M4D-MKLab/ITI-CERTH Participation in TREC Deep Learning Track 2022

Alexandros-Michail Koufakis, Theodora Tsikrika,
Stefanos Vrochidis, Ioannis Kompatsiaris

Information Technologies Institute, Centre for Research and Technology Hellas,
6th Km. Charilaou - Thermi Road, 57001 Thermi-Thessaloniki, Greece
{akoufakis, theodora.tsikrika, stefanos, ikom}@iti.gr

Abstract

Our team's (CERTH_ITI_M4D) participation in TREC Deep Learning Track this year focused on the use of the pretrained BERT model in Query Expansion. We submitted two runs in the document retrieval subtask; both include a final reranking step. The first run incorporates a novel pooling approach for the Contextualized Embedding Query Expansion (CEQE) methodology. The second run introduces a novel term selection mechanism that complements the RM3 query expansion method by filtering disadvantageous expansion terms. The term selection mechanism capitalizes on the BERT model by fine tuning it to predict the quality of terms as expansion terms and can be used on any query expansion technique.

1 Introduction

The TREC Deep Learning (DL) track [1] promotes document and passage retrieval on a large dataset derived from the MS MARCO[2] from real Bing queries. The dataset includes a large number of partially labeled queries that enable supervised and semi-supervised approaches. This year, only the passages have been manually evaluated and the documents were ranked based on the relevant passages they contain. This change may potentially have affected the document ranking track and also the continuity with respect to previous years' document retrieval.

In recent years, large pretrained deep learning models have found great success in numerous natural language tasks, including information retrieval. These models are trained on huge amount of data on various tasks and they perform remarkably well, even on new tasks with zero or one-shot learning. Moreover, they can generate word embeddings that allow operations on their own vector space instead of word-based statistics. One of the first notable such models is ELMo[1]. BERT[2] is arguably the most well known with a very wide range of applications. More recently T5[3] have found great success on numerous text-to-text tasks. New large deep language models are continuously being developed, pushing the boundaries on the trainable parameter count, for example, GPT-3[4], Gopher[5], Megatron-Turing NLG[6], LLM-OPT[7]. In the domain of IR, multiple and diverse approaches (e.g. ColBERT[8], TILDE v2[9], EPIC 2020[10], SPLADEv2[11], COIL[12], EBR[13]) have already been developed capitalizing on such models.

Query Expansion (QE) is a well established technique[14, 15, 16, 17, 18] and entails the process of adding new terms in the original query to better represent the information need. Pseudo-Relevance Feedback (PRF) is a particular family of QE techniques that works on the assumption that the top-K retrieved documents are likely to be relevant and expands the query based on the contents of those

[1] `https://microsoft.github.io/msmarco/TREC-Deep-Learning.html`

[2] `https://microsoft.github.io/msmarco/`

documents. Recently, some studies [19, 20, 21, 22, 23] examined the potential of using contextualized embeddings from large pretrained models to perform QE with great results.

While QE and PRF have proven their efficacy, not every individual expansion term is beneficial [24]. There has been work carried out on the subject of term selection; for example, Jiang et al. [25] evaluated bigrams based on their statistical features and their ability to discriminate between documents. Another distinct approach is based on modified TF-IDF formulas with the use of external resources in order to determine terms that are able to discriminate well[26, 27]. Zhang et al.[28] used handcrafted term features in order to implement term selection in a supervised setting. Finally, Imani et al. used a neural network classifier for predicting the usefulness of a term, based on GloVe embeddings [29].

In this paper, we present our participation to the TREC DL 2022 competition where we submitted two runs on the document retrieval task. The first run builds upon the Contextualized Embedding Query Expansion (CEQE)[20] and our work[23] in the previous installment of TREC DL. Our second run includes a novel term selection mechanism that uses contextualized embeddings and can be applied in any QE pipeline. The rest of the document is structured as follows: Section 2 presents the original CEQE methodology along with the novel modification and the term selection technique. Section 3 describes the two submitted runs on the TREC DL 2022 track along with their evaluation. Section 4 concludes this paper with an overview and some remarks on future work.

2 Methodology

In this section, we first present Contextualized Embedding Query Expansion (CEQE)[20], a Query Expansion approach that uses contextualized embeddings. Then we present the two novel approaches; the max mention pooling method that is based on CEQE, and the Term filter approach that removes detrimental expansion terms.

2.1 Contextualized Embedding Query Expansion

CEQE is a query expansion method that is based probabilistic language model, especially the Relevance Model [16]. CEQE uses contextualized word embeddings (e.g. BERT embeddings) to calculate similarities instead of traditional metrics like term frequency (TF) and inverse document frequency (IDF). Moreover, CEQE addresses the independence assumption made by the Relevance Model that states that the query is independent to the expansion terms. This assumption is not valid in the case of contextualized embeddings as they inherently take into consideration their context.

Probabilistic language modeling approaches quantify the relevance of terms to a query in terms of the probability that the word be generated based on a language model. In the case of PRF the language model is calculated via the set of pseudo-relevant documents and the whole corpus. Equations 1 and 2 show that the probability of a word to be relevant based on the feedback relevance model (θ_R) is proportional to summation of some simpler probabilities that can be estimated through statistical metrics.

$$p(w|\theta_R) \propto \sum_{D \in R} p(w, Q, D) \tag{1}$$

$$\sum_{D \in R} p(w, Q, D) = \sum_{D \in R} p(w|Q, D)p(D) = \sum_{D \in R} p(w|D)p(Q|D)p(D) \tag{2}$$

In order to make the simplifications of equation 2, the original RM formulation assumed that query Q and term w are independent. In CEQE they note that this assumption is not valid in case of contextualized vector representations, as each word is dependent to its context. The CEQE parametrization is presented in eq 3:

$$\sum_{D \in R} p(w, Q, D) = \sum_{D \in R} p(w|Q, D)p(Q|D)p(D) \tag{3}$$

Moreover, in CEQE they propose three methods to calculate $p(w|Q, D)$ according to the updated formulation. First, in eq 4 they define $p(w|Q, D)$ as the normalized distances between the mentions of a word in a document ($\vec{m_w^D}$) and the centroid of the query ($\vec{Q}$). A word mention within a document is the embedding of the word given its context within the document. M_w^D is the complete set of mentions of a word in a document D. The centroid of a query is defined as the mean of the individual token embeddings, i.e. $\vec{Q} \triangleq \frac{1}{|Q|} \sum_{q_i \in Q} \vec{q}$, where q_i is a query token and $\vec{q}$ its embedding. The function δ is a similarity function, e.g. cosine similarity.

The BERT tokenizer sometimes splits words into multiple tokens (especially complex and long words), for example, "surfboarding" is split into three tokens "['surf', '##board', '##ing']". Such tokens are called wordpieces [30]. As CEQE works on word-level embeddings, it aggregates the individual wordpieces to compose the corresponding word embedding. In particular, it uses the centroid of the token embeddings as the aggregation method, $\vec{w} \triangleq \sum_{p_i \in w} \vec{p_i}$, where p_i are the wordpieces of word w.

$$p(w|Q, D) \triangleq \frac{\sum_{m_w^D \in M_w^D} \delta(\vec{Q}, \vec{m_w^D})}{\sum_{m^D \in M_*^D} \delta(\vec{Q}, \vec{m^D})} \tag{4}$$

The other two proposed methods of the new formulation are based on individual query term representations, instead of the centroid. Equation 5 shows the alternative form of the $p(w|q, D)$. This equation differs to eq 4 in that the mentions of a word are compared with all individual query terms. Thus, in order to have an overall similarity between the query and the word, a pooling step is performed. In particular, eq 6 (called "MaxPool") shows the first pooling technique, which defines that similarity to the most similar query term is selected. Eq 7 shows the alternative pooling technique that multiplies the similarities between a word and all the individual query terms. Finally, eq 8 normalizes the results of eq 6 and 7 in order for the final $p(w|Q, D)$ to be a relevance distribution of terms derived from contextual representations in top retrieved documents. Z' is a normalization factor that is the sum over the terms in document D.

$$p(w|q, D) \triangleq \frac{\sum_{m_w^D \in M_w^D} \delta(\vec{q}, \vec{m_w^D})}{\sum_{m^D \in M_*^D} \delta(\vec{q}, \vec{m^D})} \tag{5}$$

$$f_{max}(w, Q, D) = max_{q \in Q} p(w|q, D) \tag{6}$$

$$f_{prod}(w, Q, D) = \prod_{q \in Q} p(w|q, D) \tag{7}$$

$$p(w|Q, D) \triangleq \frac{f_{max/prod}(w, Q, D)}{Z'} \tag{8}$$

2.2 CEQE max mention

As described in the previous section, CEQE comes with three different pooling techniques (described in equations 4, 6, 5, 8, 7) that express the probability $p(w|Q, D)$. Intuitively, this probability can be thought of as a kind of similarity between the words from a document and the query. Essentially, words that have high probability are selected as expansion terms. Out of the three pooling techniques, the MaxPool (eq. 6) has been reported by the authors to perform better. Moreover, in the work of Khattab and Zaharia [8] (where they presented ColBERT) a MaxSim similarity metric was used that is in essence very similar to MaxPool. In both cases, the respective operation essentially measures the maximum similarity between a word and all individual terms of a query/document (query in CEQE, document in ColBERT).

In this work, we experimented with pushing the MaxPool operation one step further. Instead of using the maximum similarity between a mention of the word in a document ($\vec{m_w^D}$) and then summing over all the mentions (eq. 5) of the particular word, we propose using only the maximum similarity instead of summing.

$$p(w|q, D) \triangleq max_{m_w^D \in M_w^D} \delta(\vec{q}, \vec{m_w^D}) \tag{9}$$

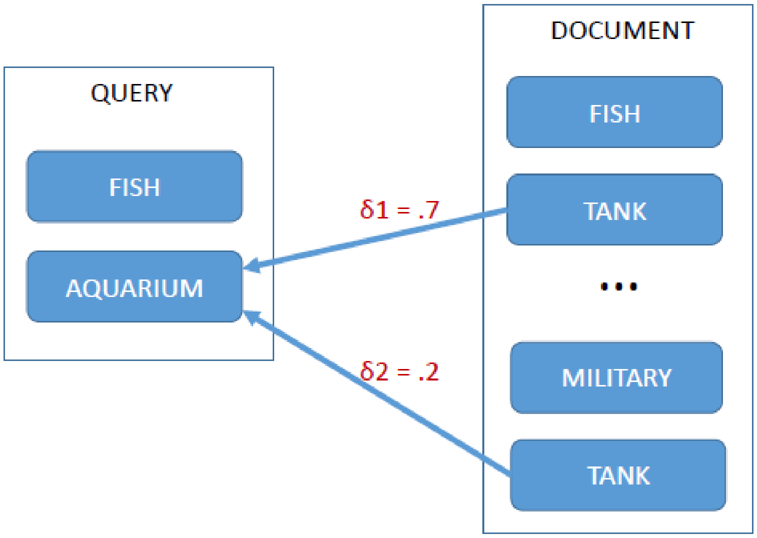

Figure 1: Example of the max mention pooling approach

For example, let us assume the query ”fish aquarium” and the document that contains the phrases ”fish tank” and ”military tank” (figure 1). In order to determine $p(w|Q, D)$ for the word ”tank”. MaxPool would sum $\delta_1 + \delta_2$ and because the second occurrence of ”tank” is in a completely different context the resulting probability would be lower. In the novel max mention method it would look optimistically only at the best occurrence of the term ”tank” (δ_1). Overall, this approach gives more importance to individual mentions of a particular word in a document.

2.3 Term Filtering

Term filtering is a novel standalone term selection technique that uses BERT. Term filtering complements any QE technique by judging the expansion terms and removing the terms that are unlikely to be helpful, thus improving the overall effectiveness. The novel term filtering technique was developed as a regression task that predicts the usefulness of a candidate expansion term. The training data were generated from the TREC DL test sets of 2019 and 2020. The regression model was developed as a fine-tuned version of BERT.

Algorithm 1 shows how the term filtering is applied on any QE expansion method. In detail, for a query$(query_i)$ the QE method generates a list of expansion terms (exp_terms). Then, each expansion term is evaluated by the term filter function $(F(term_j))$ and it is kept, only if the predicted usefulness is above a *threshold*. Finally, when every expansion term is validated, the final expanded query is combined with the useful expansion terms. The original term weights are respected and they are rescaled according to the QE algorithm parameters.

Algorithm 1 Term Filtering procedure

```
for $query_i \in Q$ do
    $exp\_terms = QE(query_i)$
    for $term_j \in exp\_terms$ do
        if $F(term_j) > threshold$ then
            $final\_exp\_terms \leftarrow term_j$
        end if
    end for
    $query_i = query_i + final\_exp\_terms$
end for
```

In order to train the Term filtering model, we used the TREC DL test sets of 2019 and 2020. We examined the improvement of individual expansion terms when added to a query. This approach was also used in previous supervised query expansion techniques [26, 29] to infer the usefulness (or gain) of a single expansion term. We calculated the gain in terms of the change in the metric bpref[31] which resembles Mean Average Precision (map) but does not assume that unjudged documents are irrelevant. In fact, it makes no assumptions about unjudged documents, but computes the score based on the rank of **known** relevant/irrelevant documents.

We used RM3[32] to generate 50 expansion terms for each of the 88 queries (combined the test set queries of 2019 & 2020). This resulted in 4,400 training examples in the form of triplets $(query, exp\_term, gain)$. The term filtering model was built upon the "bert-base-uncased"[3] fine-tuned for regression. The training data were fed to the BERT model as two sequences separated with the $< SEP >$ token, i.e: $(Query, < SEP >, exp\_term)$ with the appropriate tokenization. The model was trained for 3 epochs with maximum length of 128 tokens and batch size 8.

3 Evaluation

In this section, we discuss the two runs we submitted to the TREC-DL 2022 to the document retrieval task. In both runs, the results were generated by pipelines that include a final reranking step. The first run uses the novel Max mention pooling technique discussed in Section 2.2 and the second run uses the novel term filtering method discussed in Section 2.3. The pipelines were developed in the pyterrier platform[4] and executed on a machine with GTX 2080Ti 11GB, 128GB RAM, and an HDD for all stages of our experiments. The query expansion methods (CEQE and RM3) were tuned on TREC DL 2019 and 2020 test set via grid search. Table 1 shows the parameters that were tuned and the range they were examined. The parameters *fb_docs* and *fb_terms* represent the number of feedback documents and the number of feedback terms and they follow the naming convention of the pyterrier platform. The last parameter *lambda* (λ) defines the term weight coefficient of the expansion terms.

Parameter Name	**Range/Values**
fb_docs	5, 10, ... 25
fb_terms	5, 10, ... 40
lambda	0.1, 0.2, ... 0.9

Table 1: The parameters that were tuned via grid search

The **first submitted run** (ID: ceqe_custom_rerank) was the result of a pipeline with query expansion with CEQE (using BERT-base-uncased) and a final reranking step with ColBERT[5]. CEQE was used with the Max mention (cf. Section 2.2) pooling technique and additionally our earlier IDF modification[23] that incorporates the IDF into the term weights. The parameters that were found to perform best are $fb\_docs = 25$, $fb\_terms = 30$ and $lambda = 0.8$. The complete **pipeline** follows the steps:

1. BM25 for initial retrieval;
2. Query expansion with CEQE with Max mention pooling technique;
3. BM25 on expanded queries; and
4. Document reranking with ColBERT.

The **second submitted run** (ID: rm3_term_filter_rerank) was composed of a RM3 query expansion approach along with the Term filtering method (cf. Section 2.3) and a reranking step with ColBERT. The parameters for RM3 defined by grid search are $fb\_docs = 5$, $fb\_terms = 40$ and $lambda = 0.5$. The run **pipeline** includes the steps:

[3] https://huggingface.co/bert-base-uncased

[4] https://pyterrier.readthedocs.io/en/latest/

[5] https://github.com/terrierteam/pyterrier_colbert

1. BM25 for initial retrieval;
2. RM3 for query expansion;
3. Term filtering for improving the RM3 expansion terms;
4. BM25 on expanded queries; and
5. Document reranking with ColBERT.

Table 2 shows the comparison between our two runs with respect to Precision with cutoff at 10 (*p@10*), Normalized Discounted Cumulative Gain with cutoff at 10 (*ndcg@10*), Mean Average Precision (*map*), and Reciprocal Rank (*recip_rank*). The **ceqe_custom_rerank** performed slightly better than the **rm3_term_filter_rerank** in *p@10*, *ndcg@10*, and *map*, but only with small differences of 0.0013, 0.0200 and 0.0410, respectively. In *recip_rank*, the run **rm3_term_filter_rerank** performed better than **ceqe_custom_rerank** by 0.219. The differences are not sufficient to offer conclusive comparisons, but it appears that the two runs performed at a similar level. The difference in *recip_rank* is slightly greater but given that both runs had the same final reranking step, this difference is likely incidental.

	p@10	**ndcg@10**	**map**	**recip_rank**
CEQE Custom Rerank	**0.4842**	**0.3811**	**0.1090**	0.7382
RM3 Filter Rerank	0.4829	0.3611	0.1049	**0.7601**

Table 2: The performance of our two submitted runs on Precision at 10 (p@10), Normalized Discounted Cumulative Gain at 10 (ndcs@10), Mean Average Precision (map), Reciprocal Rank (recip_rank).

Figure 2 illustrates the performance of the submitted runs measured by Precision with cutoff at 10 (*p@10*) and Normalized Discounted Cumulative Gain with cutoff at 10 (*ndcg@10*). Only the *p@10* and *ndcg@10* were provided by the organizers for the other teams' submitted runs. The runs in the figure were sorted by Precision in descending order. Our two submitted runs are **ceqe_custom_rerank** & **rm3_term_filter_rerank** and are on the bottom half of the rankings.

Figures 3, 4, 5 and 6 show the scores achieved by our two runs per query in the test set. The symbols α and β show the performance of our runs, **ceqe_custom_rerank** and **rm3_term_filter_rerank** respectively. The performance was measured in Mean Average Precision (fig. 3), NDCG with cutoff at 10 (fig. 4), precision with cutoff at 10 (fig. 5), and Reciprocal Rank (fig. 6). We did not identify any clear trend between our two runs for any of the metrics.

4 Conclusions

In this notebook paper, we presented our participation to the TREC DL 2022 track. We submitted two runs to the document retrieval task with different approaches that both exploit BERT. The first run was a query expansion pipeline on contextualized embeddings based on CEQE with a novel pooling mechanism. The second run was a RM3 query expansion pipeline with a novel term selection mechanism built upon the BERT model. The BERT model was fine-tuned to predict the quality of expansion terms. Both submitted runs include a final reranking step with ColBERT. On the official TREC evaluation, our runs performed bellow the mean compared to the other submitted runs. Future improvements on the term selection mechanism could include additional data and more reliable evaluation of the usefulness for the train data.

5 Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under projects CREST (Grant Agreement No 833464) and APPRAISE (Grant Agreement No 101021981).

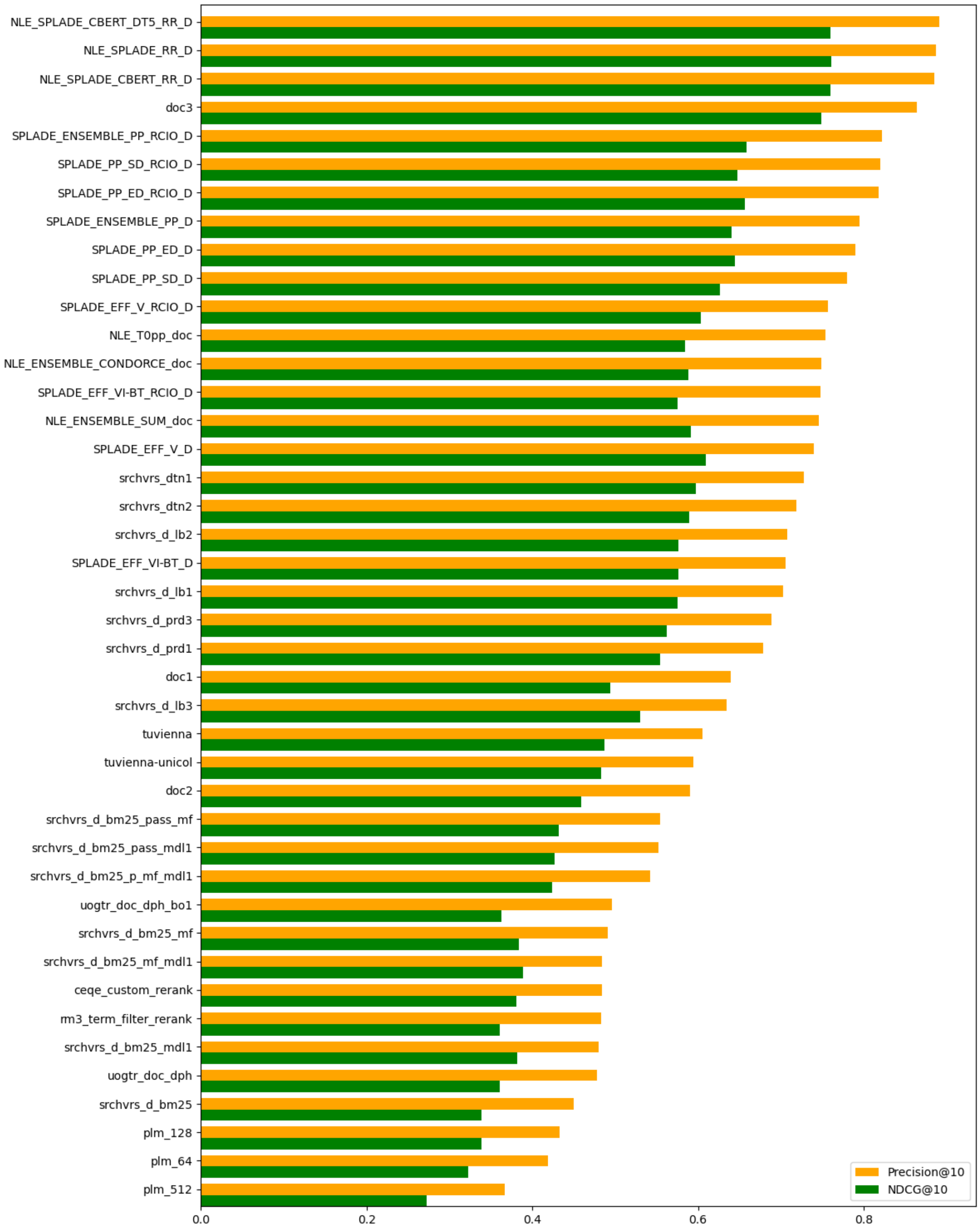

Figure 2: All the submitted runs on document retrieval subtrack (our runs are **ceqe_custom_rerank** and **rm3_term_filter_rerank**)

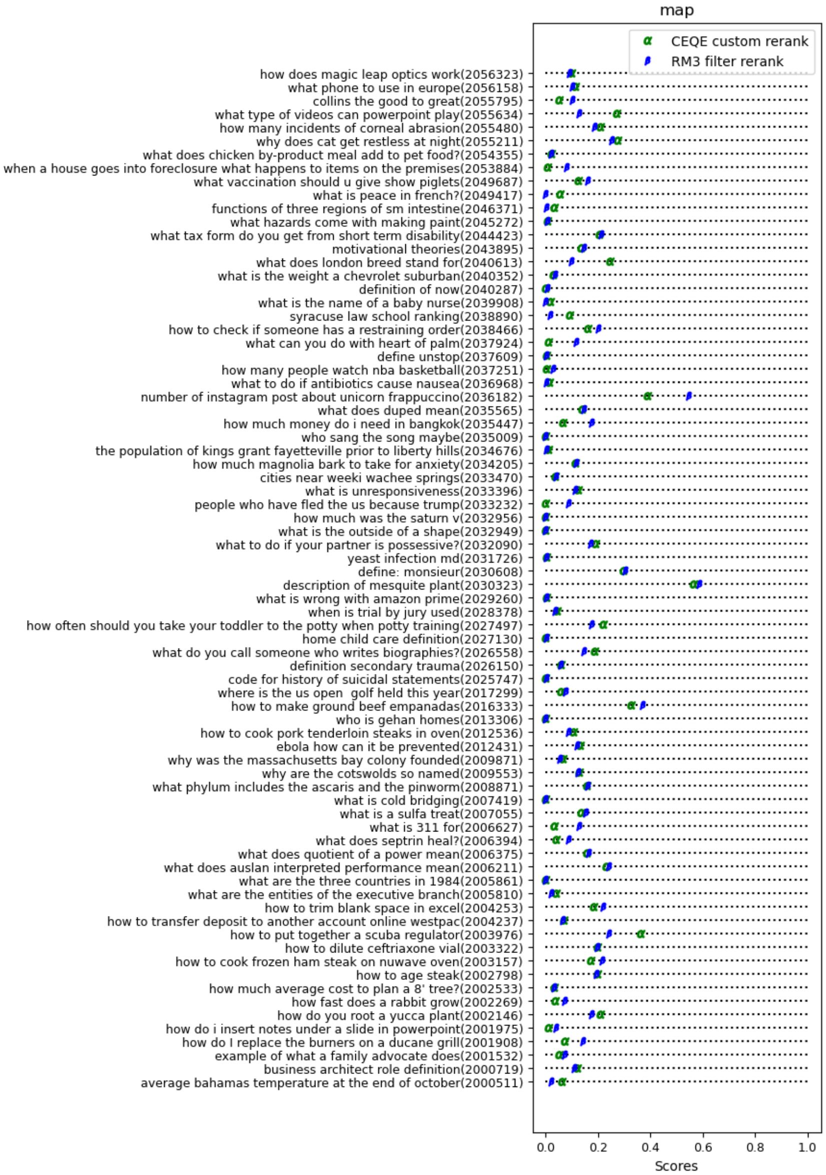

Figure 3: Mean Average Precision per query

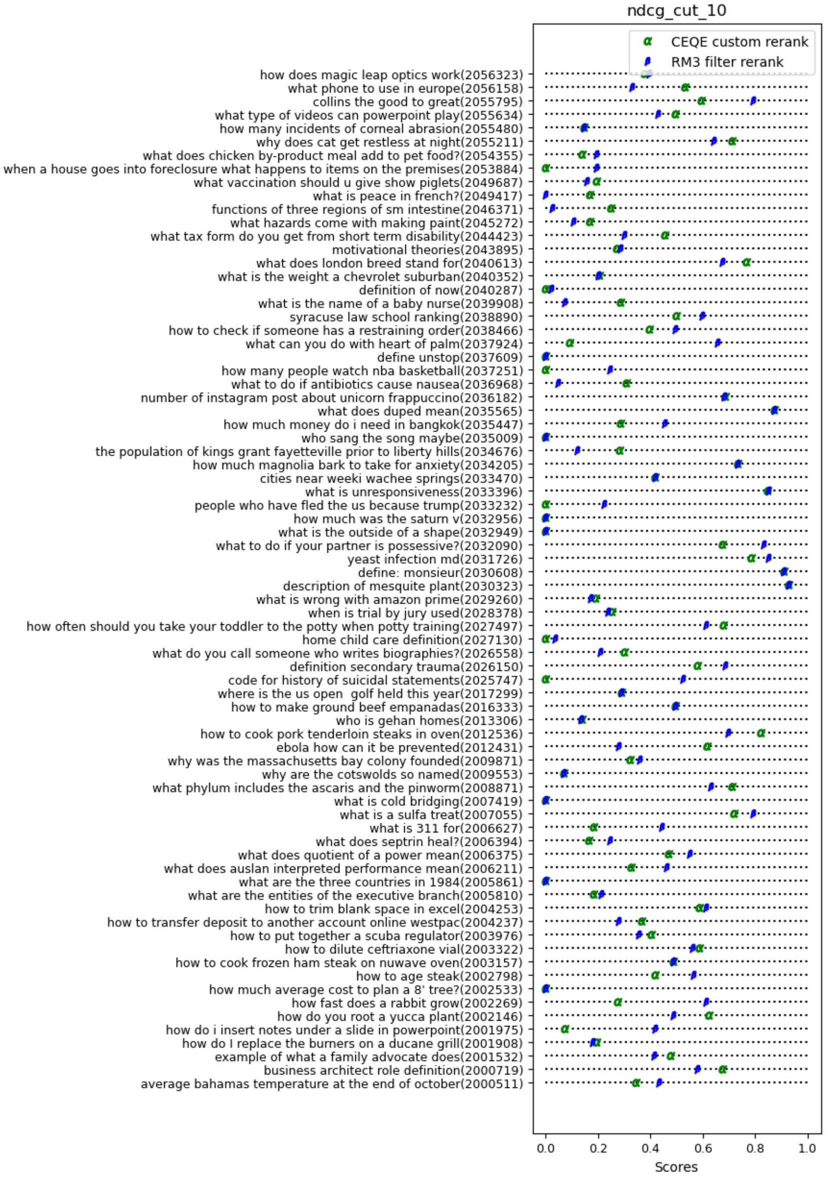

Figure 4: NDCG at 10 score per query

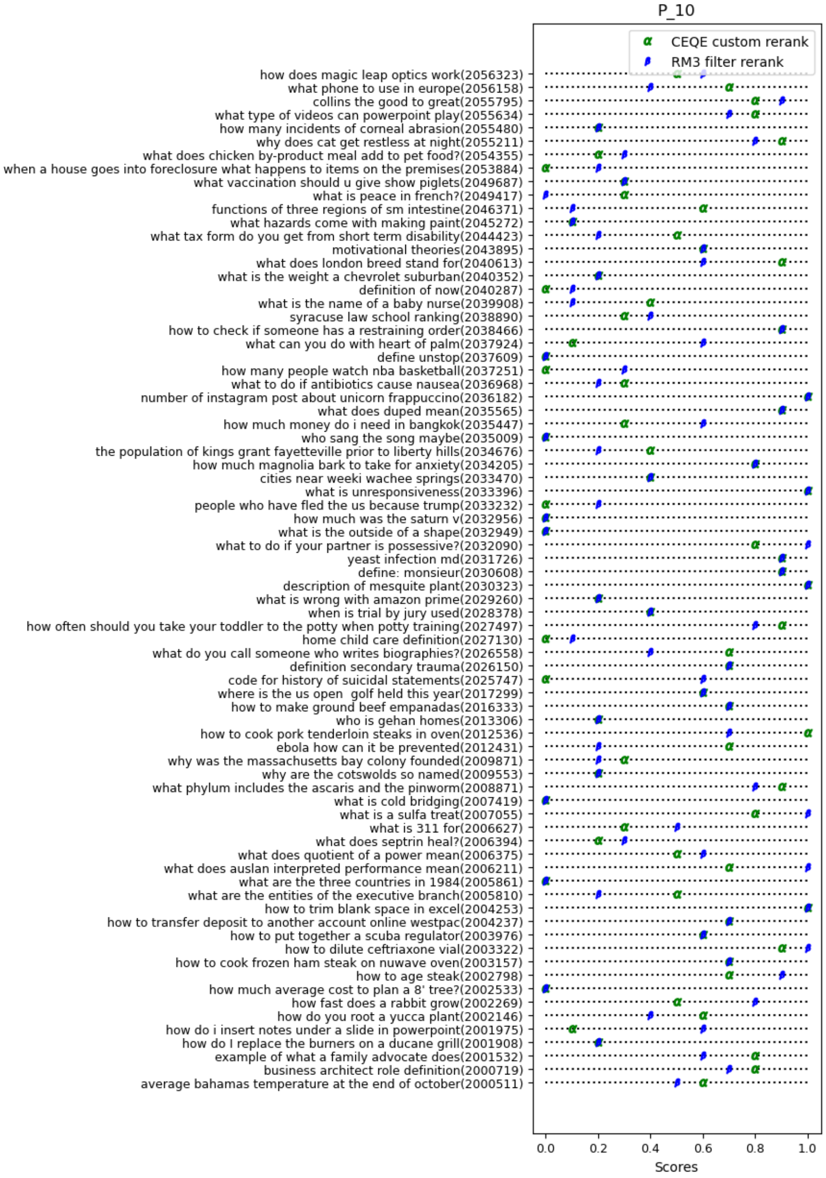

Figure 5: Precision at 10 per query

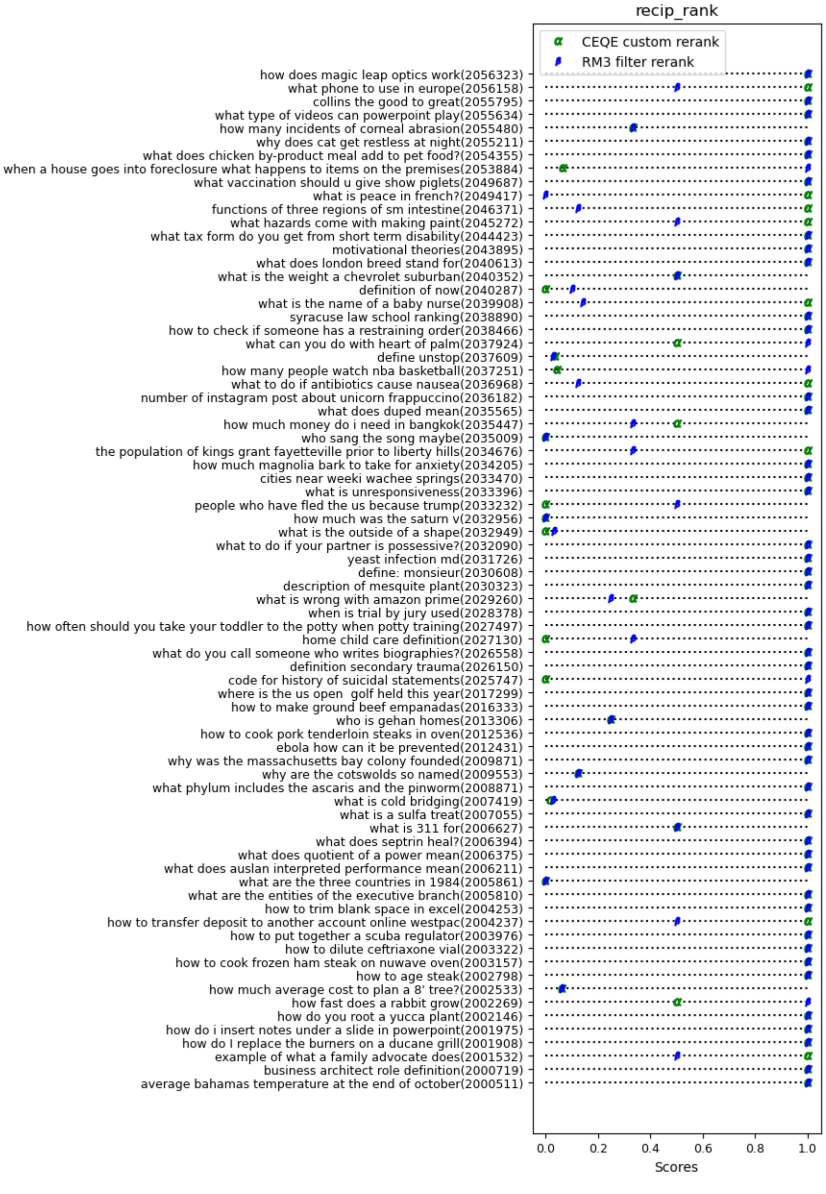

Figure 6: Reciprocal Rank per query

References

[1] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

[6] Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*, 2022.

[7] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[8] Omar Khattab and Matei Zaharia. 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*, pages 39–48, 2020.

[9] Shengyao Zhuang and Guido Zuccon. Fast passage re-ranking with contextualized exact term matching and efficient passage expansion. *arXiv preprint arXiv:2108.08513*, 2021.

[10] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. Expansion via prediction of importance with contextualization. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1573–1576, 2020.

[11] Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021.

[12] Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. *arXiv preprint arXiv:2104.07186*, 2021.

[13] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2553–2561, 2020.

[14] Joseph Rocchio. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323, 1971.

[15] Jinxi Xu and W Bruce Croft. Quary expansion using local and global document analysis. In *Acm sigir forum*, volume 51, pages 168–175. ACM New York, NY, USA, 2017.

[16] Victor Lavrenko and W Bruce Croft. Relevance-based language models. In *ACM SIGIR Forum*, volume 51, pages 260–267. ACM New York, NY, USA, 2017.

[17] Yuanhua Lv and ChengXiang Zhai. Revisiting the divergence minimization feedback model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1863–1866, 2014.

[18] Javier Parapar, Manuel A Presedo-Quindimil, and Alvaro Barreiro. Score distributions for pseudo relevance feedback. *Information Sciences*, 273:171–181, 2014.

[19] Vincent Claveau. Query expansion with artificially generated texts. *arXiv preprint arXiv:2012.08787*, 2020.

[20] Shahrzad Naseri, Jeffrey Dalton, Andrew Yates, and James Allan. Ceqe: Contextualized embeddings for query expansion. In Djoerd Hiemstra, Marie-Francine Moens, Josiane Mothe, Raffaele Perego, Martin Potthast, and Fabrizio Sebastiani, editors, *Advances in Information Retrieval*, pages 467–482, Cham, 2021. Springer International Publishing.

[21] Zhi Zheng, Kai Hui, Ben He, Xianpei Han, Le Sun, and Andrew Yates. Bert-qe: contextualized query expansion for document re-ranking. *arXiv preprint arXiv:2009.07258*, 2020.

[22] Xiao Wang, Craig Macdonald, Nicola Tonellotto, and Iadh Ounis. Pseudo-relevance feedback for multiple representation dense retrieval. *arXiv preprint arXiv:2106.11251*, 2021.

[23] Alexandros-Michail Koufakis, Theodora Tsikrika, Stefanos Vrochidis, and Ioannis Kompatsiaris. M4d-mklab/iti-certh participation in trec deep learning track 2021. 2022.

[24] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250, 2008.

[25] Maojin Jiang, Eric Jensen, Steve Beitzel, and Shlomo Argamon. Choosing the right bigrams for information retrieval. In *Classification, Clustering, and Data Mining Applications*, pages 531–540. Springer, 2004.

[26] Chenyan Xiong and Jamie Callan. Query expansion with freebase. In *Proceedings of the 2015 international conference on the theory of information retrieval*, pages 111–120, 2015.

[27] Andisheh Keikha, Faezeh Ensan, and Ebrahim Bagheri. Query expansion using pseudo relevance feedback on wikipedia. *Journal of Intelligent Information Systems*, 50(3):455–478, 2018.

[28] Zhiwei Zhang, Qifan Wang, Luo Si, and Jianfeng Gao. Learning for efficient supervised query expansion via two-stage feature selection. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 265–274, 2016.

[29] Ayyoob Imani, Amir Vakili, Ali Montazer, and Azadeh Shakery. Deep neural networks for query expansion using word embeddings. In *European Conference on Information Retrieval*, pages 203–210. Springer, 2019.

[30] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

[31] Chris Buckley and Ellen M Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, 2004.

[32] Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. Umass at trec 2004: Novelty and hard. *Computer Science Department Faculty Publication Series*, page 189, 2004.