Enhanced Retrieval Effectiveness through Selective Query Generation

Seyed Mohammad Hosseini
mohammad.hosseini75@gmail.com
Toronto Metropolitan University
Toronto, Ontario, Canada

Negar Arabzadeh
narabzad@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Morteza Zihayat
mzihayat@torontomu.ca
Toronto Metropolitan University
Toronto, Ontario, Canada

Ebrahim Bagheri
bagheri@torontomu.ca
Toronto Metropolitan University
Toronto, Ontario, Canada

Abstract

Prior research has demonstrated that reformulation of queries can significantly enhance retrieval effectiveness. Despite notable successes in neural-based query reformulation methods, identifying optimal reformulations that cover the same information need while enhancing retrieval effectiveness is still challenging. This paper introduces a two-step query reformulation framework for generating and selecting optimal target query variants which not only achieve higher retrieval performance but also preserve the original query's information need. Our comprehensive evaluations on the MS MARCO dataset and TREC Deep Learning tracks demonstrate substantial improvements over original query's performance.

CCS Concepts

• **Information systems → Information retrieval**; **Information retrieval query processing**; **Query reformulation**.

Keywords

Query Variants, Query reformulation, Information Retrieval

ACM Reference Format:
Seyed Mohammad Hosseini, Negar Arabzadeh, Morteza Zihayat, and Ebrahim Bagheri . 2024. Enhanced Retrieval Effectiveness through Selective Query Generation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679912

1 Introduction

In information retrieval (IR), the quality of search results is often dependent on how the query reflects the user's intent [18]. In many cases, the users face difficulties in precisely articulating their information needs, which can adversely affect the retrieval performance [36]. The literature has shown that it is possible to use different query formulations to represent the same information need through both an ambiguous complex query and also, a clear concise query [23, 31]. Table 1 presents empirical evidence of the advantages of effective query reformulation. Examples from the MS MARCO passage retrieval dataset [24] demonstrate how modified queries can significantly improve retrieval outcomes, as measured by the Mean Reciprocal Rank (MRR) metric. For example, the query "What is datum target" obtained MRR@10 of 0.17. However, reformulating this query to "datum target definition" increases its effectiveness to MRR@10 of 1. For this reason, researchers have explored how methods such as query expansion and reformulation [8, 28] to modify or expand user queries to better align with the intended information need. Although these methods have demonstrated effectiveness on traditional sparse retrieval methods [35], their performance is often suboptimal on more recent neural dense retrievers, occasionally even reducing the effectiveness of the search results [4]. Empirical evidence shows that dense neural retrievers require more advanced query reformulation techniques, which would do more than merely expanding or slightly modifying the original query by thoroughly rephrasing the query to maximize retrieval effectiveness [22, 38]. To the best of our knowledge, there is no previous query reformulation approach that shows improvement over both sparse and dense neural based retrievers [32]. Motivated by these observations, the formulation of an optimal *target query* could potentially enhance the effectiveness of both sparse and dense neural information retrieval methods. The objective of the target query should be to increase retrieval effectiveness for the original user query.

One potential strategy for generating such a *target query* would be to use a pseudo-relevance feedback [9, 12] where the top documents retrieved for the original query are used as a source for generating alternative queries. Nogueira et al. [26, 27] have shown that it is possible to use a transformer architecture to learn document to query translations. Such an approach could be used for generating alternative queries for the original queries by translating the top retrieved documents into potential alternate queries. Although this method has potential, it fails to produce effective queries for at least two **R**easons: (**R1**) The transformer might generate queries that do not necessarily capture the information needs depicted by the document. This discrepancy often happens when the document captures a range of information and therefore capturing its content in a concise short query may not be possible. Therefore, a generated query for such a document would potentially not be an accurate representation of the user's needs. (**R2**) Second, the use of

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679912

Table 1: Examples of *Original* query and alternative *Target* query from MS MARCO dataset.

Query	MRR@10
Original: what is datum target	0.17
Target: datum target definition	1.00
Original: what is crime rate for new port richey fl	0.00
Target: what is the crime rate of new port richey	0.33
Original: tire wear patterns and causes	0.50
Target: what causes uneven tire wear	1.00

pseudo-relevance feedback has the potential to include marginally relevant or less relevant documents to the query at the top of the ranked list of documents depending on the difficulty of the query for the retrieval method. Given the fact that finding a target query is mostly meaningful for more difficult queries [2, 16], such queries will experience less than optimally relevant documents. Therefore, using such documents to generate alternative queries would possibly only lead to generating target queries that experience significant semantic drift from the users' information needs [39]. Therefore, there is a need for a systematic approach to generating queries that not only address the information needs specified in the original query but also lead to improvement in effectiveness.

In this paper, we introduce a two-step query reformulation framework designed to address the challenges identified in generating effective queries. First, using a fine-tuned transformer model, we generate potential alternative queries, based on pseudo-relevant documents retrieved during an initial search phase. Secondly, we employ a cross-encoder model trained to select the best target query by predicting their retrieval effectiveness. Our approach aims to consistently produce target queries that not only align with the original query's information need but also have the potential to enhance retrieval effectiveness.

2 Proposed Approach

2.1 Problem Definition

Given a query q and a collection of documents C, the retriever M is tasked with retrieving the top-k documents $D_q = [d_q^1, d_q^1, \ldots, d_q^k]$, represented as $D_q \leftarrow M(q, C)$. Each query q is accompanied by a set of judged relevant documents R_q, where each document $d \in R_q$ has been annotated as satisfying the information need behind the query q so called as I_q. The performance of an IR system is assessed using an evaluation function μ, where $\mu(q, D_q|R_q)$ quantifies the quality of the ranked list of retrieved documents. The objective of our work is to reformulate an input query into a *target query variant* that not only shares the same information need but also represents the main query in an easier-to-address form for the retrieval method. Simply put, our goal is to find an alternative query variant $\hat{q}_t$ such that $\mu(\hat{q}_t, D_{\hat{q}_t}|R_q) > \mu(q, D_q|R_q)$ where the target query variant $\hat{q}_t$ and original query q share the same information need i.e., $I_q = I_{\hat{q}_t}$ but they are represented in a different lexical form.

2.2 Query Variants Generation

To reformulate the original query into a more effective alternative target query, we first generate a set of query variants $\hat{Q}_q$ for the original query q, where both q and every $\hat{q} \in \hat{Q}_q$ share the same information need. To generate these query variants, inspired by previous work [26, 27], we adopt a document-to-query translation approach. This involves translating a document into a query representation using a transformer model trained on existing query-relevant document pairs available in the relevance judgment dataset [1, 11, 32]. The transformer model, denoted as $\mathcal{T}$, learns to generate queries for an input document based on the association between existing queries and their relevant documents. Formally, the translation can be represented as: $\hat{q}_d \leftarrow \mathcal{T}(d)$ where $\mathcal{T}$ is a non-deterministic function, i.e., applying $\mathcal{T}$ to document d several times would yield different $\hat{q}_d$. As such, $\mathcal{T}^m$ represents m repeated applications of $\mathcal{T}$. The details of the transformer architecture are provided in the experimental setup section of this paper. Given this transformer, it would now be possible to generate queries for each document.

We apply the document-to-query translation function $\mathcal{T}$ to every document that could potentially answer the initial query q. The hypothesis is that documents answering the query q will likely share a common information need. We use the top-retrieved documents for query q in the initial round of retrieval and apply $\mathcal{T}$ on it. Applying $\mathcal{T}$ on documents $d_i \in D_q^k$, results in $\hat{Q}_q^{k,m}$ where it represents m repeats of applying $\mathcal{T}$ on the top-k retrieved documents for q :

$$\hat{Q}_q^{k,m} = \{\mathcal{T}^m(d) \mid d \in D_q^k\} \tag{1}$$

2.3 Query Variant Classification

The generated $\hat{Q}_q$ queries may suffer from being noisy due to several reasons (R1 and R2 in the Introduction):
1) Not all the top-retrieved documents D_q necessarily contain relevant information with respect to q. As such, they might not address the information need behind the query q, and consequently, they cannot generate a target query variant $\hat{q}_t$ that satisfies $I_q = I_{\hat{q}_t}$.
2) The transformation function $\mathcal{T}$ is not fully error-free. Therefore, since $\mathcal{T}$ is a noisy process, not every query generated from $\mathcal{T}$ necessarily represents an alternative version $\hat{q}$ of the original query q that leads to higher performance, i.e., $\mu(\hat{q}, D_{\hat{q}}|R_q) > \mu(q, D_q|R_q)$.

Due to these reasons, not all query variants $\hat{q} \in \hat{Q}_q$ satisfy our two *target query* conditions. To identify appropriate target queries that meet our desirable criteria, we train a *query variant classifier* ϕ aiming to identify target queries from $\hat{Q}_q$ that share the same information need with the original query while showing higher effectiveness. The labels for classifier $\phi(q, \hat{q})$ are defined as:

$$\phi(q, \hat{q}) = \begin{cases} 1 & \text{if } \mu(\hat{q}, D_{\hat{q}}|R_q) > \mu(q, D_q|R_q) \wedge I_q = I_{\hat{q}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We apply $\phi(\cdot)$ to query variants and original query to obtain the target query set $\mathbb{Q}_q^t$ by contrasting the original query and its retrieved documents with a query variant and its retrieved documents. Our goal is to identify a set of target query variants $\mathbb{Q}_q^t$ such that every query $q_i \in \mathbb{Q}_q^t$ satisfies the two target query conditions mentioned earlier. Additionally, we note that $\mathbb{Q}_q^t \subseteq \hat{Q}_q$. We train the query variant classifier through contrastive learning via a cross-encoder network [25] where positive and negative labels are obtained as in Equation 2. Positive labels indicate those variants that lead to better performance than the original query, while negative labels indicate those that show worse performance than the original query. To learn the target query predictor ϕ, as suggested in previous works and inspired by the success of contrastive learning [19, 34, 37], we leverage the comparison of information obtained from the original query and its query variant. In the cross-encoder

Table 2: Performance of our proposed approach as well as the original retrievers on TREC DL-19 and DL-20 datasets.

Architecture	DL-19				DL-20			
	Original		Ours		Original		Ours	
	mAP@1k	NDCG@10	mAP@1k	NDCG@10	mAP@1k	NDCG@10	mAP@1k	NDCG@10
Sparse	0.3714	0.4936	0.3457	**0.5206**	0.3414	0.4826	**0.356**	**0.5053**
DE-NN	0.3606	0.6481	**0.3898**	**0.6737**	0.3907	0.6458	0.3777	**0.6487**
DE-LI	0.3864	0.67	**0.4174**	**0.6916**	0.4059	0.6678	**0.4183**	**0.6769**
DE	0.3479	0.6368	**0.4411**	**0.6543**	0.3759	0.6565	**0.4316**	**0.6882**

architecture, where the token interactions from the information of the original query interact with information pieces from the variant, we concatenate the query variant $\hat{q}_d$ with its first retrieved document, as well as their relevance score $S(\hat{q}_d, d)$ which represents the association between the query and the document. Adding the associated scores between the inputs has previously been shown to lead to improvements in classification tasks [7]. We feed the cross-encoder network $\mathcal{F}$ with the same information from the original query. As such, the input to the ϕ function for a pair of query q and a query variant $\hat{q}$ would be:

$$\hat{\phi}(q, \hat{q}) = \mathcal{F}(q \oplus d_q^1 \oplus S(q, d_q^1), \hat{q} \oplus d_{\hat{q}}^1 \oplus S(\hat{q}, d_{\hat{q}}^1)) \qquad (3)$$

where $\oplus$ is a concatenation operator followed by special [SEP] token. In network $\mathcal{F}$, we apply a linear layer on the first vector produced by the transformer [CLS], to produce a scalar value $\hat{\phi}(q, \hat{q})$. We leverage a sigmoid layer and a Binary Cross Entropy loss function to train $\mathcal{F}$ where $N = |\hat{Q}|$ is the total number of pairs of original query and query variants for training purposes:

$$L = -\frac{1}{N}{}_{i=1}^{N}\Big(\phi(q_i, \hat{q}_i)\log(\hat{\phi}(q_i, \hat{q}_i)) + (1 - \phi(q_i, \hat{q}_i))\log(1 - \hat{\phi}(q_i, \hat{q}_i))\Big) \qquad (4)$$

3 Experiments

3.1 Datasets and Retrievers

Datasets. We use the MS MARCO passage collection V1 [24], which contains over 8.8 million passages. We report the performance on MS MARCO Dev set which includes 6,980 queries, each annotated on average with 1.06 relevant documents. We further conduct experiments on the TREC Deep Learning Track 2019 and 2020 datasets [13, 14]. The former contains 43 queries, while the latter query set includes 54 queries. On average, they have 215 and 210 annotated documents per query, respectively. The main difference between these query sets and MS MARCO Dev set is that while MS MARCO Dev set has been sparsely and binary labelled, these query sets have been judged more comprehensively [5]. The official evaluation metric for TREC DL-19 and DL-20 is nDCG@10 and MRR@10 for MS MARCO Dev set, which we adopt in our work. We additionally report the effectiveness on a deeper rank and report Mean Average Precision at 1000.

Retrievers. We evaluate the robustness of our proposed approach across different retriever types, including both sparse and dense models. We selected BM25, a widely used traditional bag-of-words high-dimensional sparse retriever, as representative of this category [20]. For dense retrievers, we include the following methods:

Dual-encoder-based retrievers: Bi-encoders or dense retrievers have become notable for their ability to learn dense, contextualized representations of queries and documents [6]. We utilized the widely adopted Dual-Encoder (DE) architecture from the Sentence Transformers library, which employs a Siamese network structure to generate semantically meaningful sentence embeddings that can be compared using cosine similarity [30].

Table 3: Comparison of hard queries in MS MARCO Dev set, categorized into 10 difficulty-based buckets.

Model	Training	0-10%	10-20%	20-30%	30-40%	40-50%
Sparse	Original	0	0	0	0	0
	Ours	0.018†	0.018†	0.018†	0.018†	0.013†
DE-NN	Original	0	0	0	0	0.104
	Ours	0.026†	0.026†	0.026†	0.026†	0.169†
DE-LI	Original	0	0	0	0	0.119
	Ours	0.018†	0.024†	0.022†	0.026	0.191†
DE	Original	0	0	0	0.022	0.078
	Ours	0.018†	0.022†	0.032†	0.032†	0.134†

DE-nearest neighbor negatives: We also incorporate ANCE (Approximate Nearest Neighbor Negative Contrastive Estimation) [37] which enhances training by incorporating hard negative mining, selecting negatives closely related to the query in the embedding space. This method is referred to as Dual-Encoder with Nearest Neighbor Negatives (DE-NN).

DE-late interaction: We also evaluated our query variant selection strategy using the ColBERT (Contextualized Late Interaction over BERT) model [21]. ColBERT allows for a more granular comparison between query and document tokens by facilitating late interaction between their representations. Consequently, we refer to this retriever as Dual-Encoder with Late Interaction (DE-LI).

3.2 Experimental Setup

Transformer Model. To generate alternative queries using transformer models, we utilize the docT5query [26] which fine-tunes T5 [29] to generate queries from a given passage.

Training Set. For each query, we generate 100 alternative queries by setting $k = 10$ to retrieve top-10 most relevant documents and $m = 10$ to produce 10 variations of alternative query. As mentioned earlier, the transformer function $\mathcal{T}$ is noisy and might assess perturbed query alternatives. To reduce this noise, we assess the similarity score between the main query and each transformer-generated alternative query. We filter out query pairs that have a similarity below a certain threshold e.g., less than 0.1. Furthermore, to mitigate potential training bias, we balance the dataset based on the distribution of labels across the queries.

Classification Model and Hyperparameters. We fine-tune the BERT-based-uncased [15] through a cross-encoder and the Sentence Transformers [30] Library with learning rate of 2e-5 and a warm-up phase comprising 10% of the total training steps. Training proceeds for one epoch with a batch size of 16, and the CrossEntropyLoss function is employed to optimize the model.

Table 4: Improvements on hard queries of the MS MARCO Dev set.

Architecture	Number of Queries	MRR@10
Sparse	111 out of 4,224	0.165
DE-NN	321 out of 2,908	0.261
DE-LI	293 out of 2,834	0.217
DE	319 out of 3,011	0.239

3.3 Results

Query Sets with Deep Labels. By comparing the performance of different models in Table 2, we observe that: **(1)** Our approach consistently improved performance across different retrievers types on both DL-19 and DL-20 datasets in terms of nDCG@10. **(2)** For the recall-oriented metric i.e., MAP@1k, our method outperforms the original models in all variations. However, with BM25 on DL-19, the original model scored higher, though this difference was not statistically significant (paired t-test, p-value < 0.05).

Query Sets with Sparse Labels. To better understand how our approach performs on datasets with sparse relevance judgments, we use the MS MARCO Dev set. Unlike DL-19 and DL-20, this set features fewer relevance documents per query. Previous research indicates that at least 40% of the queries in this set are not effectively addressed by any existing retrieval method, i.e., effectiveness of zero for these queries [4, 10]. To evaluate our approach on datasets with challenging queries, we assess whether our query variant classifier effectively aids them. We define difficult queries as those that exhibit poor performance with a given ranking method [3, 17, 33]. Specifically, we categorize the most difficult queries for a ranker as those that rank in the lower half of retrieval effectiveness compared to other queries. Therefore, to identify difficult queries, we rank the queries in the MS MARCO Dev set by their MRR@10, the official metric for this dataset, and select the bottom 50% as difficult queries. This selection process results in 3,490 queries from a total of 6,980. We further divide these difficult queries into five finer-grained difficulty buckets to analyze the performance of baseline rankers and our proposed method within each bucket.

Table 3 presents the performance results across various difficulty buckets. Notably, the bottom 3 buckets, which comprise the lowest 30% of queries, show a reciprocal rank value of 0 for all neural rankers, indicating their inability to retrieve a relevant document within the top-10 ranked documents for roughly 2,000 queries. In contrast, our approach has managed to effectively address some of these hard queries in each difficulty bucket. Although the MRR@10 scores achieved by our method are low, it is important to consider that no neural baseline in our study could address any queries in this group, with all achieving an absolute MRR@10 score of zero. These findings underscore the capability of our method to handle difficult queries more effectively than existing neural rankers.

Furthermore, Table 4 details the number of queries that our method improved from a baseline performance of zero to higher retrieval effectiveness. For instance, using the DE-NN, our method managed to address at least partially 321 out of 2,908 queries that initially had zero retrieval effectiveness. The third column shows the average retrieval effectiveness (MRR@10) for these improved queries, which, for example, is 0.261 for the 321 DE-NN queries.

Table 5: Comparison of different ranking models' performance (MRR@10) on MS MARCO Dev set.

Architecture	Original	Oracle
Sparse	0.184	0.409
DE-NN	0.322	0.638
DE-LI	0.327	0.634
DE	0.298	0.598

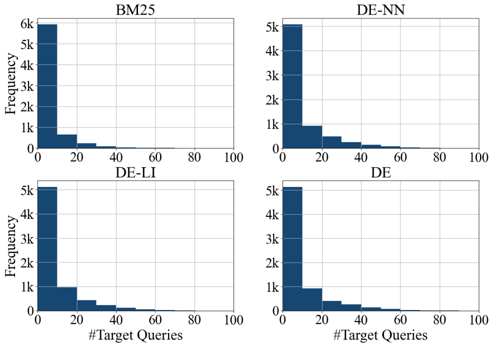

Figure 1: Oracle Distribution for each methodology.

Oracle Analysis. We also analyze the frequency of target queries in the MS MARCO Dev set. As mentioned earlier, we display the frequency of the number of target queries available when building $\hat{Q}_q^{k,m}$ with k and m set to 10, i.e., when $|\hat{Q}_q| = 100$ across 6,980 queries in the MS MARCO Dev set. Specifically, we investigate how many of these queries satisfy our dual criteria of demonstrating higher retrieval effectiveness while preserving the same information need. The results are displayed in Figure 1. Irrespective of the retriever type, approximately 4,000 of the 6,980 queries, or 57%, have at least one target query where $\phi(q, \hat{q}) = 1$. This demonstrates two key points: **(1)** A significant number of target queries are present within our query variant set, highlighting the efficacy of our query generation process. **(2)** Despite the presence of these target queries, identifying and classifying them remains challenging due to the relatively low number of target queries across the query variants, emphasizing the complexity of the task.

Moreover, in Table 5, we report the performance of the original retrievers as well as the oracle performance in terms of MRR@10 on the MS MARCO Dev set. When $\hat{Q}_q = \varnothing$, the performance of the original query is reported as the oracle. The oracle column essentially demonstrates the *'potential'* of our proposed methodology under ideal conditions with $\phi(q, \hat{q})$. The oracle column indicates substantial potential for improvement across different retrievers if the target query variant ϕ is accurately selected. For instance, the original BM25 performance of 0.184 could potentially rise to 0.409. Similarly, for dual encoders like ColBERT, performance could improve from 0.327 to 0.634. This confirms the significant enhancement opportunities available with a reliable and effective target query variant identifier ϕ.

4 Concluding Remarks

This study introduced a robust two-step query reformulation framework that utilizes a fine-tuned transformer model and a cross-encoder classifier to effectively generate queries that align with the original query's intent while enhancing retrieval performance. Our extensive evaluations, conducted on the MS MARCO dataset and TREC Deep Learning tracks, confirm that our approach significantly outperforms existing baselines.

References

[1] Negar Arabzadeh, Amin Bigdeli, Shirin Seyedsalehi, Morteza Zihayat, and Ebrahim Bagheri. 2021. Matches made in heaven: Toolkit and large-scale datasets for supervised query reformulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4417–4425.

[2] Negar Arabzadeh, Radin Hamidi Rad, Maryam Khodabakhsh, and Ebrahim Bagheri. 2023. Noisy perturbations for estimating query difficulty in dense retrievers. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 3722–3727.

[3] Negar Arabzadeh, Chuan Meng, Mohammad Aliannejadi, and Ebrahim Bagheri. 2024. Query Performance Prediction: From Fundamentals to Advanced Techniques. In *European Conference on Information Retrieval*. Springer, 381–388.

[4] Negar Arabzadeh, Bhaskar Mitra, and Ebrahim Bagheri. 2021. Ms marco chameleons: challenging the ms marco leaderboard with extremely obstinate queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4426–4435.

[5] Negar Arabzadeh, Alexandra Vtyurina, Xinyi Yan, and Charles LA Clarke. 2022. Shallow pooling for sparse labels. *Information Retrieval Journal* 25, 4 (2022), 365–385.

[6] Negar Arabzadeh, Xinyi Yan, and Charles LA Clarke. 2021. Predicting efficiency/effectiveness trade-offs for dense vs. sparse retrieval strategy selection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2862–2866.

[7] Arian Askari, Amin Abolghasemi, Gabriella Pasi, Wessel Kraaij, and Suzan Verberne. 2023. Injecting the BM25 score as text improves BERT-based re-rankers. In *European Conference on Information Retrieval*. Springer, 66–83.

[8] Hiteshwar Kumar Azad and Akshay Deepak. 2019. Query expansion techniques for information retrieval: a survey. *Information Processing & Management* 56, 5 (2019), 1698–1735.

[9] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.

[10] Amin Bigdeli, Negar Arabzadeh, and Ebrahim Bagheri. 2024. Learning to Jointly Transform and Rank Difficult Queries. In *European Conference on Information Retrieval*. Springer, 40–48.

[11] Amin Bigdeli, Negar Arabzadeh, Shirin Seyedsalehi, Bhaskar Mitra, Morteza Zihayat, and Ebrahim Bagheri. 2023. De-biasing relevance judgements for fair ranking. In *European Conference on Information Retrieval*. Springer, 350–358.

[12] Stefan Buttcher, Charles LA Clarke, and Gordon V Cormack. 2016. *Information retrieval: Implementing and evaluating search engines*. Mit Press.

[13] 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

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

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[16] Sajad Ebrahimi, Maryam Khodabakhsh, Negar Arabzadeh, and Ebrahim Bagheri. 2024. Estimating Query Performance Through Rich Contextualized Query Representations. In *European Conference on Information Retrieval*. Springer, 49–58.

[17] Faezeh Ensan and Ebrahim Bagheri. 2017. Document retrieval model through semantic linking. In *Proceedings of the tenth ACM international conference on web search and data mining*. 181–190.

[18] Ahmed Hassan, Xiaolin Shi, Nick Craswell, and Bill Ramsey. 2013. Beyond clicks: query reformulation as a predictor of search satisfaction. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2019–2028.

[19] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118* (2021).

[20] K Sparck Jones, Steve Walker, and Stephen E. Robertson. 2000. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *IPM* (2000).

[21] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. *CoRR* abs/2004.12832 (2020). arXiv:2004.12832 https://arxiv.org/abs/2004.12832

[22] Hang Li, Shuai Wang, Shengyao Zhuang, Ahmed Mourad, Xueguang Ma, Jimmy Lin, and Guido Zuccon. 2022. To interpolate or not to interpolate: Prf, dense and sparse retrievers. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2495–2500.

[23] Christina Lioma and Iadh Ounis. 2008. A syntactically-based query reformulation technique for information retrieval. *Information processing & management* 44, 1 (2008), 143–162.

[24] 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. (2016).

[25] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[26] Rodrigo Nogueira, Jimmy Lin, and AI Epistemic. 2019. From doc2query to docTTTTTquery. *Online preprint* 6, 2 (2019).

[27] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. 2019. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375* (2019).

[28] Jessie Ooi, Xiuqin Ma, Hongwu Qin, and Siau Chuin Liew. 2015. A survey of query expansion, query suggestion and query refinement techniques. In *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*. IEEE, 112–117.

[29] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR* abs/1910.10683 (2019). arXiv:1910.10683 http://arxiv.org/abs/1910.10683

[30] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[31] Soo Young Rieh et al. 2006. Analysis of multiple query reformulations on the web: The interactive information retrieval context. *Information Processing & Management* 42, 3 (2006), 751–768.

[32] Sara Salamat, Negar Arabzadeh, Fattane Zarrinkalam, Morteza Zihayat, and Ebrahim Bagheri. 2023. Learning query-space document representations for high-recall retrieval. In *European Conference on Information Retrieval*. Springer, 599–607.

[33] Abbas Saleminezhad, Negar Arabzadeh, Soosan Beheshti, and Ebrahim Bagheri. 2024. Context-Aware Query Term Difficulty Estimation for Performance Prediction. In *European Conference on Information Retrieval*. Springer, 30–39.

[34] Jie Shao, Xin Wen, Bingchen Zhao, and Xiangyang Xue. 2021. Temporal context aggregation for video retrieval with contrastive learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 3268–3278.

[35] Mahtab Tamannaee, Hossein Fani, Fattane Zarrinkalam, Jamil Samouh, Samad Paydar, and Ebrahim Bagheri. 2020. Reque: a configurable workflow and dataset collection for query refinement. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 3165–3172.

[36] Jaime Teevan, Eytan Adar, Rosie Jones, and Michael AS Potts. 2007. Information re-retrieval: Repeat queries in Yahoo's logs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 151–158.

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

[38] HongChien Yu, Chenyan Xiong, and Jamie Callan. 2021. Improving query representations for dense retrieval with pseudo relevance feedback. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3592–3596.

[39] Liron Zighelnic and Oren Kurland. 2008. Query-drift prevention for robust query expansion. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 825–826.