CONFORMER-KERNEL WITH QUERY TERM INDEPENDENCE AT TREC 2020 DEEP LEARNING TRACK

Bhaskar Mitra
Microsoft, University College London
bmitra@microsoft.com

Sebastian Hofstätter
TU Wien
s.hofstaetter@tuwien.ac.at

Hamed Zamani*
University of Massachusetts Amherst
zamani@cs.umass.edu

Nick Craswell
Microsoft
nickcr@microsoft.com

ABSTRACT

We benchmark Conformer-Kernel models under the strict blind evaluation setting of the TREC 2020 Deep Learning track. In particular, we study the impact of incorporating: (i) Explicit term matching to complement matching based on learned representations (*i.e.*, the "Duet principle"), (ii) query term independence (*i.e.*, the "QTI assumption") to scale the model to the full retrieval setting, and (iii) the ORCAS click data as an additional document description field. We find evidence which supports that all three aforementioned strategies can lead to improved retrieval quality.

***K*eywords** Deep learning · Neural information retrieval · Ad-hoc retrieval

1 Introduction

The Conformer-Kernel (CK) model [Mitra et al., 2020] builds upon the Transformer-Kernel (TK) [Hofstätter et al., 2019] architecture, that demonstrated strong competitive performance compared to BERT-based [Devlin et al., 2019] ranking methods, but notably at a fraction of the compute and GPU memory cost, at the TREC 2019 Deep Learning track [Craswell et al., 2020b]. Notwithstanding these strong results, the TK model suffers from two clear deficiencies. Firstly, because the TK model employs stacked Transformers for query and document encoding, it is challenging to incorporate long body text into this model as the GPU memory requirement of Transformers' self-attention layers grows quadratically with respect to input sequence length. So, for example, to increase the limit on the maximum input sequence length by $4\times$ from 128 to 512 we would require $16\times$ more GPU memory for each of the self-attention layers in the model. Considering that documents can contain thousands of terms, this limits the model to inspecting only a subset of the document text which may have negative implications, such as poorer retrieval quality and under-retrieval of longer documents [Hofstätter et al., 2020]. Secondly, the original TK model was designed for the reranking task and requires that every document in a given candidate set be evaluated individually with respect to the query. This is problematic if we want to use the model to retrieve from the full collection which may contain millions, if not billions, of documents. Zamani et al. [2018a] raised this concern for the first time and addressed it by learning sparse representations for query and documents for inverted indexing. Later, Mitra et al. [2019] proposed an alternative solution based on the query term independence (QTI) assumption, which was adopted by Mitra et al. [2020]. They replaced the Transformer layers with novel Conformer counterparts and incorporated the QTI assumption into the model design.

In their original paper, Mitra et al. [2020] compared their model to other retrieval methods, under the full retrieval setting, based on the test set from the TREC 2019 Deep Learning track [Craswell et al., 2020b] for which both the queries and relevance labels are currently available publicly. This evaluation is less stringent than participating in the official annual TREC benchmarking because: (a) it allows the experimenter to run multiple evaluations against the test set which may lead to overfitting, and (b) it uses pre-collected labels which may not cover additional relevant documents

*Work done while at Microsoft.

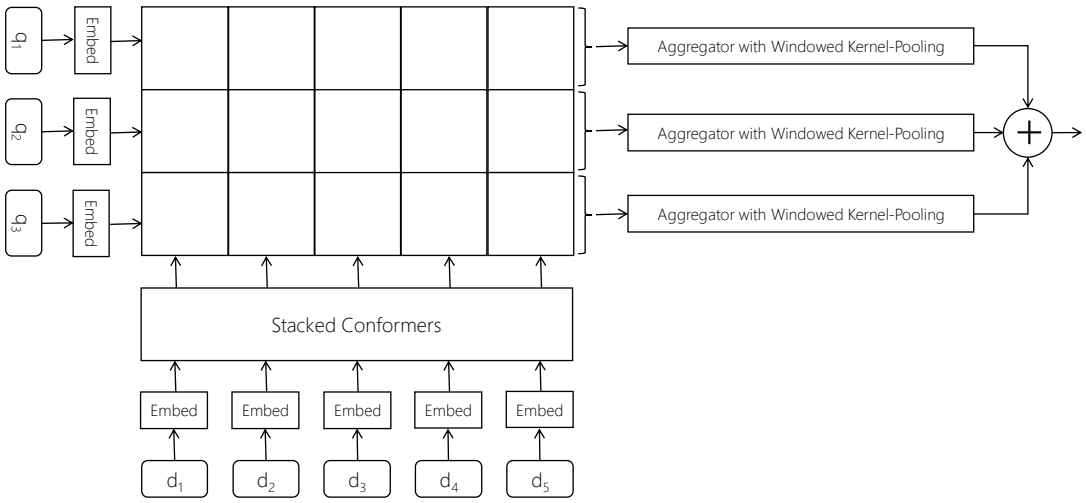


Figure 1: The NDRM1 variant of the CK model with QTI.

that a new model may surface and consequently under-report the performance of dramatically new approaches [Yilmaz et al., 2020]. Therefore, in this work, we evaluate the model under the stricter TREC benchmarking setting in the 2020 edition of the Deep Learning track [Craswell et al., 2020c].

2 TREC 2020 Deep Learning track

The TREC 2020 Deep Learning track [Craswell et al., 2020c] uses the same training data as the previous year [Craswell et al., 2020b], which was originally derived from the MS MARCO dataset [Bajaj et al., 2016]. However, the track provides a new blind test set for the second year. In our work, we only consider the document ranking task, although the track also allows participants to evaluate their models on passage ranking. The training data for the document ranking task consists of $384,597$ positively labeled query-document pairs. The test set comprised of 200 queries out of which 45 queries were selected by NIST for judging. We report four relevance metrics—NDCG@10 [Järvelin and Kekäläinen, 2002], NCG@100 [Rosset et al., 2018], AP [Zhu, 2004], and RR [Craswell, 2009]—computed over these 45 queries. Under the *rerank* setting, each model is expected to re-order a set of 100 candidate documents provided per query, and under the *fullrank* setting each model must retrieve a ranked list of maximum hundred documents from a collection containing $3,213,835$ documents in response to each query.

3 Conformer-Kernel with Query Term Independence

The CK models combine novel Conformer layers with several other existing ideas from the neural information retrieval literature [Mitra and Craswell, 2018, Guo et al., 2020]. We use the publicly available implementation[2] of CK models in our work, and adopt the same model taxonomy as in the code to describe the different variants.

The *NDRM1* variant builds on the TK architecture [Hofstätter et al., 2019] by incorporating two key changes: (i) It replaces the Transformer layers with Conformer layers, and (ii) factorizes the model to incorporate the QTI assumption. Figure 1 visualizes the NDRM1 architecture. Unlike other attempts [Hofstätter et al., 2020] at extending the TK architecture to long text by treating the document as a collection of passages, the Conformer layer replaces the standard self-attention mechanism with a separable self-attention mechanism whose memory complexity of $\mathcal{O}(n \times d_{\text{key}})$—where n is input sequence length and d_{key} is the size of the learned key embeddings—is a significant improvement over the quadratic $\mathcal{O}(n^2)$ complexity of standard self-attention. Furthermore, the Conformer layer complements the self-attention with an additional convolutional layer to more accurately model local context within the text. Next, to incorporate query term independence, the model evaluates the relevance of the document to each query term independently and then linearly combines those relevance estimates to obtain the aggregated estimate for the full query. By incorporating

[2] `https://github.com/bmitra-msft/TREC-Deep-Learning-Quick-Start`

Table 1: Official TREC results. All metrics are computed at a rank threshold of 100, unless explicitly specified.

Run description	Run ID	Subtask	NDCG@10	NCG@100	AP	RR
NDRM1	ndrm1-full	fullrank	0.5991	0.6280	0.3858	0.9333
NDRM1	ndrm1-re	rerank	0.6161	0.6283	0.4150	0.9333
NDRM3	ndrm3-re	rerank	0.6162	0.6283	0.4122	0.9333
NDRM3	ndrm3-full	fullrank	0.6162	0.6626	0.4069	0.9333
NDRM3 + ORCAS	ndrm3-orc-re	rerank	0.6217	0.6283	0.4194	0.9241
NDRM3 + ORCAS	ndrm3-orc-full	fullrank	0.6249	0.6764	0.4280	0.9444

the QTI assumption, we can precompute all term-document scores at indexing time and employ an inverted index data structure to perform fast retrieval at query time.

The *NDRM2* model can be described as a learned relevance function that only inspects the count of exact matches of query terms in the document and bears a similar form as BM25 [Robertson et al., 2009]. Similar to BM25, the NDRM2 model is also compliant with the QTI assumption. A linear combination of NDRM1 and NDRM2 gives us the NDRM3 model. This strategy of combining an exact term matching subnetwork with a representation learning based matching subnetwork has been previously studied in the context of the Duet architecture [Mitra et al., 2017, Mitra and Craswell, 2019a, Nanni et al., 2017, Mitra and Craswell, 2019b], and have been reported to be specifically effective under the full retrieval setting [Mitra et al., 2016, 2020, Kuzi et al., 2020, Gao et al., 2020, Wrzalik and Krechel, 2020]. Because of the limit on the number of run submission to TREC, we only evaluate the NDRM1 and NDRM3 models in this work, although we have confirmed on the TREC 2019 test set that the NDRM2 model is competitive with a well-tuned BM25 baseline.

For the second edition of the TREC Deep Learning track, participants were also provided a click log dataset called ORCAS [Craswell et al., 2020a] that can be used in any way the participants deem appropriate. We use clicked queries in the ORCAS data as additional meta description for the corresponding documents to complement the intrinsic document content in the form of URL, title, and body text. While previous work [Zamani et al., 2018b] have explored using fielded document input representation in the context of deep neural ranking models, in this work we simply concatenate the text from different fields to produce a flat unstructured input representation of the document that is fed into the model.

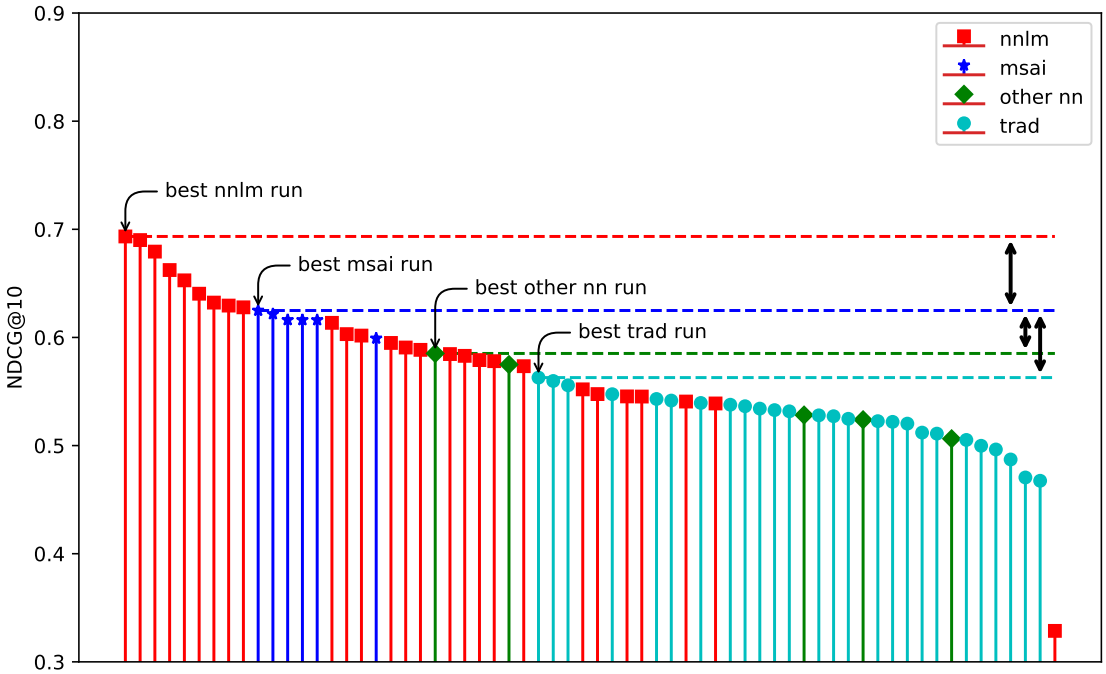
We test each model variant under both the rerank and the fullrank settings of the document ranking task in the Deep Learning track. We use the same hyperparameters and other configuration settings as prescribed by Mitra et al. [2020].

4 Results

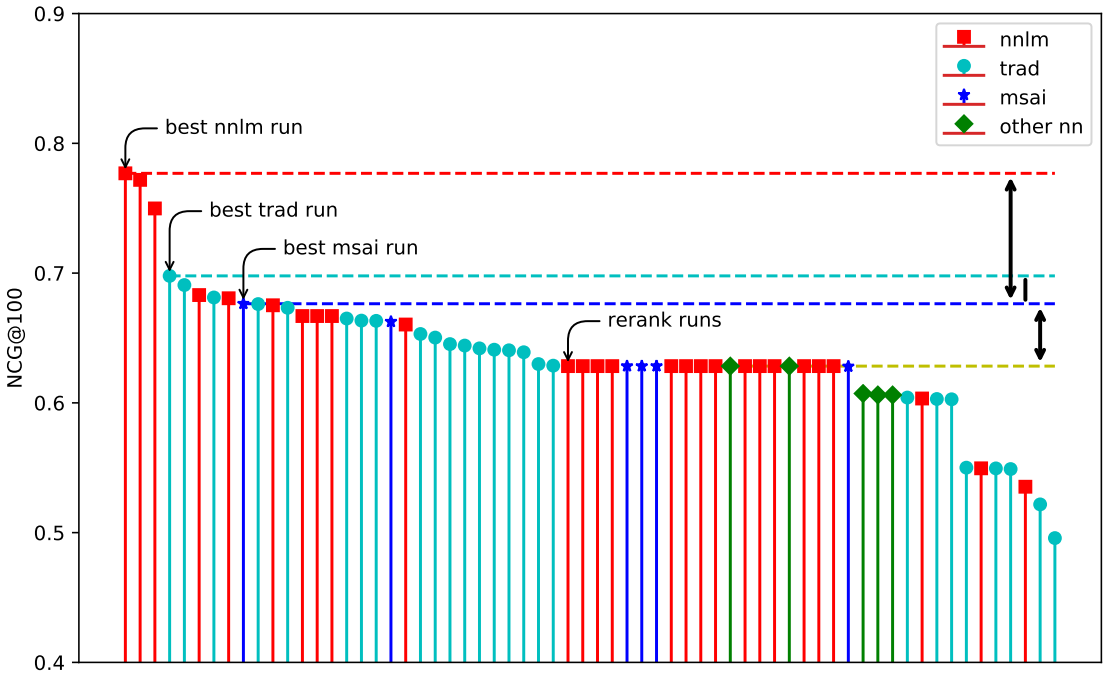
Table 1 summarizes the relevance metrics corresponding to all the submitted runs. According to the taxonomy proposed by Craswell et al. [2020b], the CK models can be described as "nn" models—*i.e.*, neural models without large scale pretraining as has been popularized by models like BERT [Devlin et al., 2019]. We do not know exactly how these models perform relative to runs from other groups until all the evaluation numbers are made public after the TREC conference. However, we do have the median per-query NDCG@10 information across all submissions to the track. Figure 3 shows the per-query performance of our best and worst performing runs compared to the median performance. We note that this figure suggests that the CK models achieve competitive retrieval quality while, much like the TK model, it requires significantly less resources to train and evaluate compared to BERT-based rankers.

In keeping with the typical TREC tradition of mainly focusing on comparing runs within groups, we focus our study on three specific research questions.

RQ1. Does explicit term matching improve retrieval quality? To shed light on this question, we compare the NDRM1 and the NDRM3 models, where the only difference between the two models is that the latter incorporates the explicit term matching signal while former does not. We find that under the reranking setting—*i.e.*, when comparing the "ndrm1-re" and the "ndrm3-re" runs—there is no clear evidence that the explicit term matching is beneficial. This is likely because the candidate documents for reranking were generated by a first-stage BM25 ranker and hence the explicit term matching signal is already part of the end-to-end retrieval stack. However, under the fullrank setting—*i.e.*, when comparing the "ndrm1-full" and the "ndrm3-full" runs—we see moderate improvements across all metrics: 2.9% improvement in NDCG@10 and 5.5% improvement in both AP and NCG@100. These observations are supported by Kuzi et al. [2020], who find that exact term matching are important for the fullrank setting, and also by Xiong et al. [2020] who observe that their proposed model which does not incorporate exact matching fare better in the rerank setting than on the fullrank subtask.



(a) NDCG@10



(b) NCG@100

Figure 2: Comparing our runs with runs submitted by other groups. We adopt the same "nnlm", "nn", and "trad" taxonomy for models as in the track overview [Craswell et al., 2020c]. All our runs are "nn" runs under this classification but we label them specifically as "msai" to distinguish from "other nn runs". The runs in each plot are sorted independently based on the corresponding metric.

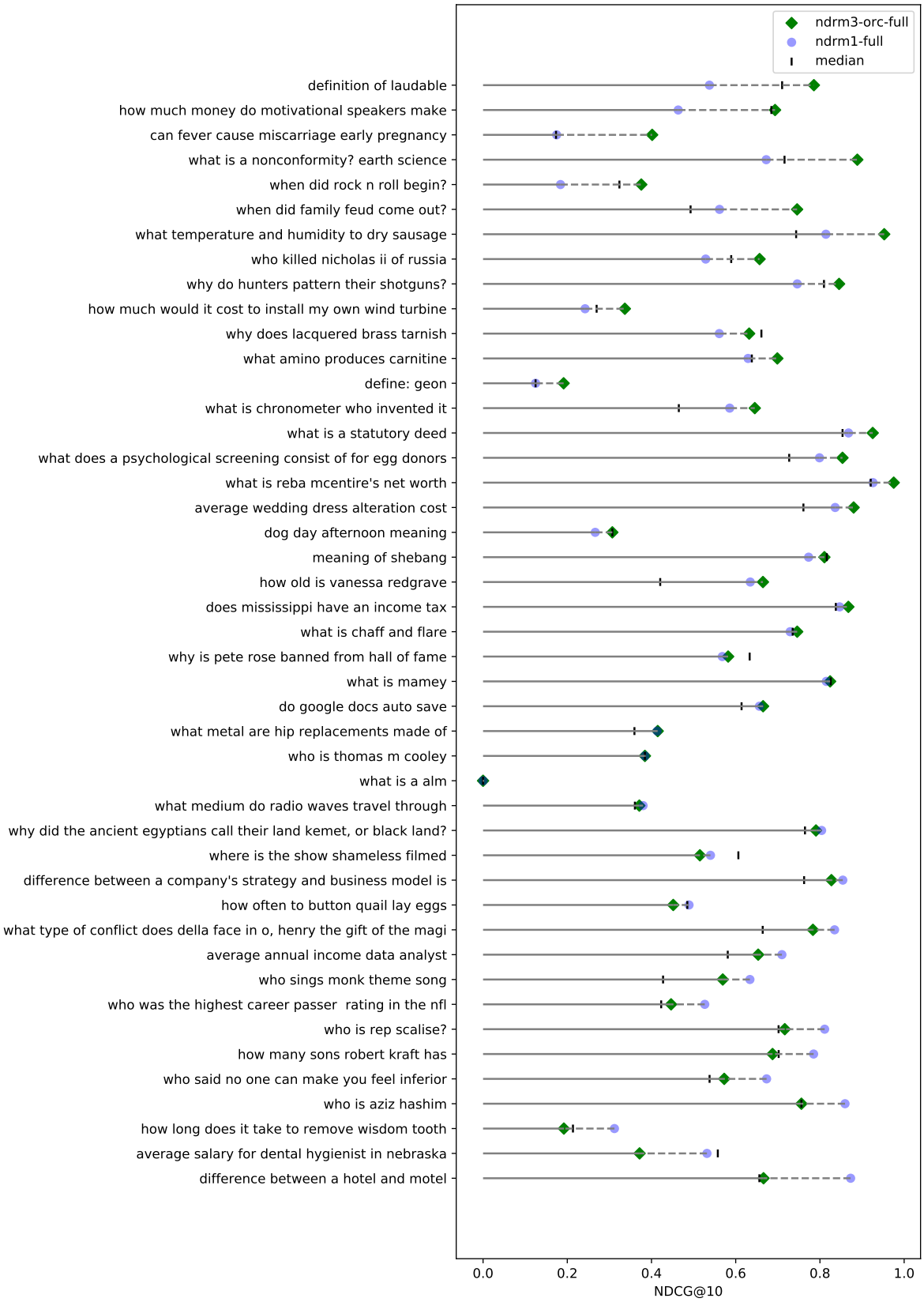


Figure 3: Per-query comparison between our worst performing run ("ndrm1-full") and our best performing run ("ndrm3-orc-full") based on the NDCG@10 metric. Median NDCG@10 across all track submissions also shown for reference.

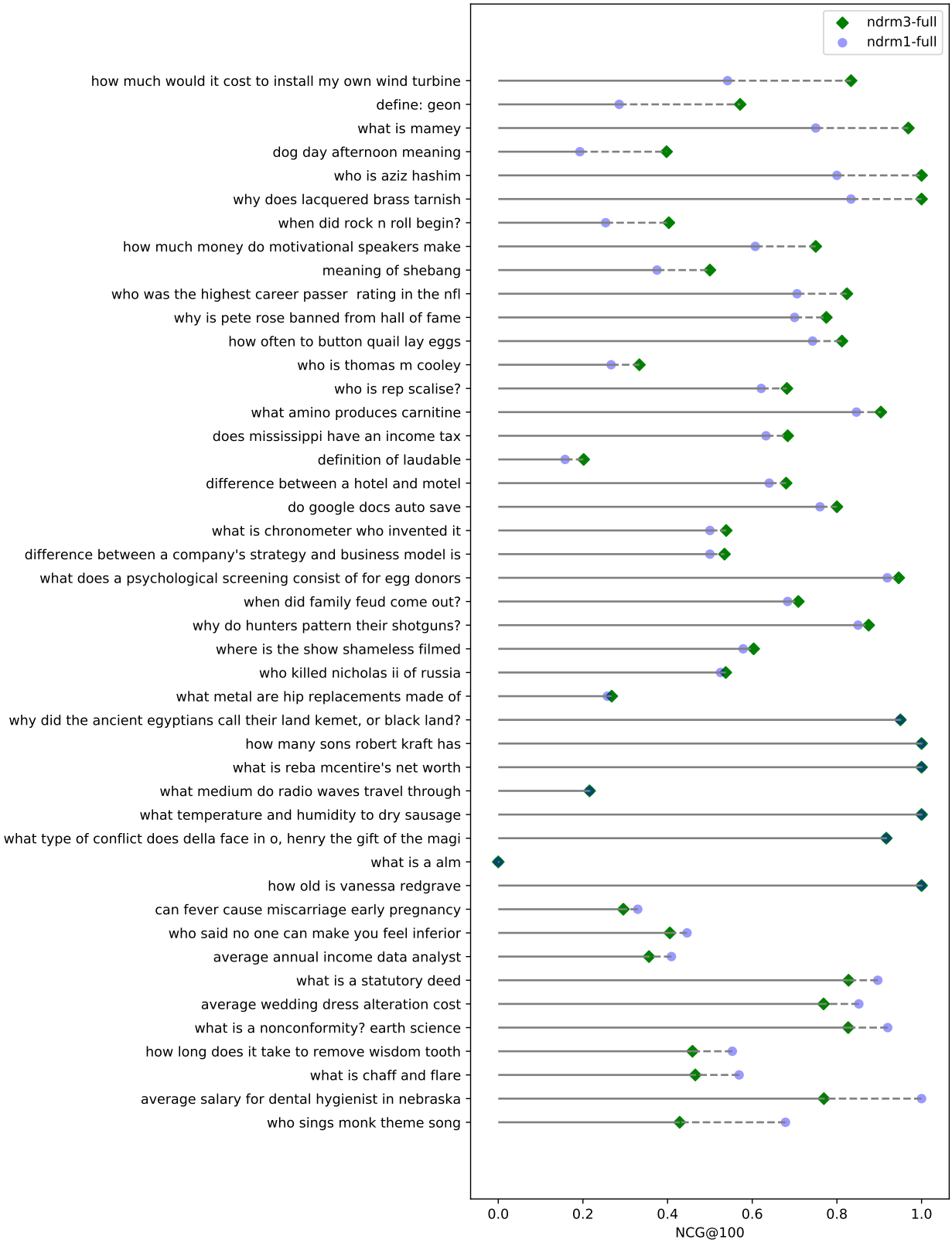


Figure 4: Per-query comparison between the “ndrm1-full” and the “ndrm3-full” runs based on the NCG@100 metric.

Figure 4 compares how the "ndrm1-full" and the "ndrm3-full" runs perform on the 45 different queries in the test set. Based on a qualitative inspection of the queries, it appears that exact term matching may be important for queries containing named entities—*e.g.*, "who is *aziz hashim*" and "why is *pete rose* banned from hall of fame"—where it is necessary to ensure that the retrieved documents are about the correct entity.

RQ2. How does the retrieval quality differ for our model between the fullrank and the rerank setting? As expected, we find that without exact term matching, the retrieval quality for CK models are lower under the fullrank setting compared to the rerank setting—*i.e.*, "ndrm1-re" is better than "ndrm1-full". In contrast, when exact term matching is incorporated, the CK model achieves 5.5% improvement in NCG, which is a recall-oriented metric, in the fullrank setting ("ndrm3-full") compared to its counterpart under the rerank setting ("ndrm3-re"). However, on all the other metrics we see no difference (NDCG@10 and RR) or small regression (1.3% for AP) under the fullrank setting. Finally, if we introduce the ORCAS data—*i.e.*, compare "ndrm3-orc-full" and "ndrm3-orc-re"—we see improvements under the fullrank setting across all metrics: 7.7% for NCG@100, 2.2% for RR, 2.1% for AP, and 0.5% for NDCG@10.

In adhoc retrieval, a common strategy involves sequentially cascading multiple rank-and-prune stages [Matveeva et al., 2006, Wang et al., 2011, Chen et al., 2017, Gallagher et al., 2019, Nogueira et al., 2019] for better effectiveness-efficiency trade-offs. Following a similar strategy, we may be able to improve on these results by introducing additional reranking stages on top of a first stage retrieval using query term independent CK models. We anticipate that this may be an interesting area for future exploration.

RQ3. Does using ORCAS queries as an additional document description field improve retrieval quality? Finally, we want to study if the incorporation of click log datasets, such as ORCAS [Craswell et al., 2020a], can be beneficial for retrieval quality. We find that on the rerank subtask, both NDCG@10 and AP improve by 0.9% and 1.7%, respectively, although RR degrades by 1%. On the fullrank subtask, the addition of ORCAS signal seems to improve all metrics: AP by 5.2%, NCG@100 by 2.1%, NDCG@10 by 1.4%, and RR by 1.2%. These results indicate that ORCAS, and other similar click log datasets, may be useful for achieving better retrieval relevance.

5 Conclusion

In this work, we benchmark CK models under the strict blind evaluation setting of the TREC 2020 Deep Learning track. We find that incorporating (i) exact term matching (the "Duet principle"), (ii) query term independence (the "QTI assumption"), and (iii) ORCAS data as an additional document field all generally contribute positively to retrieval quality, and the run "ndrm3-orc-full" that incorporates all three techniques achieves our best performance. We posit that considering the significantly lower cost of training and evaluating CK models, these models provide interesting alternatives to BERT-based rankers with different operating points on the effectiveness-efficiency curve.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Ruey-Cheng Chen, Luke Gallagher, Roi Blanco, and J Shane Culpepper. Efficient Cost-Aware Cascade Ranking in Multi-Stage Retrieval. In *Proc. of SIGIR*, 2017.

Nick Craswell. Mean Reciprocal Rank. In *Encyclopedia of Database Systems*, pages 1703–1703. Springer, 2009.

Nick Craswell, Daniel Campos, Bhaskar Mitra, Emine Yilmaz, and Bodo Billerbeck. ORCAS: 18 Million Clicked Query-Document Pairs for Analyzing Search. In *Proc. of CIKM*, 2020a.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2019 Deep Learning Track. In *Proc. of TREC*, 2020b.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the TREC 2020 Deep Learning Track. In *Proc. of TREC (to be published)*, 2020c.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL*, 2019.

Luke Gallagher, Ruey-Cheng Chen, Roi Blanco, and J Shane Culpepper. Joint Optimization of Cascade Ranking Models. In *Proc. of WSDM*, 2019.

Luyu Gao, Zhuyun Dai, Zhen Fan, and Jamie Callan. Complementing Lexical Retrieval with Semantic Residual Embedding. *arXiv preprint arXiv:2004.13969*, 2020.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. A Deep Look into Neural Ranking Models for Information Retrieval. *IP&M*, 2020.

Sebastian Hofstätter, Markus Zlabinger, and Allan Hanbury. TU Wien@ TREC Deep Learning'19–Simple Contextualization for Re-ranking. In *Proc. of TREC*, 2019.

Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. Local Self-Attention over Long Text for Efficient Document Retrieval. In *Proc. of SIGIR*. ACM, 2020.

K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM TOIS*, 20(4):422–446, 2002.

Saar Kuzi, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. Leveraging Semantic and Lexical Matching to Improve the Recall of Document Retrieval Systems: A Hybrid Approach. *arXiv preprint arXiv:2010.01195*, 2020.

Irina Matveeva, Chris Burges, Timo Burkard, Andy Laucius, and Leon Wong. High Accuracy Retrieval with Multiple Nested Ranker. In *Proc. of SIGIR*, 2006.

Bhaskar Mitra and Nick Craswell. An Introduction to Neural Information Retrieval. *Foundations and Trends® in Information Retrieval*, 2018.

Bhaskar Mitra and Nick Craswell. Duet at TREC 2019 Deep Learning Track. 2019a.

Bhaskar Mitra and Nick Craswell. An Updated Duet Model for Passage Re-ranking. *arXiv preprint arXiv:1903.07666*, 2019b.

Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. A Dual Embedding Space Model for Document Ranking. *arXiv preprint arXiv:1602.01137*, 2016.

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

Bhaskar Mitra, Corby Rosset, David Hawking, Nick Craswell, Fernando Diaz, and Emine Yilmaz. Incorporating Query Term Independence Assumption for Efficient Retrieval and Ranking using Deep Neural Networks. *arXiv preprint arXiv:1907.03693*, 2019.

Bhaskar Mitra, Sebastian Hofstatter, Hamed Zamani, and Nick Craswell. Conformer-Kernel with Query Term Independence for Document Retrieval. *arXiv preprint arXiv:2007.10434*, 2020.

Federico Nanni, Bhaskar Mitra, Matt Magnusson, and Laura Dietz. Benchmark for Complex Answer Retrieval. In *Proc. ICTIR*. ACM, 2017.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-Stage Document Ranking with BERT. *arXiv preprint arXiv:1910.14424*, 2019.

Stephen Robertson, Hugo Zaragoza, et al. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Corby Rosset, Damien Jose, Gargi Ghosh, Bhaskar Mitra, and Saurabh Tiwary. Optimizing Query Evaluations using Reinforcement Learning for Web Search. In *Proc. of SIGIR*, 2018.

Lidan Wang, Jimmy Lin, and Donald Metzler. A Cascade Ranking Model for Efficient Ranked Retrieval. In *Proc. of SIGIR*, 2011.

Marco Wrzalik and Dirk Krechel. CoRT: Complementary Rankings from Transformers. *arXiv preprint arXiv:2010.10252*, 2020.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. *arXiv preprint arXiv:2007.00808*, 2020.

Emine Yilmaz, Nick Craswell, Bhaskar Mitra, and Daniel Campos. On the Reliability of Test Collections for Evaluating Systems of Different Types. In *Proc. of SIGIR*, 2020.

Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In *Proc. of CIKM*, 2018a.

Hamed Zamani, Bhaskar Mitra, Xia Song, Nick Craswell, and Saurabh Tiwary. Neural Ranking Models with Multiple Document Fields. In *Proc. of WSDM*, 2018b.

Mu Zhu. Recall, Precision and Average Precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2:30, 2004.