significant improvements, that may not be worth deploying.

Our decision framework in Section 4 tells us how to make a decision based on multiple criteria. Still, the cost of deploying the system may include a variety of different aspects, such as query latency, indexing throughput, and storage requirements. The decision maker may decide that a particular cost factor is the most important one, for example that query latency is important, but indexing costs can be ignored. Another option is to apply a comparative transform  $\Phi$  to each cost factor, to put them on the same scale, and use a weighted combination to summarize the cost factors.

The comparative transform  $\Phi$  can take many forms, involving any monotonic function, but a simple version is a scaled fraction comparing the new approach's performance  $x$  to baseline performance  $y$  with importance weight  $\alpha$ :

$$\Phi(x, y) = \frac{x}{y} * \alpha \quad (2)$$

By transforming several cost measurements and summing, the decision maker can summarize several cost factors as a single number.

**C-Efficient** *The cost of the new model B should not exceed the cost of the old model A by more than a certain factor (N times) or more than a certain margin (D distance).*

This can be applied on individual cost factors or a transformed and combined cost factor. Once the decision maker has chosen application-specific criteria for overall C-Effective and C-Efficient, they can consider some criteria that do not measure overall performance: robustness criteria.

### 3.2 Robustness Criteria

Overall improvements in effectiveness, particularly when some query performances become a lot better, can hide systematic problems in smaller sub-groups of queries. For example, a system that is better overall might be much worse at handling queries with rare words. This could be a *deal breaker*, if this makes it impossible to retrieve certain content. Users may notice that they can not search for a person by name, if the person's name has rare words that were not seen during model training, and they may reject the new system outright.

**C-Robust** *The new system B should not exhibit systematic failure patterns compared to system A; which might be hidden in the aggregated metrics.*

Observing the result metric  $R$  for a certain subset of model independently selected queries  $\hat{Q}$ , there should be no statistically significant loss:

$$\text{NOT} \left( R_B(\hat{Q}) \underset{\text{sig.T}}{\ll} R_A(\hat{Q}) \right) \quad (3)$$

Now, we are able to define various subsets of our query distribution to study these systematic model differences. The new system B should not categorically fail on specific query characteristics, such as length or term rarity, even if those queries are underrepresented in the data.

In IR we have a rich history of studying query characteristics, common dimensions include the query token length. For **C-Length** we define a set of queries with specific token length between  $m$  and  $n$  as:

$$\hat{Q} = \{q \mid m < \text{len}(q) < n, q \in Q\} \quad (4)$$

Another common categorization of queries is to utilize the query token frequency. We define **C-Frequency** as a set of queries with specific minimum token frequency  $TF$  between  $m$  and  $n$  as:

$$\hat{Q} = \{q \mid m < \min TF(q) < n, q \in Q\} \quad (5)$$

Selecting regions of the query distribution is not limited to these two examples. Any slice of queries is possible, albeit it is important to select them system independently. If the selection depends directly or indirectly on the quality of one of the participating systems, the criterion in Eq. 3 losses its expressiveness.

**C-Lexical** *Novel types of possible failures from a new system  $B$  need to be specifically tested to show that  $B$  is robust.*

In the case of DR models, one such novel failure type can be determined by the subset of queries, which contains a query if the top ranked (up to  $r$ ) passages  $p$  contain only up to  $n$  token overlaps (as defined by the set intersection  $\cap$ ):

$$\hat{Q} = \{q \mid n = |q \cap p|, \text{rank}(q, p) < r, p \in P, q \in Q\} \quad (6)$$

**C-Memory** *If a system relies on machine learning it should be able to handle the open nature of the retrieval problem: Out of distribution query types and topics during inference.*

Given a measure of query similarity  $C$ , we can determine for every training and evaluation query their distance, to observe if very different queries in the evaluation set  $Q$  (here  $Q_{Eval}$  for readability), as defined by the threshold  $\epsilon$ , to the training set  $Q_{Train}$ , still provide similar retrieval performance for the set:

$$\hat{Q} = \{q \mid C(q_T, q) > \epsilon, q \in Q_{Eval}, q_T \in Q_{Train}\} \quad (7)$$

The query similarity  $C$  can range from simple word count overlaps to more sophisticated vector semantic models. A requirement is to incorporate the two query sequences (with potential statistics about all queries  $Q$ ) and output a single value measuring the similarity.

C-Robust is not limited to our defined sets (Eq. 4, 5, 6), as many other approaches fit into its definition. Mackie et al. [41] create a special set of hard queries from the TREC DL Track '19 & '20, with a combination of automatic and manual hardness classification.

**C-Margin** *The new system  $B$  should only regress results on queries up to a threshold compared to the old system  $A$ .*

Formally defined, as the threshold  $t$  between the result metric with at least a distance margin  $\delta$  per query of the old system  $R_A(q)$  and new system  $R_B(q)$  over all queries  $Q$  as:

$$t \geq \frac{|\{R_A(q) - R_B(q) \geq \delta, q \in Q\}|}{|Q|} \quad (8)$$

We define a minimum result change margin to be able to focus on embarrassing failures and not small differences explainable by noise in the evaluation. This is closely related to Robustness Index (RI) [10].

**Additional Important Criteria.** Even though the mentioned criteria are extensive, they may not be fully sufficient for the full consideration of deploying a system. In Section 7, we review a number of additional criteria that decision makers should take into account.

## 4 DECISION FRAMEWORK

After the decision maker has chosen a set of criteria that are suitable for their application, and gathered observations against the criteria, they can apply our decision framework. We would first ask the decision maker to classify each criterion as primary or secondary. Among the primary criteria, we need to see at least one significant improvement to justify launching a new system. A secondary criterion is a ‘guardrail’ or ‘deal breaker’, where we do not require an improvement but a significant regression would cause us to decide against the new system.

Although we cannot tell the decision maker which criteria to use, we believe our proposed dense retrieval criteria are comprehensive, probing for known weaknesses of dense retrieval systems. Other dense retrieval studies could apply the same criteria, but with their own query workload and corpus. For other kinds of information retrieval deployment decision, the decision maker can choose criteria that are appropriate for their setting.

Our decision framework has a *significance rule* that takes into account primary and secondary criteria, and a *Pareto rule* that focuses on tradeoffs between primary criteria. To be deployed, a new system must satisfy both rules.

**Significance Rule.** We summarize the results for each criterion as a win/tie/loss for the new system, denoted by ✓/≈/✗. This is determined via statistical significance tests (e.g. improved mean NDCG) and/or tests of practical significance (e.g., 0.01 NDCG improvement, or 20% query latency reduction). We strongly suggest using statistical significance tests for measurement of system effectiveness, but not for a criterion that is a single number, such as index size. The practical significance test can check whether the index size has grown above some limit (✗). We can also create a combined criterion, which requires both a statistically significant gain and sufficient magnitude of gain (✓). In this case, a small but statistically significant gain would be considered ≈.

Given the per-criterion outcome ✓/≈/✗, the significance rule considers primary and secondary criteria. For primary criteria, there should be some improvement (✓), to make the change worthwhile, and no loss (✗). For example, a statistically significant improvement in NDCG could be enough to satisfy this rule. For secondary criteria, we are looking at ‘deal breaker’ or ‘guard rail’ cases. We do not require a win but there should be no losses (✗). In summary, a system passes the significance rule if it has an improvement on a primary criterion, and no losses on any primary or secondary criteria.

**Pareto Rule.** The Pareto rule considers tradeoffs between criteria without considering significance, making it complementary to the significance rule. Under Pareto analysis, we can consider the old and new system, but can also consider several parameterizations of the new system. In Pareto analysis, if system B is better on one criterion than system A, without being worse on any other, then B is a Pareto improvement over A, so we do not need to consider A. If no system is better than system B on any criterion without being worse on some other, then B is Pareto optimal. The set of Pareto optimal systems is the Pareto frontier.

For example, if a system has both higher NDCG and lower query latency than another system, then we can discard the latter system, if we are only considering those two criteria. If one system has higher NDCG and lower latency, while the other has higher latency and lower NDCG, then we should not discard either. Both are on the Pareto frontier.

If the old system is not on the Pareto frontier, it is Pareto dominated by the new systems. In any case, we should choose a Pareto optimal system, since otherwise it is possible to do better on a criterion without doing worse on any others. One way of doing so is to transform each criterion into a comparable scale, by selecting an appropriate monotonic transform  $\Phi$ , then we can simply choose the alternative furthest from the origin. A related option is to consider the change in total cost of running the search system, and trade this off against the magnitude of the NDCG

improvement. This requires the decision maker to have some notion of the value of an NDCG improvement, but having some feel for this is probably fundamental to making any decision related to search quality, so we leave it to the decision maker to consider the return on investment in their application setting.

## 5 INTENDED AUDIENCE

The decision criteria and framework proposed in this work can be helpful for various stakeholders in the field of information retrieval. Without the loss of generality, we describe some common uses:

**Practitioners** could adapt our framework and decision criteria to their specific setting and use it to guide them, whether they should consider deploying a neural retrieval system in place or in addition of a traditional term-based retrieval system.

**Paper reviewers** could use our framework as a guide to critically inspect novel claims, especially if these claims only focus on either the cost benefits or the effectiveness of a new method.

**Paper authors** may use the full or parts of our framework to evaluate their proposed methods. Our robustness criteria can function as a strong secondary evaluation aspect for novel retrieval systems, if the main overall cost-effectiveness remains inconclusive.

## 6 CASE STUDY

In this section, we showcase our framework in a case study on dense retrieval models using MS MARCO and TREC Deep Learning Tracks data. Since we expect dense retrieval to have better search effectiveness but with higher cost, our primary criteria are NDCG@10 (requiring a statistically significant difference with a two-tailed paired t-test;  $p < 0.05$ ; C-Effective ✓) and an aggregated cost that combines query latency and indexing costs (Eq. (11)).

Making decisions about cost factors is not as general as effectiveness results. Different practitioners are likely constrained in different ranges, therefore we use our decision framework to showcase different decision makers, without making a general claim. Providing a single answer to the question *Are we there yet (as a community)?*, however convenient is simply not possible, due to the diverse nature of the retrieval settings. However, we provide the tools to answer specific use cases and settings for the question: *Are we there yet (in our setting)?* We carry out a Pareto analysis on our primary criteria, considering several alternative new systems for our specific use case.

We also consider several robustness criteria, which we count as secondary, where we are looking for failures rather than improvements (C-Robust ✗).

**The Dense Retrieval Model.** In our experiments, we use the BERT<sub>DOT</sub> model as the dense retrieval system. It uses two independent BERT computations (each time pooling the CLS vector output) to obtain the query  $q_{1:m}$  and passage  $p_{1:n}$  representations. It then computes the retrieval score based on the dot product similarity of the two representations:

$$\begin{aligned}\vec{q} &= \text{BERT}([\text{CLS}; q_{1:m}]) \\ \vec{p} &= \text{BERT}([\text{CLS}; p_{1:n}]) \\ \text{BERT}_{\text{DOT}}(q_{1:m}, p_{1:n}) &= \vec{q} \cdot \vec{p}\end{aligned}\tag{9}$$

This architecture decouples the costly encoding from the search. We can store every passage in an (approximate) nearest neighbor index  $I$  for direct vector-based retrieval. The retrieval of the top  $k$  hits for a given query  $q$  is then formalized as:

$$\text{top}_k \{ \vec{q} \cdot \vec{p} \mid \vec{p} \in I \}\tag{10}$$



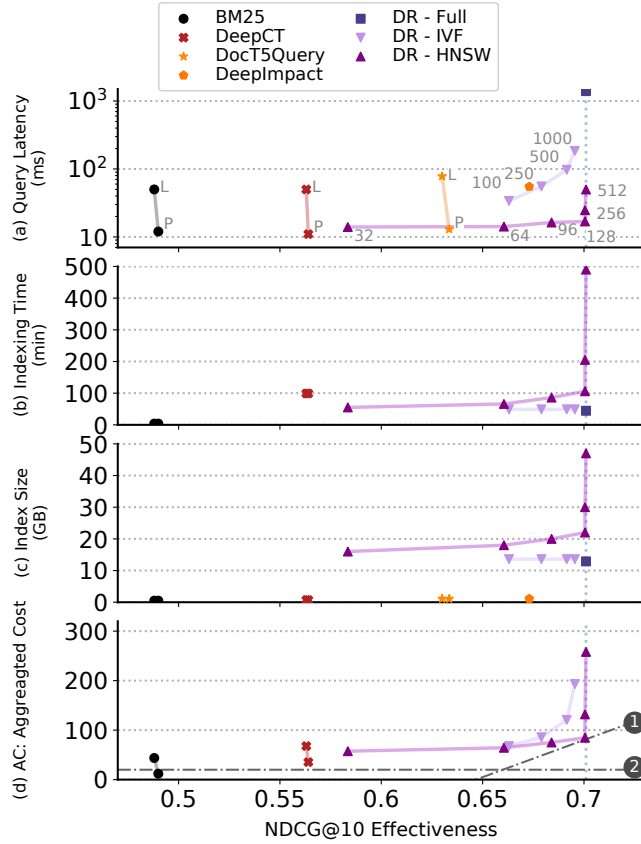


Fig. 1. TREC-DL passage retrieval comparison of the cost-effectiveness tradeoff for term-based and dense retrieval.

In this study, we use the *Standalone* and *TAS-Balanced* trained instances of  $BERT_{DOT}$ , developed by Hofstätter et al. [26]. The *Standalone* version is trained with binary relevance labels from MS MARCO [3]. The *TAS-Balanced* retriever is trained with pairwise and in-batch negative knowledge distillation using topic-aware sampling to compose batches. It is currently a state-of-the-art training technique for dense retrieval [63].

**Term-based Retrieval Baselines.** We use BM25 [57] and the neural augmented indexes DeepCT [15], DocT5query [54], and DeepImpact [44] using both Lucene [35] as well as highly tuned results with PISA [45] reported by Mallia et al. [44].

**Datasets.** We conduct our experiments on the MS MARCO-v1 and TREC 2019-20 Deep Learning Track collections with 9 million passages and 3 million documents. Following Xiong et al. [67], for the document collection we utilize two approaches: (1) taking only the first passage of a document (FirstP), and (2) using the maximum passage score as document score (MaxP).

## 6.1 Primary Criteria

We first apply the decision framework to our primary criteria, NDCG@10 and aggregated cost.

**Study Goal.** Identify a solution that has a significant NDCG@10 gain (significance rule) and the right tradeoff of NDCG@10 improvement and aggregated cost (Pareto rule). Since the tradeoff depends on the decision maker’s priorities, we consider a decision maker that is willing to make a tradeoff (❶) and one that is very cost-sensitive (❷).

**Study Design.** We utilize the mentioned TAS-Balanced trained dense retriever with three search approaches: (1) exhaustive search; and (2) the approximate nearest neighbor (ANN) method based on Hierarchical Navigable Small World graph (HNSW) [43]; and (3) the ANN method based on Inverted Files (IVF) [60]. We use the Faiss library [30] for conducting our experiments. We modulated HNSW and IVF with different hyper-parameter settings that guide the internal tradeoff between cost and quality (i.e., the number of graph-node neighbors and the number of lookup clusters). For measuring cost, we use encoding & indexing time (using a GPU), storage requirement, and single average query latency (using a CPU). For measuring effectiveness, we use NDCG@10 averaged over the TREC-DL query sets. We conducted our experiments on a system with a 16 core Intel Xeon E5 and TITAN RTX GPU.

The aggregation of cost factors – such as latency  $l$ , indexing time  $i$ , and storage  $s$  – is highly dependent on the scenario: whether we have a high query workload, a high document update frequency, or other requirements. Here, we instantiate our cost side of C-Tradeoff with an exemplary balanced aggregated cost (AC) anchored to BM25 per method  $m$ :

$$AC_m = \frac{l_m}{l_{BM25}} * \alpha + \frac{i_m}{i_{BM25}} * \beta + \frac{s_m}{s_{BM25}} * \gamma \quad (11)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  control the importance of each cost component. In Figure 1, we set  $\alpha$  to 10, and  $\beta$  &  $\gamma$  to 1. As this weighting has major impact on the decision and conclusion, we also provide alternative weighting results in Appendix A. We model a decision maker who cares much more about relative increases in response time, since these affect users directly, than they do about increases in indexing costs, perhaps because they already have sufficient available resources to handle a new system with moderate requirements.

**Study Analysis.** For the passage dataset, we show our findings combined in Figure 1. Each subfigure shares the effectiveness on the x-axis. All improvements over BM25 are statistically significant (C-Effective ✓), making many systems viable on that criterion. We will discuss cost factors then move on to our efficiency criterion.

On top in Figure 1 (a) we compare the effectiveness with the log-scaled query latency for a single query (on CPUs). Query latency is one of the most common used cost metrics in IR and highly influential for user satisfaction [33, 68]. Here, we observe DR with HNSW approximation is Pareto dominant over other DR approaches. If we imagine a decision maker who mainly cares about the NDCG@10 and latency, they would be likely to select HNSW 128.

If we instead look at indexing time and index storage requirements in Figure 1 (b) and (c) respectively, HNSW exhibits much higher cost than a simple inverted index or the alternative ANN method IVF. Concurrently with this work, compression techniques have been proposed to reduce the required storage [38, 69], however they also follow a tradeoff pattern, reducing their effectiveness as well. DocT5Query & DeepImpact (which uses DocT5Query) are magnitudes slower (with 19,200 minutes) than all other approaches at indexing time, therefore they are not visible in Figure 1 (b) & (d).

We show combined costs with our exemplary aggregation formula in Eq. 11 in Figure 1 (d). Here, we added two decision lines (marked with ❶ and ❷) for our two exemplary types of decision makers. Each line indicates a set of

Table 1. Document retrieval results for TREC-DL query sets. Line # superscript indicates stat.sig. improvement; paired t-test ( $p < 0.05$ ). nDCG cutoff at 10, Recall at 100.

|   | Model           | TREC'19                  |                          | TREC'20                   |                          | Index Size |
|---|-----------------|--------------------------|--------------------------|---------------------------|--------------------------|------------|
|   |                 | MRR                      | Rec.                     | nDCG                      | Rec.                     |            |
| 1 | BM25            | .523                     | .581                     | .507                      | .706                     | 2.3 GB     |
| 2 | DocT5Query      | .597                     | .599                     | .589                      | .759                     | 2.5 GB     |
| 3 | DR+Full: FirstP | .630                     | .556                     | .598                      | .705                     | 5 GB       |
| 4 | DR+Full: MaxP   | <b>.636</b> <sup>1</sup> | <b>.610</b> <sup>5</sup> | <b>.639</b> <sup>15</sup> | <b>.757</b> <sup>5</sup> | 10 GB      |
| 5 | DR+HNSW: FirstP | .607                     | .542                     | .586                      | .684                     | 8 GB       |
| 6 | DR+HNSW: MaxP   | .606                     | .561                     | .630 <sup>1</sup>         | <b>.760</b> <sup>5</sup> | 18 GB      |

solutions that the decision maker would consider equally good, intersecting with the Pareto frontier. A decision maker that is willing to make some tradeoff ❶ would choose the bottom right point, which is HNSW-128 (C-Efficient ✓). However, if absolutely no cost increase is allowed, as in scenario ❷, then all neural approaches would be rejected (C-Efficient ✗).

For the document dataset, we are particularly interested in the recall of dense retrieval methods, as both the FirstP and MaxP approaches used cut off the text (at 512 and 4000 tokens respectively) and do not index every single word, as the term-based indexes do. Our results on the TREC-DL query sets in Table 1 show a similar trend to our passage retrieval in Figure 1: Dense retrieval, both in full and approximate (HNSW with 128 neighbors) settings, outperforms term-based indexes in most cases. Looking at the recall, we observe that a MaxP approach is needed to substantially outperform term-based methods. However, MaxP naturally increases the storage requirements compared to FirstP, as more vectors need to be stored.

To conclude, in our application setting with MS MARCO data, dense retrieval models with the right ANN hyperparameter choice can deliver both an equally low query latency and higher effectiveness than term-based methods. However, this comes at the cost of increased indexing and storage resource requirements. The decision to replace term-based retrieval with dense retrieval is therefore constrained based on individual budgets and constraints. Next we consider secondary significance criteria.

## 6.2 Robustness by Query Characteristic

The mean effectiveness gains of dense retrieval (even with ANN search) compared to term-based approaches are large. DR could completely fail on a large subset of queries and still perform better than a baseline on average. However, this can be a practical dealbreaker. Therefore, we make use of our secondary guardrail criteria to evaluate retrieval models by query characteristics.

**Study Goal.** We address the following question: *Do DR models struggle on certain queries categorized by length or term frequency?*

**Study Design.** We utilize the criterion C-Robust to compare approaches by query characteristic, specifically we make use of length based C-Length and frequency based C-Frequency. We evaluate term-based and exhaustive TAS-Balanced DR on the large MSMARCO-DEV-49K passage query set (with almost 49 thousand queries; distinct from the training set) in terms of MRR@10. The reason for switching from TREC-DL to MSMARCO-DEV for this experiment is

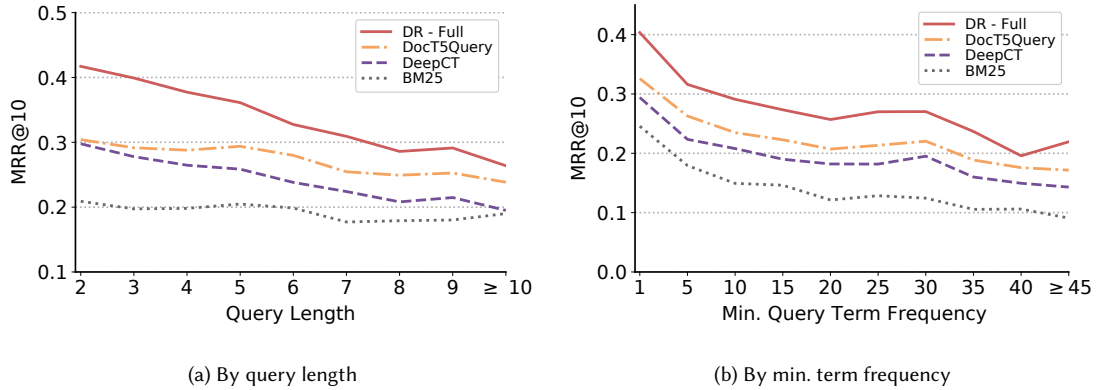


Fig. 2. Comparison of MRR@10 effectiveness results on MSMARCO-DEV-49K by query characteristic.

Table 2. Annotation analysis of queries from MSMARCO-DEV-Large (49K queries) with zero or one subword overlap on the top-1 retrieved passage by the BERT<sub>DOT</sub> dense retrieval model.  $\Delta$  P@1 shows the difference to the DEV-7K set.

| Training     | Common Tokens@1 | Queries |         | Query Tokens |          | Our Annotations |              |                                 |     |
|--------------|-----------------|---------|---------|--------------|----------|-----------------|--------------|---------------------------------|-----|
|              |                 | #       | %       | Total        | Subwords | P@1             | $\Delta$ P@1 | 4-Graded Relevance Distribution |     |
| None         | 0               | 22,600  | 46.50 % | 6.0          | 15%      | .000            | —            | 100%                            |     |
|              | 1               | 14,640  | 30.12 % | 6.9          | 13%      | .000            | —            | 100%                            |     |
| Standalone   | 0               | 163     | 0.33 %  | 5.0          | 16%      | .313            | -.435        | 21% 10% 13%                     | 55% |
|              | 1               | 1824    | 3.75 %  | 4.3          | 7%       | .580            | -.168        | 45% 13% 16%                     | 26% |
| TAS-Balanced | 0               | 37      | 0.01 %  | 4.0          | 8%       | .622            | -.173        | 43% 19% 11%                     | 27% |
|              | 1               | 1284    | 2.64 %  | 3.9          | 3%       | .715            | -.080        | 57% 14% 14%                     | 14% |

that we benefit from using as many test queries as possible to reduce noise with uncommon characteristics. Note that the TREC-DL data contains less than 100 queries.

**Study Analysis.** We present the analysis by query length in Figure 2a and by minimum term frequency in Figure 2b. In both cases, we observe dense retrieval performing better than term-based alternatives. This is especially surprising for queries with very rare terms (Figure 2b). Even in cases where a query term is essentially unseen during training, the TAS-Balanced approach which does explicitly match terms is outperforming term-based approaches. These are positive results showing that DR is consistent in the improvements and therefore DR does not fail our C-Robust tests ( $\approx$ ).

### 6.3 Lexical Match Robustness

Retrieving passages from an unconstrained vector space opens up a new failure scenario that term-based retrieval models cannot suffer from: returning passages without any term overlaps.

**Study Goal.** We aim at answering the following question: *How well do different DR training approaches solve the lexical matching task?* We measure the extent of queries, where a DR model returns a top passage with low or no lexical text overlap. We constrain the problem to the first returned passage and the P@1 metric. For this case study, we instantiate our guardrail criteria C-Lexical.

Table 3. Training and test set MRR@10 results for TAS-Balanced using different training cluster-splits. *Line # superscript indicates stat.sig. improvement; paired t-test ( $p < 0.05$ ).*

|   | Training Data       | Ratio | Train (400K)                |                             | Test (49K)                 |                            |
|---|---------------------|-------|-----------------------------|-----------------------------|----------------------------|----------------------------|
|   |                     |       | All                         | C-Subset                    | All                        | C-Subset                   |
| 1 | BM25                | –     | .172                        | .173                        | .194                       | .190                       |
| 2 | All                 | 1.0   | <b>.345</b> <sup>1345</sup> | <b>.346</b> <sup>135</sup>  | <b>.340</b> <sup>134</sup> | <b>.338</b> <sup>134</sup> |
| 3 | Uniform Reduction   | 0.1   | .299 <sup>14</sup>          | .300 <sup>1</sup>           | .315 <sup>14</sup>         | .310 <sup>1</sup>          |
| 4 | Subset-Clusters     | 0.1   | .280 <sup>1</sup>           | <b>.429</b> <sup>1235</sup> | .298 <sup>1</sup>          | .321 <sup>13</sup>         |
| 5 | Non-Subset-Clusters | 0.9   | <b>.343</b> <sup>134</sup>  | <b>.302</b> <sup>1</sup>    | <b>.339</b> <sup>134</sup> | <b>.334</b> <sup>134</sup> |

**Study Design.** We use Eq. 6 to select a set  $\hat{Q}$  of queries, using the highest ranked passage by our dense retrieval models. We assume that cases without any ( $n = 0$ ) or only one ( $n = 1$ ) subword-token overlap between the query and the highest ranked passage represent hard-failures of missing lexical matches.

Using exhaustive search, we compare three different methods: (1) a pre-trained DistilBERT model without fine-tuning; (2) a standalone fine-tuned model; and (3) the mentioned dense retrieval model based on TAS-Balanced. To receive robust results, we again conduct this study on the large MSMACRO-DEV-49K set. Furthermore, we manually judged all selected queries and their first retrieved passages (for the non-finetuned baseline we sampled 100 queries). The judgements were conducted by non-expert annotators on a 4-graded relevance scale, following the TREC-DL definition (Relevant: Perfect & Partial; Non-Relevant: Topic & Wrong). The annotators had to select relevant text spans for the two relevant classes, reducing the chance of inadvertent false positives. With this we can confidently draw conclusions from this study that would not be possible with the noisy incomplete relevance judgements from MS MARCO.

**Study Analysis.** In Table 2, we report the resulting query set sizes as well as our annotation results. First, we observe that our pre-trained only baseline completely fails - returning an irrelevant top-1 result with zero or one matching token for the majority of queries. This shows that there is indeed no guarantee that dense retrieval can do token matching or term (token n-gram) matching. However, when we turn towards the fine-tuned models with relevance data, we see that the standalone training already reduces the ratio of queries without matches to .33%. The state-of-the-art TAS-Balanced retriever reduces this number further to .01% and the number of queries with a single match to 2.64%. Our annotations show that the true failures of these query sets are even smaller, because the P@1, while lower than the average P@1 of a random query sample, is still above .622 (no common tokens) and .715 (1 common token) for TAS-Balanced. To conclude, we only observe a practically insignificant fraction of queries where a missing term match is the cause of a non-relevant top retrieved passage, therefore TAS-Balanced does pass our C-Robust tests ( $\approx$ ).

#### 6.4 Memorization vs. Generalization

In contrast to many other traditional retrieval models which only have a few hyper-parameters to tune on a collection, dense retrieval models contain millions of parameters and require large-scale training data. This opens the possibility that DR models just memorize the query distribution (even if the test queries are technically distinct from the training set) and what looks like generalization across different query sets is actually memorization, as posited by us in C-Memory.

**Study Goal.** We aim to observe the robustness of dense retrieval to reduced training data and answer the question: *Does the effectiveness of DR models come from memorization or a mix with generalization capabilities?*

Table 4. Failure analysis of queries selected from MSMARCO-DEV (49K queries) with BM25 having an RR=1 and dense models RR=0 on sparse judgements. We annotated the top-1 retrieved result from the dense models.

| Training     | Queries |       | P@1  | Our Annotations                 |     |     |     |
|--------------|---------|-------|------|---------------------------------|-----|-----|-----|
|              | #       | %     |      | 4-Graded Relevance Distribution |     |     |     |
| Standalone   | 911     | 1.87% | .528 | 37%                             | 16% | 28% | 19% |
| TAS-Balanced | 472     | 0.97% | .638 | 48%                             | 16% | 27% | 9%  |

**Study Design.** We utilize the topic-cluster-guided training of TAS-Balanced as query-similarity measure to reduce the training to different cluster subsets. Queries are assigned a cluster using a baseline DR representation with dot-product similarity. We compare the following training splits in Table 3: training on all 2000 query clusters (row 2); training on 10% randomly selected queries, but using all clusters (row 3); and then our cluster-subsets with training on 10% of clusters (row 4), as well as training on the remaining 90% (row 5). We then evaluate these model instances on the training and the test set (both split in all available queries and queries associated with the 10% subset clusters).

**Study Analysis.** In Table 3 we see in rows 1, 2, & 3 that BM25 and training on all clusters produces stable results with low margins between all and c-subset for both training and testing. As before we observe DR to vastly outperform BM25, albeit slightly reduced when we only train on 10% of the training data. Even the strongly reduced-training DR instances pass our C-Robust tests (✓) compared to BM25. If we only train on a 10% cluster-subset (row 4) we see that it strongly memorizes the training set, outperforming the other DR methods substantially on the trained clusters. But on the test set it is outperformed on the same query clusters by the full training (row 2). Similarly, the 10% cluster-subset training is also outperformed by more training data that does not contain queries from the evaluated clusters (row 5). To conclude, we do observe a tendency to memorize the training set, if the number of queries is small. A 10% cluster-subset would fail our C-Robust tests (✗) compared to the other DR instances, but not to BM25 (✓). A 10% drift in the query distribution from train to test (row 5) is as robust as no drift at all ( $\approx$ ).

## 6.5 Per Query Robustness of Improvement

Commonly in IR, the wins and losses on a query level between two competitive systems will always have some queries on both sides. The important factor is how many losses are tolerated, as we set in C-Margin. Once a system is over a given threshold of missed queries, we need to reconsider deployment.

**Study Goal.** In the previous studies (Sec. 6.2, 6.3, 6.4) we sliced dense retrieval results by query characteristics and effectiveness independent result properties. Now, we start our study from the other side and select queries using evaluation metrics. We want to answer: *How large is the query set with the highest failure margin between DR and term-based models?*

**Study Design.** We select the queries from MSMARCO-DEV-49K with the largest RR@10 margin between BM25 and BERT<sub>DOT</sub>. For our analysis we set  $\delta = 1$  in Eq. 8, the maximum possible margin. Furthermore, to gain clarity on the actual result quality, without sparse label noise, we follow our labelling approach from Section 6.3 to annotate the relevance of the first retrieved passage.

**Study Analysis.** In Table 4, we show the results for standalone and TAS-Balanced trained dense retrieval models. BM25 shows a RR@1 and P@1 of 1 for the selected queries. The number of queries where BM25 outperforms BERT<sub>DOT</sub>

is 1.87% using a standalone training and 0.97% for TAS-Balanced. Judging from the sparse labels, this would already be a small number, albeit it could cause concern if the new search system fails on that many queries. However, our annotations show that while not all queries are answered, the true P@1 loss compared to BM25 for this query set is even smaller. The difference between standalone and TAS-Balanced shows that the small number of failures against BM25 are not endemic to dense retrieval, rather they can be even reduced further by better training techniques.

If we set the threshold of allowed failures to a reasonable, albeit arbitrary, 1% of queries we observe that TAS-Balanced passes our C-Robust tests ( $\approx$ ) compared to BM25.

## 6.6 Overall Decision

When probing TAS-Balanced on the MS MARCO datasets, we surprisingly find that it passed all our robustness tests that were designed to target suspected flaws in dense retrieval.

The main drawback of dense retrieval is the cost of vectorization and building the ANN index. In an application that needs to run at extremely low cost, the decision maker might put a higher weight on indexing size and time than they do on latency. They might also choose a point on the Pareto frontier, which loses significant NDCG@10 in order to greatly reduce cost. If 100 minutes of indexing and 22GB is impossible, for example on a very low-cost application, then the decision maker may choose the traditional index (tradeoff ②). However, we think that in many real applications the cost in minutes and gigabytes is affordable, so for our case study – using MSMARCO – we would expect the decision maker to mostly choose to replace term-based indexing with the new DR approach (tradeoff ①).

## 7 ADDITIONAL CRITERIA

In addition to the criteria introduced in Section 3, there are many other important considerations, ranging from critical externalities to technical debt, that should be emphasized in any deployment decision. In this section, we review several of these additional criteria that decision makers should pay attention to. These considerations are also important in the context of dense and term-based retrieval systems and should be studied in future work.

**C-Bias** Search engines, like other large-scale information access systems, act as gatekeepers to the world’s information. The ranking of results that these systems produce directly influences what information and content users are exposed to. When deploying new systems, it is therefore important to also consider potential representational and allocative harms that may result from biases encoded in these socio-technical systems. For example, large deep learning models are known to not only pick up historical societal biases present in most training datasets, but can often amplify them, leading to harmful stereotyping of and promote negative sentiments towards marginalized groups [4]. Biases in the models can also raise concerns around user-side and producer-side fairness [19, 47]. Therefore any system deployment decisions must ascertain that all benefits of the new system must be equitably distributed across different demographic groups and does not introduce any significant new societal harms.

**C-Environment** As the worldwide scientific community rings the warning bell on critical dangers of continuing climate change [46] and different institutions, commercial and otherwise, move towards more ambitious reduction of their carbon footprint and negative ecological impact (*e.g.*, [61]), the environmental cost becomes an increasingly critical criteria to be considered in any deployment decisions. The research community has recently raised various concerns on the environmental impact of large-scale deep learning models [4]. Same arguments also hold when these models are applied to retrieval models. Therefore, the environmental impact, *e.g.*, carbon footprint, of the deployed system should be also considered in the decision making process. Many of these impacts are measurable, *e.g.*, see [62].

**C-Maintainability** Commercial search engines are large and complex systems often deployed over large-scale distributed cloud infrastructure. In this paper, we have focused on the cost and benefit of a new system measured at the point of deployment, which misses the additional cost associated with maintaining and further improving the system post deployment. For a machine-learned dense retrieval system, this includes the cost of incremental updates to the index as new documents are discovered and added to the collection, as well as old document contents are refreshed and re-indexed. Over time, as the volume and the distribution of documents in the collection and the query workload naturally evolve, it may become necessary to periodically re-train and re-deploy the machine-learned models. This introduces additional efficiency-effectiveness tradeoffs associated with future maintenance of any deployed system that decision makers must account for. Neglecting these considerations can result in build up of technical debts in machine learning based systems [59].

## 8 CONCLUSIONS

Given the recent popularity of dense retrieval models, this paper takes a deeper look to evaluate these models and provide tools to answer the main question: *Are we there yet? Can we switch from a term-based to a dense retrieval system?* We described a general framework for evaluating retrieval models based on multiple criteria that cover effectiveness, efficiency, and robustness measures.

In our case study, we observed that state-of-the-art ANN models can provide significantly higher effectiveness with similar query latency to term-based models. However, they have substantially higher cost in terms of storage usage and indexing time. Surprisingly, we observed that dense retrieval models deliver consistently better search results for queries with different length, different query term frequency, and robust lexical match capabilities.

More broadly for the field and in practical applications, the answer to the question *are we there yet*, will be determined by many evaluations, many case studies. The proposed decision framework provides a guideline for systematic evaluation of new retrieval models which can be used by search engine practitioners to make informed decision about new retrieval models before deploying them into production. To say whether DR becomes a ubiquitous solution, perhaps even supplanting traditional indexing, requires many decisions to be made and many application-specific scenarios. We are certainly not there yet, but through many such studies we will find out.

## REFERENCES

- [1] Qingyao Ai, Liu Yang, Jiafeng Guo, and W Bruce Croft. 2016. Analysis of the paragraph vector model for information retrieval. In *Proc. of ICTIR*.
- [2] Nima Asadi and Jimmy Lin. 2013. Effectiveness/Efficiency Tradeoffs for Candidate Generation in Multi-Stage Retrieval Architectures. In *Proc. of SIGIR*.
- [3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew Mcnamara, Bhaskar Mitra, and Tri Nguyen. 2016. MS MARCO : A Human Generated MACHine Reading COMprehension Dataset. In *Proc. of NIPS*.
- [4] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?  In *Proceedings of FAccT 2021*.
- [5] Leonid Boytsov, David Novak, Yury Malkov, and Eric Nyberg. 2016. Off the beaten path: Let's replace term-based retrieval with k-nn search. In *Proc. of CIKM*.
- [6] Luca Busin and Stefano Mizzaro. 2013. Axiometrics: An Axiomatic Approach to Information Retrieval Effectiveness Metrics. In *Proc. of ICTIR*.
- [7] Charles L. Clarke, J. Shane Culpepper, and Alistair Moffat. 2016. Assessing Efficiency—Effectiveness Tradeoffs in Multi-Stage Retrieval Systems without Using Relevance Judgments. *Inf. Retr.* 19, 4 (Aug. 2016), 351–377.
- [8] Stéphane Clinchant and Eric Gaussier. 2013. A Theoretical Analysis of Pseudo-Relevance Feedback Models. In *Proc. of ICTIR*.
- [9] Daniel Cohen, Bhaskar Mitra, Katja Hofmann, and W Bruce Croft. 2018. Cross domain regularization for neural ranking models using adversarial learning. In *Proc. of SIGIR*.
- [10] Kevyn Collins-Thompson. 2009. Reducing the Risk of Query Expansion via Robust Constrained Optimization. In *Proc of CIKM*. Association for Computing Machinery, New York, NY, USA, 837–846.



- [11] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2019. Overview of the TREC 2019 deep learning track. In *TREC*.
- [12] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2020. Overview of the TREC 2020 deep learning track. In *TREC*.
- [13] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. MS MARCO: Benchmarking Ranking Models in the Large-Data Regime. In *Proc. of SIGIR*.
- [14] Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice*. Addison-Wesley.
- [15] Zhuyun Dai and Jamie Callan. 2019. Context-Aware Sentence/Passage Term Importance Estimation For First Stage Retrieval. *arXiv:1910.10687 [cs.IR]*
- [16] Ovidiu Dan and Brian D. Davison. 2016. Measuring and Predicting Search Engine Users' Satisfaction. *ACM Comput. Surv.* (2016).
- [17] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL*.
- [18] Yingqi Qu Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2020. RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering. *arXiv preprint arXiv:2010.08191* (2020).
- [19] Michael D Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. 2021. Fairness and Discrimination in Information Access Systems. *arXiv preprint arXiv:2105.05779* (2021).
- [20] Hui Fang, Tao Tao, and ChengXiang Zhai. 2004. A Formal Study of Information Retrieval Heuristics. In *Proc. of SIGIR*.
- [21] John F. Fenlon. 1913. The Catholic Encyclopedia, volume 4. [https://en.wikisource.org/wiki/Catholic\\_Encyclopedia\\_%281913%29/Concordances\\_of\\_the\\_Bible](https://en.wikisource.org/wiki/Catholic_Encyclopedia_%281913%29/Concordances_of_the_Bible). Accessed: 2021-08-05.
- [22] Debasis Ganguly, Dwaipayan Roy, Mandar Mitra, and Gareth JF Jones. 2015. Word embedding based generalized language model for information retrieval. In *Proc. of SIGIR*.
- [23] Jiafeng Guo, Yixing Fan, Qingyao Ai, and Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *Proc. of CIKM*.
- [24] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2020. A Deep Look into neural ranking models for information retrieval. *Information Processing & Management* (2020).
- [25] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proc. of SIGIR*.
- [26] 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 *Proc. of SIGIR*.
- [27] S. Hofstätter, N. Rekabsaz, M. Lupu, C. Eickhoff, and A. Hanbury. 2019. Enriching Word Embeddings for Patent Retrieval with Global Context. In *Proc. of ECIR*.
- [28] Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. 2020. Local Self-Attention over Long Text for Efficient Document Retrieval. In *Proc. of SIGIR*.
- [29] Ko-Jen Hsiao, Jeff Calder, and Alfred O. Hero III. 2015. Pareto-Depth for Multiple-Query Image Retrieval. *IEEE Trans. Image Process.* 24, 2 (2015), 583–594.
- [30] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2021).
- [31] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proc. of EMNLP*.
- [32] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proc. of SIGIR*.
- [33] Ron Kohavi, Alex Deng, Brian Frasca, Toby Walker, Ya Xu, and Nils Pohlmann. 2013. Online controlled experiments at large scale. In *Proc. of SIGKDD*.
- [34] Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Leveraging semantic and lexical matching to improve the recall of document retrieval systems: a hybrid approach. *arXiv preprint arXiv:2010.01195* (2020).
- [35] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: An Easy-to-Use Python Toolkit to Support Replicable IR Research with Sparse and Dense Representations. *arXiv:2102.10073 [cs.IR]*
- [36] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, Dense, and Attentional Representations for Text Retrieval. *arXiv preprint arXiv:2005.00181* (2020).
- [37] Claudio Lucchese and Franco Maria Nardini. 2017. Efficiency/Effectiveness Trade-Offs in Learning to Rank. In *Proc. of ICTIR*.
- [38] Xueguang Ma, Minghan Li, Kai Sun, Ji Xin, and Jimmy Lin. 2021. Simple and Effective Unsupervised Redundancy Elimination to Compress Dense Vectors for Passage Retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.
- [39] Xu Ma, Pengjie Wang, Hui Zhao, Shaoguo Liu, Chuhan Zhao, Wei Lin, Kuang-Chih Lee, Jian Xu, and Bo Zheng. 2021. Towards a Better Tradeoff between Effectiveness and Efficiency in Pre-Ranking: A Learnable Feature Selection Based Approach. In *Proc. of SIGIR*.
- [40] Joel Mackenzie and Alistair Moffat. 2020. Examining the Additivity of Top-k Query Processing Innovations. In *Proc. of CIKM*.
- [41] Iain Mackie, Jeffrey Dalton, and Andrew Yates. 2021. How Deep is your Learning: the DL-HARD Annotated Deep Learning Dataset. In *Proc. of SIGIR*.
- [42] Victor Makarenkov, Bracha Shapira, and Lior Rokach. 2015. Theoretical Categorization of Query Performance Predictors. In *Proc. of ICTIR*.
- [43] Yu. A. Malkov and D. A. Yashunin. 2018. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *arXiv:1603.09320 [cs.DS]*
- [44] Antonio Mallia, Omar Khattab, Torsten Suel, and Nicola Tonello. 2021. Learning passage impacts for inverted indexes. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1723–1727.

- [45] Antonio Mallia, Michal Siedlaczek, Joel Mackenzie, and Torsten Suel. 2019. PISA: Performant Indexes and Search for Academia. In *Proceedings of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, OSIRRC@SIGIR 2019, Paris, France, July 25, 2019*. 50–56. <http://ceur-ws.org/Vol-2409/docker08.pdf>
- [46] V. Masson-Delmotte, P. Zhai, and et al. 2021. IPCC, 2021: Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change.
- [47] Rishabh Mehrotra, Ashton Anderson, Fernando Diaz, Amit Sharma, Hanna Wallach, and Emine Yilmaz. 2017. Auditing search engines for differential satisfaction across demographics. In *Proc. of WWW*.
- [48] Sewon Min et al. 2021. NeurIPS 2020 EfficientQA Competition: Systems, Analyses and Lessons Learned. *arXiv preprint arXiv:2101.00133* (2021).
- [49] Bhaskar Mitra, Nick Craswell, et al. 2018. An Introduction to Neural Information Retrieval. *Foundations and Trends® in Information Retrieval* 13, 1 (2018).
- [50] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. In *Proc. of WWW*.
- [51] Bhaskar Mitra, Sebastian Hofstätter, Hamed Zamani, and Nick Craswell. 2021. Improving Transformer-Kernel Ranking Model Using Conformer and Query Term Independence. In *Proc. of SIGIR*.
- [52] Ali Montazerlghaem, Hamed Zamani, and Azadeh Shakeri. 2016. Axiomatic Analysis for Improving the Log-Logistic Feedback Model. In *Proc. of SIGIR*.
- [53] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).
- [54] Rodrigo Nogueira and Jimmy Lin. 2019. From doc2query to docTTTTTquery. *Online preprint* (2019).
- [55] Prafull Prakash, Julian Killingback, and Hamed Zamani. 2021. Learning Robust Dense Retrieval Models from Incomplete Relevance Labels. In *Proc. of SIGIR*.
- [56] Daniel Rennings, Felipe Moraes, and Claudia Hauff. 2019. An Axiomatic Approach to Diagnosing Neural IR Models. In *Proc. of ECIR*.
- [57] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In *Proc. of TREC*.
- [58] Corby Rosset, Bhaskar Mitra, Chenyan Xiong, Nick Craswell, Xia Song, and Saurabh Tiwary. 2019. An Axiomatic Approach to Regularizing Neural Ranking Models. In *Proc. of SIGIR*.
- [59] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. *Proc. of NeurIPS* (2015).
- [60] Sivic and Zisserman. 2003. Video Google: a text retrieval approach to object matching in videos. In *Proc. of ICCV*.
- [61] Brad Smith. 2020. Microsoft will be carbon negative by 2030. <https://blogs.microsoft.com/blog/2020/01/16/microsoft-will-be-carbon-negative-by-2030/>. Accessed: 2021-08-13.
- [62] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and Policy Considerations for Modern Deep Learning Research. In *Proc. of AAAI*.
- [63] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *arXiv preprint arXiv:2104.08663* (2021).
- [64] Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. 2021. Pseudo-Relevance Feedback for Multiple Representation Dense Retrieval. In *Proc. of ICTIR*.
- [65] Hans H Wellisch. 1986. The oldest printed indexes. *Indexer* 15, 2 (1986).
- [66] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proc. of SIGIR*.
- [67] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *Proc. of ICLR '21*.
- [68] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In *Proc. of CIKM*.
- [69] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Jointly Optimizing Query Encoder and Product Quantization to Improve Retrieval Performance. *arXiv preprint arXiv:2108.00644* (2021).
- [70] Justin Zobel and Alistair Moffat. 2006. Inverted files for text search engines. *Comput. Surveys* 38, 2 (2006).

## A APPENDIX: IMPACT OF COST WEIGHTING

In Section 6.1, we define a simple cost aggregation function (see Eq. 11) with three parameters that control the weight of latency, indexing time, and storage requirement costs. In Figure 3, we showcase additional weighting combinations in addition to the scenario in Figure 1. Many of the benchmarked methods have diametrical cost behaviors. While HNSW trades indexing time and storage space for lower latency, IVF does the opposite. The two methods based on T5 (DocT5Query & DeepImpact) have enormous indexing times with very low latency and storage requirements.

We find that by modulating the weighting, a decision maker would come to different conclusions, as the combined Pareto frontier is set by different methods in Figure 3. Note that these parameters are collection- and task- and domain-specific and should be carefully selected by domain experts and decision makers.

- **Figure 3 (a)** repeats the results of Figure 1 (d) for easier comparison. It models an emphasis on query latency, and equally includes indexing time and storage.
- **Figure 3 (b)** increases the emphasis on indexing time. This is important for scenarios with many updates and index refreshes. We demonstrate that in this case the IVF method shows a better tradeoff than HNSW.
- **Figure 3 (c)** sets all weights to 1, which again favors IVF over HNSW and the difference between BM25 and the neural approaches becomes stronger, requiring a greater permissible cost increase factor for a selection of dense methods.
- **Figure 3 (d)** shows a scenario of a static collection, where indexing is a dismissible (and therefore ignored) one-time cost. The only cost factors that matter are storage space and latency. Now, the neural augmented term-based methods become much more viable to select, as their main drawback is the slow inference over all passages.

This analysis again shows how we will not arrive at a single general recommendation for or against deploying a certain system. It depends strongly on the individual situation. Our framework provides the guidance for informed decision makers.

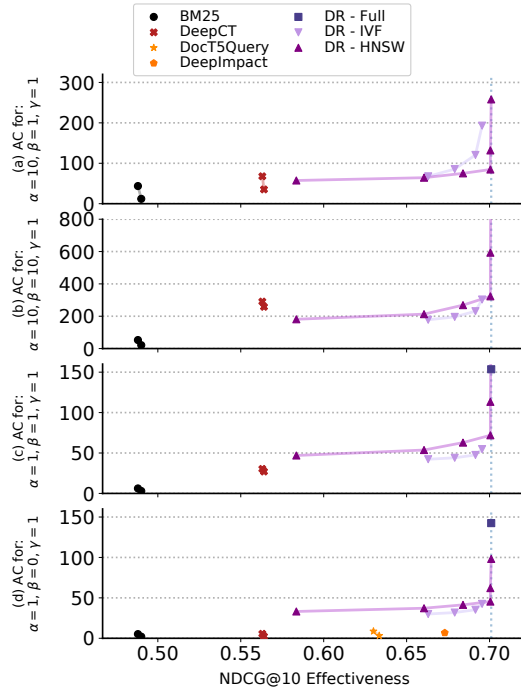


Fig. 3. Comparing different cost aggregation strategies on TREC-DL passage retrieval comparison