M4D-MKLab/ITI-CERTH Participation in TREC Deep Learning Track 2021

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*) goal in the TREC Deep Learning Track was to study how the Contextualized Embedding Query Expansion (CEQE) [1] method performs in such setting and how our proposed modifications affect the performance. In particular, we examine how CEQE performs with the addition of bigrams as potential expansion terms, and how an IDF weight component affects the performance. The first run we submitted is produced by a query expansion pipeline that uses BM25 for retrieval and CEQE with the IDF modification for query expansion. The second submitted run used a modification of CEQE with the addition of bigrams as candidate expansion terms and a re-ranking step using CEDR. Our runs showed promising results, especially for Average Precision.

1 Introduction

The TREC Deep Learning (DL) track[1] is actively developing and improving a large dataset based on MS MARCO for document and passage retrieval. It offers ample data for the development of novel methodologies while it has the challenge of partially labeled training data that incentivizes semi-supervised and transfer learning.

Recently, transformers [2, 3, 4, 5, 6, 7] and other DL architectures [8, 9] were used to produce word embeddings that allow operations in the vector space instead of using word-based statistics. Moreover, in some cases contextualized word embeddings were produced (which means that the embeddings depend on the neighboring words). Such contextualized embeddings are better equipped to tackle polysemy and other issues that require understanding of the context. Word embeddings have been already studied [10, 11, 12, 13, 14, 15] in numerous Information Retrieval (IR) tasks showcasing great results.

Query Expansion (QE) is a well established technique [16, 17, 18, 19, 20] 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 documents. Recently, some studies [21, 22, 1, 23, 24] examined the potential of using contextualized embeddings to perform QE. These studies reported great results, but there are more directions open for exploration. In particular, we performed some preliminary studies on variations of Contextualized Embedding Query Expansion (CEQE) [1]:

- How does IDF affect the performance?
- How does the addition of bigrams affect the performance?

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

- How does the addition of term weight affects the performance?

The rest of the document is structured as follows; Section 2 presents the contextualized embedding methodology and the variations we examined, Section 3 analyzes the experiments and evaluates our performance in TREC DL 2021, and finally Section 4 concludes with an overview of the paper and some future directions for improvement.

2 Methodology

In this section we present briefly the original CEQE [1] formulation and its derivation from probabilistic language modeling approaches, especially the Relevance Model [18]. Subsequently, the two new modifications of CEQE are presented. The two modifications correspond to the runs submitted by our team to TREC DL 2021.

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 in original RM formulation is 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 [25]. 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.

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

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

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

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

In this work we only used the MaxPool method as it produced the best results as the authors of CEQE showed, and we confirmed it through our own experimentation. The MaxPool formulation seems to be effective in a broader context within the contextualized embeddings as in ColBERT used a similar function for similarity. In particular Khattab and Zaharia [26] named it MaxSim but is essentially the max similarity between a word and the individual terms of a query/document (query in CEQE, document in ColBERT).

2.1 CEQE with IDF and term weights

Clinchant and Gaussier in their analysis on PRF models [27] observed that well-known PRF models of that period (published in 2013) tend to select common terms with low IDF, violating the heuristics constraints Fang et al. [28] formulated, indicating sub-optimal performance.

In proposed CEQE implementations, the IDF effect is not explicitly satisfied via the equations, as the probability $p(w|Q, D)$ considers solely the similarity within the feedback set. No term distribution metrics are used.

$$f'_{max}(w, Q, D) = max_{q \in Q}(IDF_w * p(w|q, D))$$

The updated equation (modifies the equation 6) by injecting the IDF of a word w as weight to the probability $p(w|q, D)$. For our preliminary tests we do not modify the normalization step (eq 8) which means that the final scores are not probabilities. However, we performed QE by adding the proposed terms with their corresponding weights to the original query, and the fact that their score is not a probability does not affect this method.

Moreover, we tried to use the scores as weights in the final retrieval step. The motive for this is to encourage terms that are highly similar to the query (thus having increased score) while discouraging less similar terms. In particular, the parameter λ $(0 < \lambda < 1)$ applies a weight to the original query terms and $\lambda - 1$ weight is applied on the expansion terms. For example, without using weights the query "types of dysarthria from cerebral palsy" with $\lambda = 0.2$ would be expanded:

type^0.2 cerebral^0.2 dysarthria^0.2 of^0.2 palsy^0.2 from^0.2 speech^0.8 motor^0.8 muscles^0.8 ...

Where the first terms are terms from the original query with the weight $\lambda = 0.2$ and the following terms are the expansion terms with weight $\lambda = 1 - 0.2 = 0.8$. While when applying the scores as weights it would be similar to:

type^0.2 cerebral^0.2 dysarthria^0.2 of^0.2 palsy^0.2 from^0.2 speech^0.059 motor^0.018 muscles^0.018 ...

2.2 CEQE with bigrams

Another direction that modern QE works on contextualized embeddings seem to leave space for improvement is the utilization of n-grams. Particularly so, because BERT and contextualized embeddings inherently consider context, we hypothesized that groups of neighboring words would represent better the particular meaning of a query. For example, if a word in a particular context is found to match better the meaning of the query, then the neighboring words are likely to be important as well for the particular meaning. Thus, the expansion with bigrams could prove more beneficial than single words in the representation of the information need.

We generated both unigrams and bigrams from the feedback documents as candidate expansion terms that undergo a similar selection process. The bigram embeddings were generated as the centroid of the two individual terms, in similar fashion to the transition from wordpieces to words. We devised a procedure for selecting the best expansion terms from the pool of unigrams and bigrams. This procedure includes some filtering through the potential bigram expansion terms to remove near duplicates and noisy terms. In detail, the following list presents the filtering conditions:

1. No bigrams with stopwords.
2. No bigrams with numbers.
3. No bigrams with terms from the original query.
4. No bigrams with terms that are already selected as unigram expansion term.
5. No bigrams with terms with document frequency less than 25.

We observed that stopwords and numbers (conditions 1 & 2) tended to dilute the bigrams and result in bigrams that are not significantly different than their unigram counterparts. Bigrams that include terms from the original query (condition 3) tended to be already sufficiently represented. Likewise, terms that were already selected as unigrams (condition 4) tended to be sufficiently represented already. Finally, similar to AlQatan et al. [29] we used words above a minimum document frequency (condition 5). Too infrequent terms often were result of imperfect data collection, for example the mangled words ”unmuteif” and ”palsydysarthria” have low document frequency and indeed they are not good expansion terms. While BERT can make sense of such words, traditional word based methods are hindered by them. Moreover, words that are so under-represented in the collection offer limited potential for improvement as they influence just a handful of document. For example, the query ”types of dysarthria from cerebral palsy” is expanded[2]:
#combine:0=0.95:1=0.95:2=0.95:3=0.95:4=0.95:5=0.95:6=0.05:7=0.05: [...] 41=0.05:42=0.05 [...] (types of dysarthria from cerebral palsy speech symptoms [...] #1(lobe cranially) #1(also symptom) [...])
Where the first terms are from the original query with weight 0.95 and the following are unigram and bigram expansion terms with weight 0.05.

3 Experiments

In this section we describe our two submitted runs and offer some analysis on the results. Our modifications on the original CEQE methods were implemented based on the official code[3]. CEQE and the new modifications were integrated to the pyterrier platform[4]. We had a machine with GTX 2080Ti 11GB, 128GB RAM, and an HDD for all stages of our experiments. In both of our runs, the best parameters for the models were selected after grid search using the qrels of TREC DL 2019 and 2020 on the new dataset (this year's MS MARCO v2). In particular, the parameters that were tuned are shown in the table 1. 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 that we used. The last parameter *lambda* (λ) defines the term weight coefficient as described

[2] The query formatting is different than in subsection 2.1 in order to enable pair of words to match

[3] https://github.com/sherinaseri/ceqe-release

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

Parameter Name	**Range/Values**
fb_docs	5, 10, 15
fb_terms	10, 15, ... 60
lambda	0.05, 0.1, ... 0.9, 0.95

Table 1: The parameters that were tuned via grid search

in section 2.1. Documents were preprocessed with Porter Stemmer and punctuation and stopword removal.

3.1 Run 1: CEQE with IDF and term weights

Our first submitted run (ID: bigrams_cont_qe[5]) was the result of a query expansion pipeline without reranking. The query expansion was performed via the CEQE algorithm with the addition of IDF component optimized for Normalized Discounted Cumulative Gain with cutoff at 10 (NDCG@10) and the addition of the CEQE scores as term weights (cf. section 2.1). The parameters that yielded the best results in the validation set were $fb\_docs = 5$, $fb\_terms = 45$, $lambda = 0.2$. The pipeline follows the three steps:

1. BM25 for initial retrieval.
2. Query expansion with CEQE IDF with the score as weights.
3. BM25 on expanded queries.

Initially the default BM25 retrieves a set of documents (the number is dictated by the *fb_docs* parameter) for the original query, then the CEQE algorithm expands the query with *fb_terms* expansion terms, and the final documents are retrieved according to the expanded query using the default BM25 again. Inference took 30 seconds per query, but the delay is mostly due to the large size of the index and the absence of an SSD drive.

Figure 1 shows the percentage of queries that were above, below or at the median in our first run. Each piechart shows the performance for a different metric. In Average Precision ("map") 74% queries achieved better performance than the median. In Precision at 10 ("P_10") 21% of queries were above median and 26% below. In the case of Reciprocal Rank ("recip_rank") is not easy to evaluate our performance because the median in all but one query was 1, indicating that most runs retrieved a relevant document in the first position. As the organizers pointed out, due to the large size of the dataset, the number of positive labels is very large. This caused a large number of perfect scores in Precision at 10 and Reciprocal Rank. In Normalized Discounted Cumulative Gain with cutoff at 10 ("ndcg_cut_10") we scored above median in slightly less than half (46%) of the queries.

Figures 3, 4, 5 and 6 show the performance of our runs per query for the four different metrics. The black horizontal lines cover the whole range between the best and worst performance across all the submitted runs. Figures 5 and 6 confirm that that many runs reached the perfect score at precision@10 and reciprocal due to the numerous positive results in the dataset. Otherwise, there does not seem to be a clear tendency in our run 1.

Overall, our run performs very well in Average Precision, which is the only metric that evaluates all the top 100 documents. This indicates that our run performs comparatively better outside the top 10 documents. NDCG@10 is the only metric that considers the different similarity labels (0: irrelevant, 1-4: gradually more relevant). The other metrics transform the labels to binary (0: irrelevant, 1-4: relevant). NDCG@10 seems to follow a similar pattern with Precision at 10 in that it performs slightly bellow median, indicating no meaningful difference with the two scoring approaches.

3.2 Run 2: CEQE with Bigrams and CEDR reranking

The second run (ID: bigram_qe_cedr) improves on the previous pipeline with the addition of a reranking step at the end. This run uses the bigram variation of CEQE algorithm (cf. section 2.2) and was

[5] The ID mistakenly suggests the use of bigrams

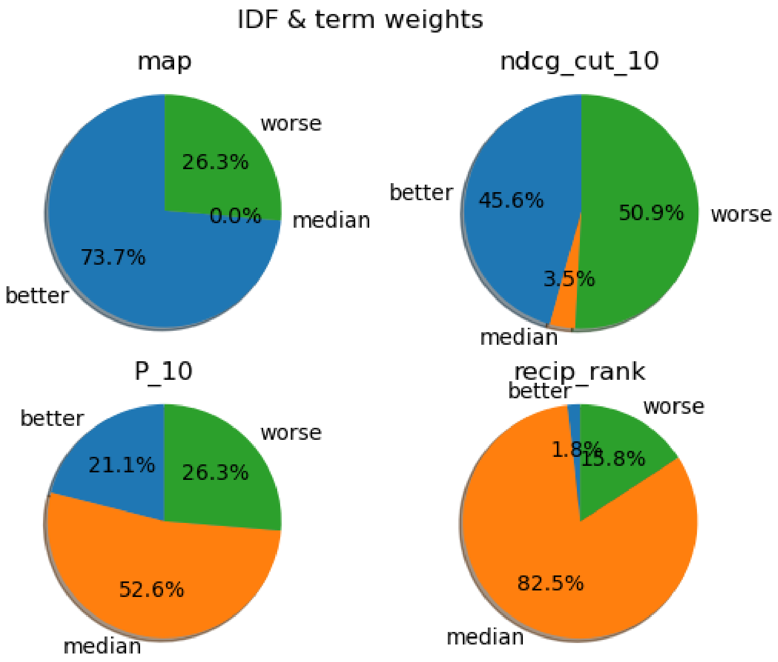

Figure 1: Percentage of queries that performed better/worse than the median

optimized for recall@100 as an reranking step was employed. The best parameters were $fb\_docs = 5$, $fb\_terms = 30$, $lambda = 0.95$.

In detail the steps:

1. BM25 for initial retrieval.
2. Query expansion with CEQE Bigrams.
3. BM25 on expanded queries.
4. Reranking with CEDR.

BM25 performed the initial retrieval stage, followed by QE with Bigram variation of CEQE, then BM25 retrieved documents based on the expanded queries, finally, a CEDR model re-ranked the top 100 documents. CEDR was trained with batch size 512, for 126 iterations with early stopping. We used CEDR as a re-ranking step after the query expansion step, to examine the effect of the improved recall along with a final document re-ranking stage. We chose to use CEDR for re-ranking as it produces good results in a reasonable time frame without the necessity of the resource intensive BERT re-training/fine-tuning. We used the PyTerrier port of the CEDR algorithm[6] and the "bert-base-uncased"[7] BERT model.

Inference took approximately 35 seconds per query which is 5 seconds slower than the first run that did not include a re-ranking step. This indicates that the re-ranking step did not add a great overhead, and as previously the main cause of delay was the slow disk access (HDD).

Figure 2 shows the percentage of queries that performed better/worse than median for our second run. The second run seems to follow the tendencies of the first, meaning that it performs very well on Average Precision but less so in the other metrics. This indicates once more that our methodology under-performs in the top 10 documents while performing better than the median in the top 100, in

[6] https://github.com/cmacdonald/pyterrier_bert

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

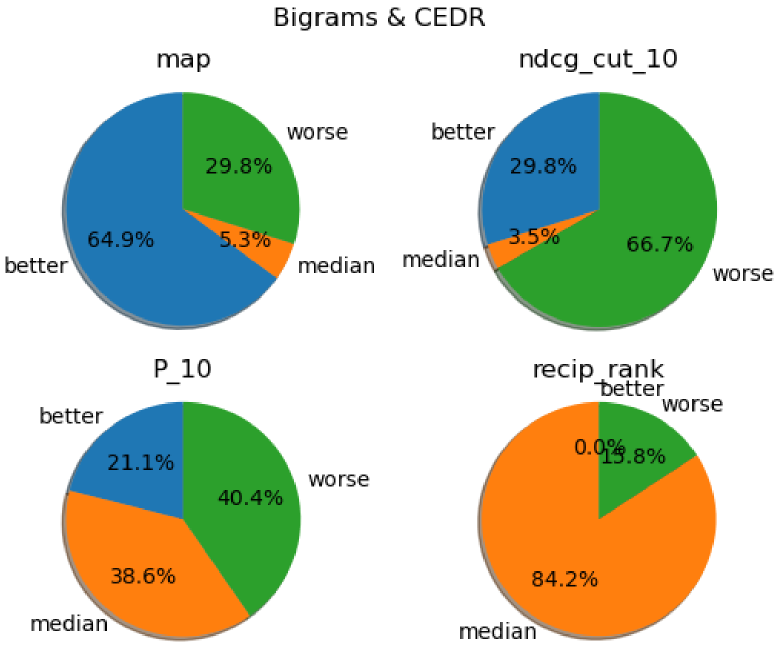

Figure 2: Percentage of queries that performed better/worse than the median

most cases. This result is counter-intuitive given that run 2 has an re-ranking step contrary to run 1. The re-ranking step is expected to rearrange the top 100 documents and boost the score of the most similar, thus, increasing the top 10 metrics (P_10, ndcg_cut_10). In practice, it appears that the re-ranking step harmed the performance uniformly in all metrics. This is possibly due to the use of the default BERT model (not fine-tuned) for CEDR, incompatibility issues with the training data (previous years' qrels were used), or some other inaccuracy in CEDR's training.

Figures 3, 4, 5 and 6 seem to reinforce the notion that run 2 performed better than median on Average Precision and less so in the other metrics, but we did not identify any other pattern in relation to run 1.

4 Conclusion

In this paper we presented our participation in TREC DL 2021. We submitted two runs for the document ranking task with some variations of PRF based on BERT embeddings. In our first run we examined the direct addition of IDF as a score component and used the scores as term weights. In our second run we performed QE on both unigrams and bigrams and finally added a CEDR re-ranking stage. The resuls show that our runs performed very well on Average Precision on top 100 documents while they were not as effective in the metrics that examined the top 10 documents. Interestingly, the second run that included a re-ranking step did not perform as well as the first run, likely due to ineffective re-ranking. Overall, our runs' performance was promising and indicates that it can benefit greatly from an effective re-ranking stage.

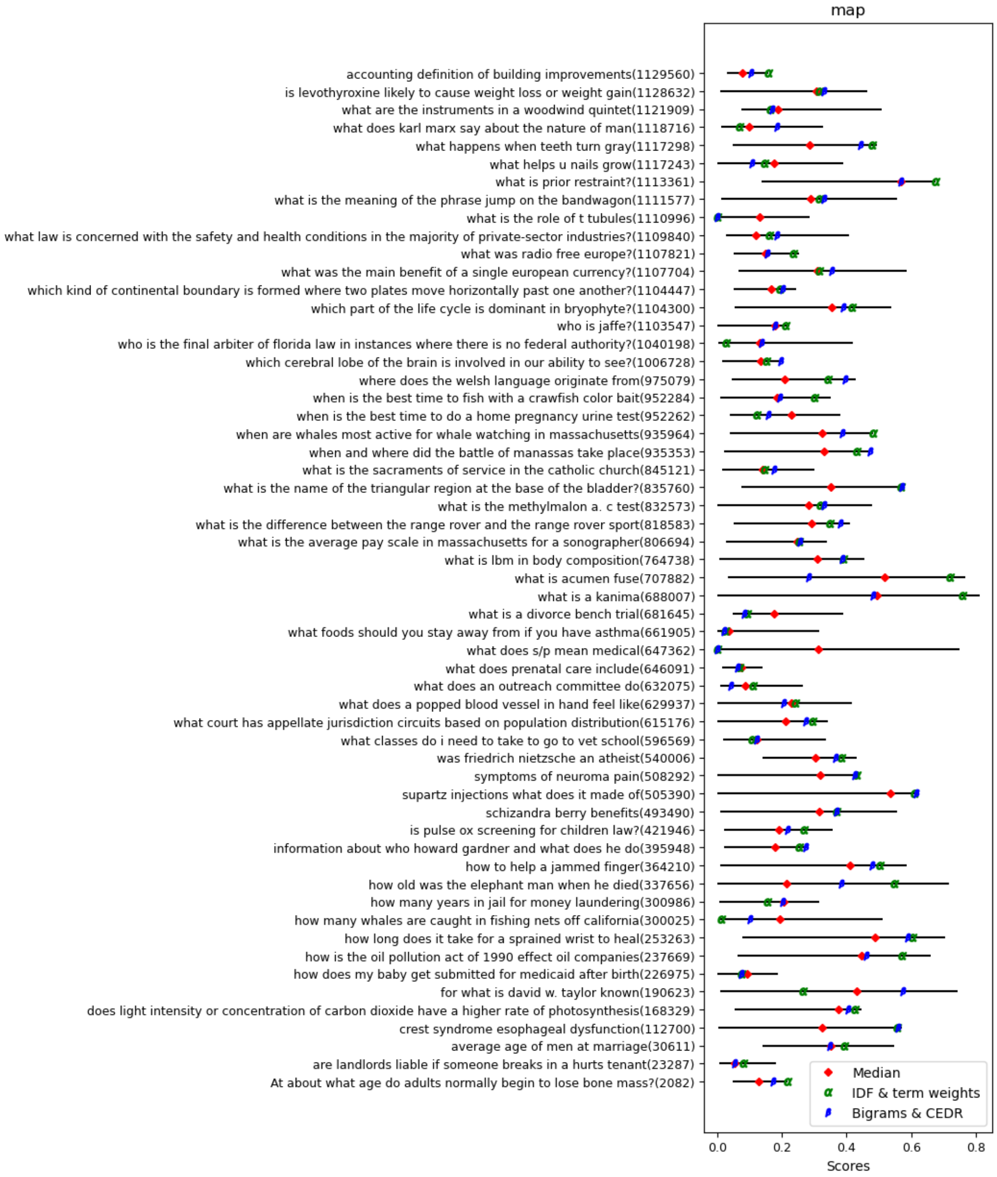

Figure 3: Average Precision per query

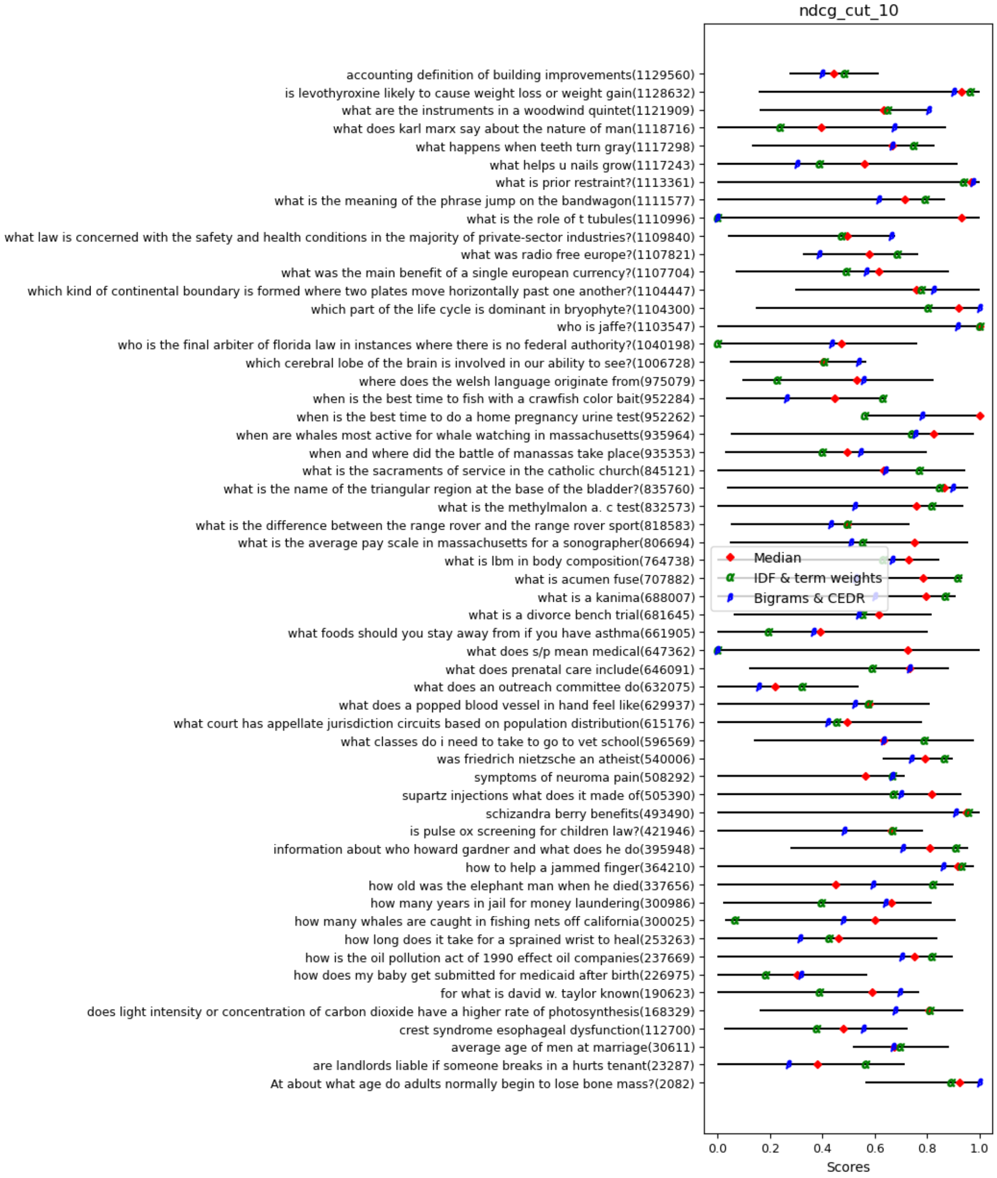

Figure 4: NDCG at 10 score per query

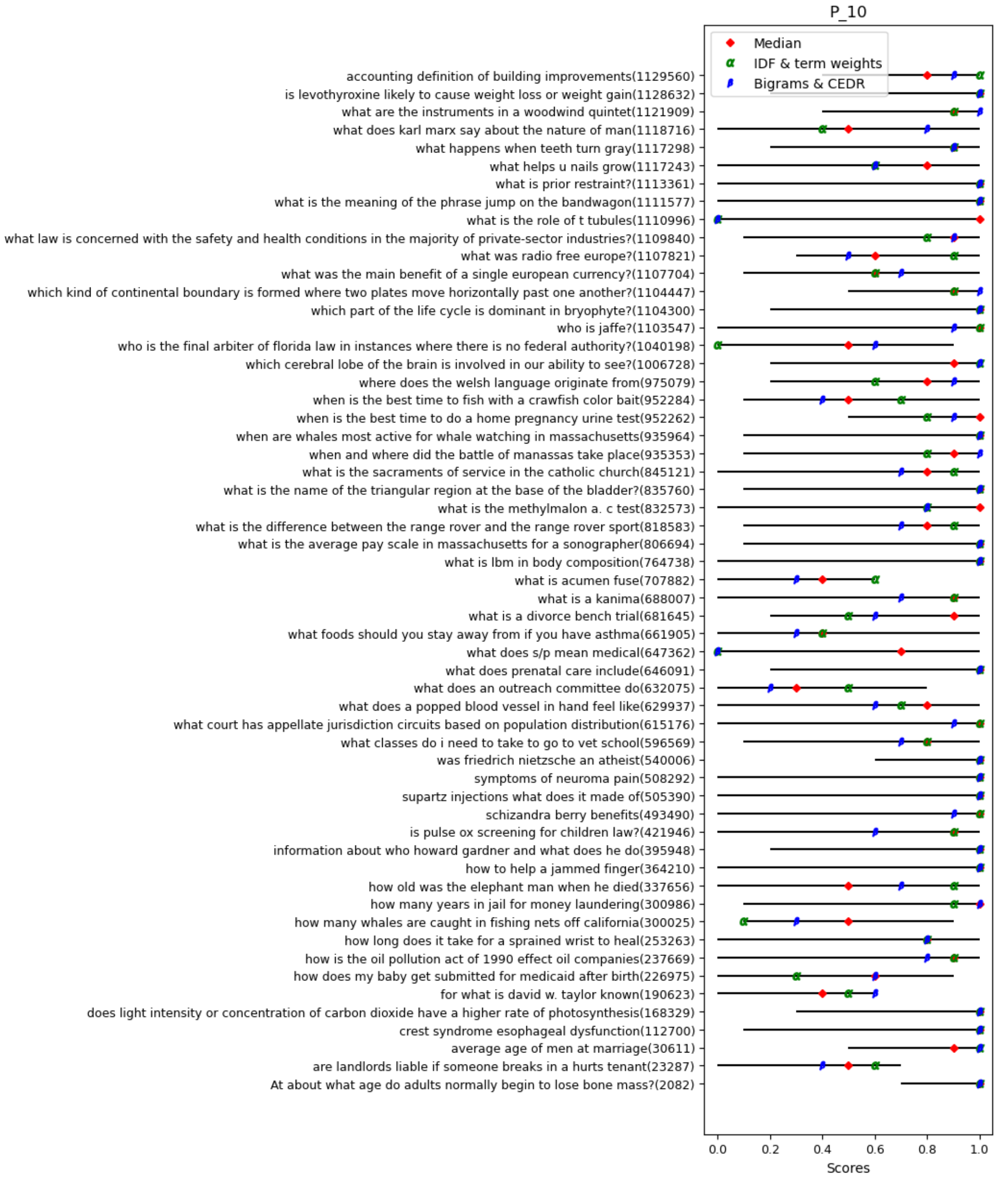

Figure 5: Precision at 10 per query

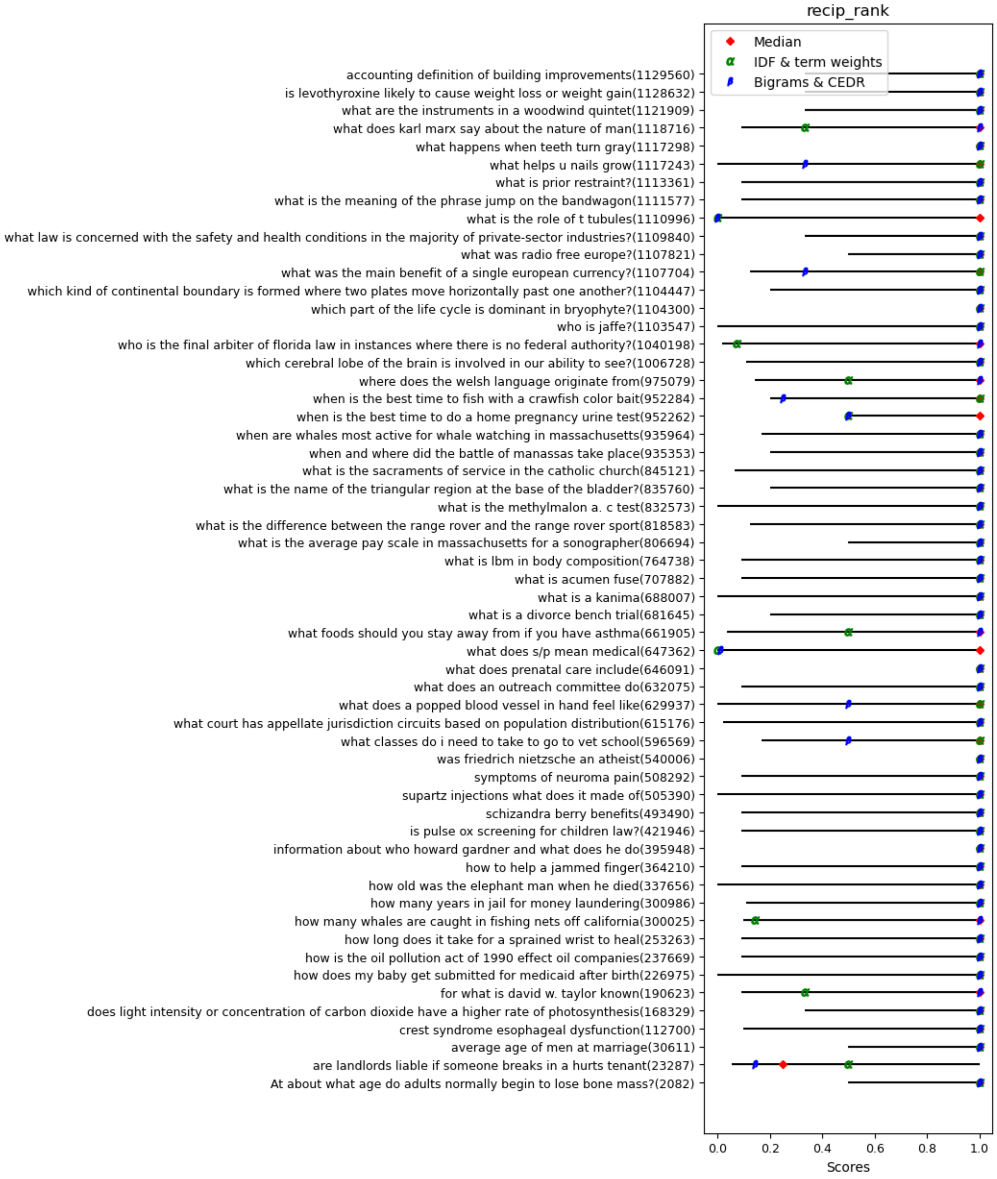

Figure 6: Reciprocal Rank per query

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), CONNEXIONs (Grant Agreement No 786731) and INFINITY (Grant Agreement No 883293).

References

[1] 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.

[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] 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.

[4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[5] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[6] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[7] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[10] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104, 2019.

[11] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. Query expansion with locally-trained word embeddings. *arXiv preprint arXiv:1605.07891*, 2016.

[12] Saar Kuzi, Anna Shtok, and Oren Kurland. Query expansion using word embeddings. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1929–1932, 2016.

[13] Dwaipayan Roy, Debjyoti Paul, Mandar Mitra, and Utpal Garain. Using word embeddings for automatic query expansion. *arXiv preprint arXiv:1606.07608*, 2016.

[14] Jeffrey Dalton, Shahrzad Naseri, Laura Dietz, and James Allan. Local and global query expansion for hierarchical complex topics. In *European Conference on Information Retrieval*, pages 290–303. Springer, 2019.

[15] Shafayet Ahmed, Abu Nowshed Chy, and Md Zia Ullah. Exploiting various word embedding models for query expansion in microblog. In *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*, pages 1–6. IEEE, 2020.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] Victor Dibia. Neuralqa: A usable library for question answering (contextual query expansion+ bert) on large datasets. *arXiv preprint arXiv:2007.15211*, 2020.

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

[23] 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.

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

[25] 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.

[26] 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.

[27] Stéphane Clinchant and Eric Gaussier. A theoretical analysis of pseudo-relevance feedback models. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, pages 6–13, 2013.

[28] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2004.

[29] Abdulaziz AlQatan, Leif Azzopardi, and Yashar Moshfeghi. Analyzing the influence of bigrams on retrieval bias and effectiveness. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, pages 157–160, 2020.