Learning Robust Dense Retrieval Models from Incomplete Relevance Labels

Prafull Prakash, Julian Killingback, and Hamed Zamani
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
{prafullpraka,jkillingback,zamani}@cs.umass.edu

ABSTRACT

Recent deployment of efficient billion-scale approximate nearest neighbor (ANN) search algorithms on GPUs has motivated information retrieval researchers to develop neural ranking models that learn low-dimensional dense representations for queries and documents and use ANN search for retrieval. However, optimizing these dense retrieval models poses several challenges including negative sampling for (pair-wise) training. A recent model, called ANCE, successfully uses dynamic negative sampling using ANN search. This paper improves upon ANCE by proposing a robust negative sampling strategy for scenarios where the training data lacks complete relevance annotations. This is of particular importance as obtaining large-scale training data with complete relevance judgment is extremely expensive. Our model uses a small validation set with complete relevance judgments to accurately estimate a negative sampling distribution for dense retrieval models. We also explore model penalization for making "easy-to-avoid" mistakes using a lexical matching signal and pseudo-relevance feedback during evaluation. Our experiments of the TREC Deep Learning Track benchmarks demonstrate the effectiveness of our solutions.[1]

ACM Reference Format:
Prafull Prakash, Julian Killingback, and Hamed Zamani. 2021. Learning Robust Dense Retrieval Models from Incomplete Relevance Labels. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463106

1 INTRODUCTION

Development of various learning to rank models in the past two decades and neural ranking models in recent years have led to significant improvements in retrieval effectiveness for a wide range of information retrieval tasks. For efficiency reasons, search engines adopt a multi-stage cascaded architecture and use learning to rank models in late stages. Therefore, these models only re-rank a small set of documents retrieved by early stage models. There are major drawbacks in this design. For instance, multi-stage cascaded architectures suffer from an error propagation problem and their end-to-end optimization is difficult or even impractical.

To address these issues, Zamani et al. [21] revisited the foundations of learning to rank models and proposed the first *standalone* neural ranking model, called SNRM, that is capable of retrieving documents from large-scale collections. SNRM learns extremely high-dimensional sparse representations for queries and documents and constructs an inverted index based on the learned document representations for efficient retrieval. However, not all of the state-of-the-art neural network architectures support sparse operations. Since converting sparse tensors to regular (dense) tensors requires massive GPU memories, this approach becomes impractical.

Recent development of efficient approximate nearest neighbor (ANN) search algorithms on GPUs, e.g., [8], has provided an alternative solution for efficient standalone retrieval over billion-scale collections. This has recently motivated information retrieval researchers to learn dense query and document representations using neural models and conduct efficient retrieval using ANN search over the learned dense representations [3, 9, 11, 18]. This is often called *dense retrieval* and is the focus of this paper.

A challenging part of developing standalone neural retrieval models is their optimization and most importantly negative sampling for (pair-wise) training. In more detail, taking negative samples from the output of an existing retrieval model, e.g., BM25, which is the common practice for training re-ranking models, is sub-optimal for training standalone ranking models. Recently, Xiong et al. [18] proposed an effective negative sampling strategy for dense retrieval, called ANCE, that uses ANN search based on the representations produced by the model being trained and *uniformly* takes "hard negative samples" based on the top retrieved documents. However, most large-scale data for training neural models (e.g., click data or MS MARCO) suffer from lack of complete relevance annotations and such sampling increases the chance of unjudged relevant documents being selected as negative instances. Figure 1 plots the number of relevant documents at each rank in the result lists produced by the ANCE model on the TREC 2019 Deep Learning Track data (a data with relatively complete relevance judgments). The graph shows that a large number of top retrieved documents are actually relevant to the query and in case of incomplete relevance annotations (e.g., on the MS MARCO dataset), the ANCE negative sampling method would actually sample a large number of unjudged relevant documents as negative.

We address this issue by estimating a negative sampling distribution on the fly. The sampling distribution is estimated based on the model's performance on a small validation set with relatively complete relevance annotations. It also uses a discounting factor that discourages the model to take "easy negative samples".

[1] Our code and trained models are available at https://github.com/purble/RANCE.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463106

Our experiments also suggest that dense retrieval can be further improved by using pseudo-relevance feedback during evaluation and taking advantage of a lexical matching signal, i.e., BM25, for penalizing the model when making "easy to avoid" mistakes. The latter is achieved by dynamic adjustment of loss margin using the lexical matching signal. Our experiments on both passage and document retrieval tasks offered by the TREC Deep Learning Track demonstrate the effectiveness of the proposed solutions.

2 RANCE

2.1 Background

Various neural ranking models have been proposed for document and passage retrieval tasks [5]. Due to efficiency reservations, most of these models, e.g., [2, 4, 6, 14], only re-rank a small number of documents retrieved by an early stage retrieval model, such as BM25 [16] or query likelihood [15]. As mentioned in Section 1, dense retrieval is an efficient and effective standalone retrieval solution that has recently attracted considerable attention in the IR literature [3, 9, 11, 18]. Dense retrieval models learn relatively low-dimensional dense representations for queries and documents and then employ ANN search algorithms, such as Faiss [8] that uses product quantization for efficient billion-scale retrieval. The recent ANCE model [18], which provides the basis for this work, is one of these effective dense retrieval methods. It uses RoBERTa [13] for query and document representations and inner product for computing their similarity. The power of ANCE lies on the dynamic negative sampling solution they developed, which samples negative documents for pair-wise training from the documents retrieved by the model being trained. In fact, an ANN search is used for taking negative samples whilst training. This negative sampling strategy has shown substantial improvements over strong baselines, including negative sampling from the documents retrieved by BM25 which is a common practice in training learning to rank models.

In the following subsections, we describe our improvement over the ANCE model by proposing more accurate negative sampling and training techniques. We use the same neural network architecture and pre-training as in ANCE.

2.2 A Robust Approximate Negative Sampling Approach

Various objective functions have been proposed for optimizing learning to rank models, including point-wise, pair-wise, and list-wise objectives [12]. Due to the nature of retrieval tasks, each training query is associated with a small number of relevant documents and countless non-relevant documents. Therefore, regardless of the training objective, optimizing retrieval and ranking models requires negative sampling. Negative documents in learning to rank models are typically sampled from the top retrieved documents returned by an early stage retrieval model (M_1), such as BM25 [16] or query likelihood [15]. Such an approach works well when the model solely re-ranks a number of documents returned by M_1. However, Xiong et al. [18] recently showed that it would be sub-optimal for dense retrieval models that retrieve documents from a large collection. For every training query, ANCE uses an ANN search algorithm based on the model's representations and randomly sample from the top 200 documents returned by ANN search. The intuition is to force

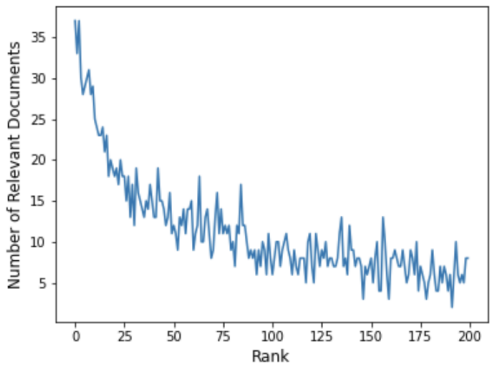

Figure 1: Number of relevant documents at each rank for the TREC 2019 Deep Learning Track query set (document ranking task) retrieved by ANCE [18].

the model to distinguish between positive documents and "hard" negative documents drawn from the model's mistake. This negative sampling solution has led to significant performance improvement in dense retrieval.

When training on data with incomplete relevance annotations such as MS MARCO or even click data, it is likely that ANCE selects negative samples which are actually relevant to the query but have not been annotated. Figure 1 highlights this fact. This is one of the major shortcomings of ANCE which may mislead the ranking model. We propose RANCE which improves the robustness of the ANCE negative sampling approach for data with incomplete relevance annotations. Instead of uniformly sampling negative instances from the top retrieved documents, we estimate a more accurate sampling distribution from a small validation dataset with complete relevance judgments.

Let $Q = \{q_1, q_2, \cdots, q_n\}$ and $Q' = \{q'_1, q'_2, \cdots, q'_m\}$ respectively denote a large-scale training query set and a small scale validation query set ($m \ll n$). Let the sets R and R' contain relevance annotations for the query sets Q and Q', respectively. This means that the judged relevant documents for every query in Q / Q' are included in R / R'. Note that R consists of incomplete relevance annotation (e.g., MS MARCO) and R' includes relatively complete annotations (e.g., the TREC DL Track data).

For every training query $q \in Q$, RANCE runs ANN search over the document collection C and takes the top N documents ($N = 200$ in our experiments). We aim at sampling documents that are likely to be non-relevant and are difficult for the model to distinguish from the relevant documents. To satisfy both of these constraints, RANCE takes negative samples from the top N retrieved documents using the following distribution:

$$P_{\text{sampling}}(r) = \frac{1}{Z}\left(1 - P_{\text{relevant}}(r)\right)\text{discount}(r) \tag{1}$$

where Z is a normalization factor, $P_{\text{relevant}}(r)$ denotes the probability of a document in rank r of the retrieval list being relevant, and $\text{discount}(r)$ denotes a discounting factor giving higher weights to the documents in the higher ranks. This discounting function discourages the model from sampling low ranked documents, as they are not likely to be "difficult negative documents" for the model.

RANCE estimates the probability $P_{\text{relevant}}(r)$ using the small query set Q' with complete relevance annotation. Therefore, if L'_q

is the result list returned by an ANN search for a query $q \in Q'$, $P_{\text{relevant}}(r)$ is estimated as:

$$P_{\text{relevant}}(r) = \frac{\sum_{q \in Q'} \mathbb{1}\{L'_q(r) \in R'(q)\}}{|Q'|} \quad (2)$$

where the numeration counts the number of queries whose r^{th} retrieved document is relevant. We empirically found that the reverse logarithm function is an effective discounting function. This function has been also used for discounting ranks in computing NDCG [7]. Therefore, $\text{discount}(r) = \frac{1}{\log(r+1)}$. The normalization factor Z cancels the effect of the logarithm base, thus it does not impact the results. In our experiments, we plot this distribution to further shed light into this negative sampling distribution.

Once the distribution P_{sampling} is estimated, we smooth the distribution by running a window of size $K = 9$ over the distribution and replacing the middle probability by averaging all probabilities in the window. This reduces irregular fluctuations in the estimated distribution. We further fit a polynomial distribution with a degree of 4 to the estimated probability.

2.3 PRF for Evaluation

Pseudo-Relevance feedback (PRF) methods, such as the Rocchio's algorithm [17], assume that the top retrieved documents in response to a query are relevant to the query. They often use these documents for query expansion [10, 19, 22]. Zamani et al. [21] used these techniques for updating latent query representations. Previous work used PRF at query time mainly for resolving the vocabulary mismatch problem. We use PRF for improving the quality of query representation when performing evaluation. In more detail, for each test query, we use the Rocchio's algorithm to update the query representation, i.e., $\vec{q^*} = \vec{q} + \alpha \sum_{i=1}^{k} \vec{L_q}(i)$, where $\vec{q}$ denotes the query representation and $\vec{L_q}(i)$ denotes the i^{th} retrieved document for the query q. k and α are hyper-parameters controlling the number and the impact of pseudo-relevant documents, respectively. We then use the new query vector q^* to retrieve documents for evaluation.

2.4 Training with Dynamic Error Margin

To train our model, we adapt the residual learning approach introduced by Gao et al. [3]. This approach uses a lexical matching signal to adjust the margin in the Hinge loss function. The intuition behind this approach is related to weak supervision where a term matching retrieval model, such as BM25 [16], can be used for training neural ranking models. This concept has been firstly introduced by Dehghani et al. [2] and its theoretical justification has been later discovered by Zamani and Croft [20]. Unlike in weak supervision, our training uses the weak signal to penalize the model once it makes "easy-to-avoid" mistakes, such as term matching. In more detail, for a pair-wise training instance (q, d_1, d_2), the loss function is defined as:

$$\mathcal{L} = \max\{0, m - y\,[M(q, d_1) - M(q, d_2)]\} \quad (3)$$

where $y \in \{-1, 1\}$ is the relevance label and $M(\cdot, \cdot)$ denotes the retrieval score for the given query-document pair. m is a margin that is computed as:

$$m = \epsilon - \lambda y \left[\hat{M}(q, d_1) - \hat{M}(q, d_2)\right] \quad (4)$$

where ϵ and λ are constant hyper-parameters and $\hat{M}$ is a weak signal (i.e., BM25). Intuitively, this margin over-penalizes the model when it makes mistakes where BM25 does not. At query time, we interpolate the M and $\hat{M}$ scores linearly with the coefficient of λ'.

3 EXPERIMENTS

3.1 Data

We use the TREC 2019 Deep Learning Track dataset [1] for evaluating our models. In more detail, we use the provided MS MARCO dataset that suffers from incomplete relevance annotations for training. We used two-fold cross-validation accross the queries annotated by TREC assessors (with relatively complete annotations) for validation and evaluation. The validation part is used for hyper-parameter tuning and negative sampling distribution estimation. The data consists of both passage and document retrieval collections. The collection contains 3.2 million documents and 8.8 million passages. For the document set there are some 367 thousand training queries and 3.2 million documents. The passage set contains 8.8 million passages and 503 thousand training queries. Each query has at least one passage or document that is a known positive. Although each query has at least one relevant document or passage there is no guarantee that other passages or documents are not relevant. 43 queries are annotated by TREC assessors.

3.2 Experimental Setup

For experiments, we built upon the code and trained checkpoints made available publicly accompanying ANCE [18]. We re-used most of the hyper-parameter settings as prescribed, except reduced the learning rate down to 2e-6 for document retrieval and 8e-7 for passage retrieval for further fine-tuning of models. For document retrieval, we experimented with the MaxP setting.

For PRF, we employed a grid search over the set {5, 10, 15, 20} to find the best value for k, and over {0.1, 0.2, .., 2.0} for α. During training, this hyper-parameter search is done at each ANN-index generation step. For the residual learning approach, we used the same hyper-parameter settings as suggested in [3].

3.3 Evaluation Metrics

We use the following precision- and recall-oriented evaluation metrics in our experiments: (1) Normalised Discounted Cumulative Gain [7] for the top 10 retrieved documents (NDCG), (2) Mean Reciprocal Rank for the top 10 retrieved documents (MRR), and (3) Recall for the top 100 documents. We use two-tailed paired t-test for identifying statistically significant performance differences.

3.4 Results and Discussion

We use BM25 and ANCE as baselines in our experiments. BM25 runs are conducted by the TREC Deep Learning Track organizers. For ANCE, we include the reported performance by the authors [18] and the performance obtained by our own ANCE implementation. Although many neural ranking models exist for document and passage retrieval, we do not report their performance, because: (1) ANCE outperforms the majority of existing neural ranking models and is a strong baseline, and (2) our goal is to show that our robust negative sampling strategy outperforms the one used by ANCE.

Table 1: The re-ranking and retrieval performance on the TREC 2019 Deep Learning Track data for both document and passage ranking tasks, in terms of NDCG@10, MRR@10, and Recall@100. Superscripts † and ‡ denote statistical significant improvements over ANCE (ours) with $p\_value < 0.1$ and $p\_value < 0.05$, respectively.

Model	Re-ranking						Retrieval					
	TREC DL Document			TREC DL Passage			TREC DL Document			TREC DL Passage		
	NDCG	MRR	Recall	NDCG	MRR	Recall	NDCG	MRR	Recall	NDCG	MRR	Recall
BM25	0.519	0.805	–	0.506	0.704	–	0.519	0.805	–	0.506	0.704	–
ANCE [18]	0.671	–	–	0.677	–	–	0.628	–	–	0.648	–	–
ANCE (ours)	0.671	0.913	0.312	0.675	0.963	0.676	0.635	0.909	0.301	0.653	0.936	0.667
RANCE	$0.702^{\ddagger}$	0.908	$0.325^{\dagger}$	$0.702^{\ddagger}$	0.954	0.676	$0.679^{\ddagger}$	0.908	$0.314^{\dagger}$	$0.695^{\ddagger}$	0.939	$0.697^{\ddagger}$

Table 2: Ablation study results on the TREC 2019 Deep Learning Track data for document ranking, in terms of NDCG@10, MRR@10, and Recall@100.

Model	Re-ranking			Retrieval		
	NDCG	MRR	Recall	NDCG	MRR	Recall
RANCE	0.702	0.908	0.325	0.679	0.908	0.314
RANCE- PRF	0.685	0.910	0.315	0.658	0.895	0.297
RANCE- PRF - DEM	0.680	0.920	0.315	0.642	0.917	0.294

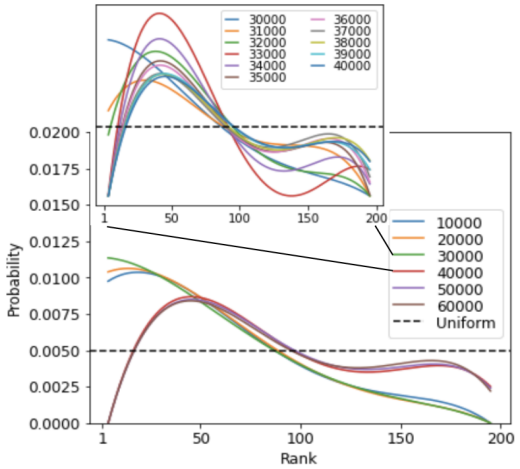

Figure 2: Sampling probability distributions produced by RANCE for every 10,000 steps of training.

The results are reported in Table 1. Our implementation of ANCE performs on par with the reported numbers for re-ranking tasks and performs better than the original implementation for retrieval tasks. The proposed RANCE model consistently improves the ANCE model in terms of NDCG and Recall. Note that due to the graded relevance labels in the data, NDCG better reflects the model's precision compared to MRR. The NDCG imrovements are statistically significant in all cases. The recall improvements are generally higher in retrieval tasks, compared to re-ranking, which shows the ability of dense retrieval models to retrieve relevant documents that cannot be found using BM25 (the first stage model in re-ranking). The recall improvement is significant for the passage retrieval task.

Ablation Study. To demonstrate the impact of different components of the proposed solution, we ran ablation study experiments by turning off the proposed components one at a time. For the sake of space, we only ran this ablation study on the document ranking task. The results are reported in Table 2. According to the results, by disabling PRF during evaluation, we observe performance drops in terms of NDCG and Recall for both re-ranking and retrieval settings. And by further disabling the dynamic error margin (DEM) approach in the loss function, we observe a further performance drop in terms of NDCG and Recall, only for the retrieval setting. This suggests that the dynamic error margin is important for retrieval. The reason is that in re-ranking settings, the candidate documents already have high term matching scores, therefore having a term matching signal during training for re-ranking tasks is not beneficial. However, the performance difference in retrieval setting is considerable. Note that MRR is not a reliable metric for TREC DL collections (compared to NDCG).

Additional Analysis. To provide a deeper understanding of the model performance, Figure 2 plots the negative sampling distribution for every 10,000 training steps. The negative sampling distribution of ANCE follows the uniform distribution (the dashed line). The proposed RANCE model assigns high probabilities to the top retrieved documents in early steps and then reduces their sampling probability as the number of training steps increases. This suggests that once the model becomes stable and performs strongly, RANCE discourages the model from taking negative samples from the top ranked documents as they are likely to be relevant. However, in early steps, the top retrieved documents would provide useful "hard negatives" for model optimization. This plot nicely demonstrates the behavior of the proposed RANCE model.

4 CONCLUSIONS AND FUTURE WORK

Experiments on the TREC Deep Learning Track data for both passage and document ranking tasks showed that the proposed RANCE model significantly outperforms the ANCE baseline in terms of NDCG, highlighting the impact of negative sampling distribution for dense retrieval. Our ablation study demonstrated the impact of each component of RANCE and showed that the dynamic error margin component is important only when the dense retrieval model is used for retrieval tasks instead of re-ranking. In the future, we will explore generating negative instances instead of selecting them from the collection to further improve the model's robustness.

5 ACKNOWLEDGEMENTS

This work was supported in part by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] Nick Craswell, Bhaskar Mitra, E. Yilmaz, Daniel Fernando Campos, and E. Voorhees. 2020. Overview of the TREC 2019 deep learning track. *ArXiv* abs/2102.07662 (2020).

[2] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural Ranking Models with Weak Supervision. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 65–74. DOI: http://dx.doi.org/10.1145/3077136.3080832

[3] Luyu Gao, Zhuyun Dai, Zhen Fan, and Jamie Callan. 2020. Complementing lexical retrieval with semantic residual embedding. *arXiv preprint arXiv:2004.13969* (2020).

[4] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In *CIKM'16*. 55–64.

[5] 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* 57, 6 (2020), 102067. DOI: http://dx.doi.org/https://doi.org/10.1016/j.ipm.2019.102067

[6] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data. In *CIKM'13*. 2333–2338.

[7] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* 20, 4 (Oct. 2002), 422–446. DOI: http://dx.doi.org/10.1145/582415.582418

[8] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).

[9] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 39–48. DOI: http://dx.doi.org/10.1145/3397271.3401075

[10] Victor Lavrenko and W. Bruce Croft. 2001. Relevance Based Language Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '01)*. Association for Computing Machinery, New York, NY, USA, 120–127. DOI: http://dx.doi.org/10.1145/383952.383972

[11] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent Retrieval for Weakly Supervised Open Domain Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 6086–6096. DOI: http://dx.doi.org/10.18653/v1/P19-1612

[12] Tie-Yan Liu. 2009. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.* 3, 3 (March 2009), 225–331. DOI: http://dx.doi.org/10.1561/1500000016

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[14] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to Match Using Local and Distributed Representations of Text for Web Search. In *WWW'17*. 1291–1299.

[15] Jay M. Ponte and W. Bruce Croft. 1998. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*. Association for Computing Machinery, New York, NY, USA, 275–281. DOI: http://dx.doi.org/10.1145/290941.291008

[16] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference (TREC-3)*. Gaithersburg, MD: NIST, 109–126.

[17] J. J. Rocchio. 1971. *Relevance Feedback in Information Retrieval*. Prentice Hall, Englewood, Cliffs, New Jersey.

[18] 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).

[19] Jinxi Xu and W. Bruce Croft. 1996. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96)*. Association for Computing Machinery, New York, NY, USA, 4–11. DOI: http://dx.doi.org/10.1145/243199.243202

[20] Hamed Zamani and W. Bruce Croft. 2018. On the Theory of Weak Supervision for Information Retrieval. In *Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR '18)*. Association for Computing Machinery, New York, NY, USA, 147–154. DOI: http://dx.doi.org/10.1145/3234944.3234968

[21] 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 *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 497–506. DOI: http://dx.doi.org/10.1145/3269206.3271800

[22] Chengxiang Zhai and John Lafferty. 2001. Model-Based Feedback in the Language Modeling Approach to Information Retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM '01)*. Association for Computing Machinery, New York, NY, USA, 403–410. DOI: http://dx.doi.org/10.1145/502585.502654