

# Document Expansion by Query Prediction

Rodrigo Nogueira,<sup>1</sup> Wei Yang,<sup>2</sup> Jimmy Lin,<sup>2</sup> and Kyunghyun Cho<sup>3,4,5,6</sup>

<sup>1</sup> Tandon School of Engineering, New York University

<sup>2</sup> David R. Cheriton School of Computer Science, University of Waterloo

<sup>3</sup> Courant Institute of Mathematical Sciences, New York University

<sup>4</sup> Center for Data Science, New York University

<sup>5</sup> Facebook AI Research <sup>6</sup> CIFAR Azrieli Global Scholar

## Abstract

One technique to improve the retrieval effectiveness of a search engine is to expand documents with terms that are related or representative of the documents' content. From the perspective of a question answering system, this might comprise questions the document can potentially answer. Following this observation, we propose a simple method that predicts which queries will be issued for a given document and then expands it with those predictions with a vanilla sequence-to-sequence model, trained using datasets consisting of pairs of query and relevant documents. By combining our method with a highly-effective re-ranking component, we achieve the state of the art in two retrieval tasks. In a latency-critical regime, retrieval results alone (without re-ranking) approach the effectiveness of more computationally expensive neural re-rankers but are much faster.

Code to reproduce experiments and trained models can be found at <https://github.com/nyu-dl/dl4ir-doc2query>.

## 1 Introduction

The “vocabulary mismatch” problem, where users use query terms that differ from those used in relevant documents, is one of the central challenges in information retrieval. Prior to the advent of neural retrieval models, this problem has most often been tackled using query expansion techniques, where an initial round of retrieval can provide useful terms to augment the original query. Continuous vector space representations and neural networks, however, no longer depend on discrete one-hot representations, and thus offer an exciting new approach to tackling this challenge.

Despite the potential of neural models to match documents at the semantic level for improved ranking, most scalable search engines use exact

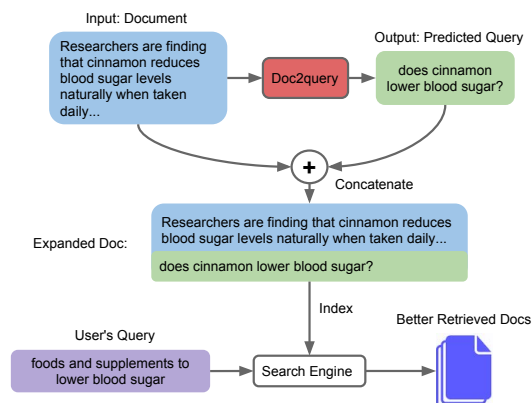


Figure 1: Given a document, our Doc2query model predicts a query, which is appended to the document. Expansion is applied to all documents in the corpus, which are then indexed and searched as before.

term match between queries and documents to perform initial retrieval. Query expansion is about enriching the query representation while holding the document representation static. In this paper, we explore an alternative approach based on enriching the document representation (prior to indexing). Focusing on question answering, we train a sequence-to-sequence model, that given a document, generates possible questions that the document might answer. An overview of the proposed method is shown in Figure 1.

We view this work as having several contributions: This is the first successful application of document expansion using neural networks that we are aware of. On the recent MS MARCO dataset (Bajaj et al., 2016), our approach is competitive with the best results on the official leaderboard, and we report the best-known results on TREC CAR (Dietz et al., 2017). We further show that document expansion is more effective than query expansion on these two datasets. We accomplish this with relatively simple models using existing open-source toolkits, which allows easy replication of our results. Document expansion

also presents another major advantage, since the enrichment is performed prior to indexing: Although retrieved output can be further re-ranked using a neural model to greatly enhance effectiveness, the output can also be returned as-is. These results already yield a noticeable improvement in effectiveness over a “bag of words” baseline without the need to apply expensive and slow neural network inference at retrieval time.

## 2 Related Work

Prior to the advent of continuous vector space representations and neural ranking models, information retrieval techniques were mostly limited to keyword matching (i.e., “one-hot” representations). Alternatives such as latent semantic indexing (Deerwester et al., 1990) and its various successors never really gained significant traction. Approaches to tackling the vocabulary mismatch problem within these constraints include relevance feedback (Rocchio, 1971), query expansion (Voorhees, 1994; Xu and Croft, 2000), and modeling term relationships using statistical translation (Berger and Lafferty, 1999). These techniques share in their focus on enhancing *query* representations to better match documents.

In this work, we adopt the alternative approach of enriching *document* representations (Tao et al., 2006; Pickens et al., 2010; Efron et al., 2012), which works particularly well for speech (Singhal and Pereira, 1999) and multi-lingual retrieval, where terms are noisy. Document expansion techniques have been less popular with IR researchers because they are less amenable to rapid experimentation. The corpus needs to be re-indexed every time the expansion technique changes (typically, a costly process); in contrast, manipulations to query representations can happen at retrieval time (and hence are much faster). The success of document expansion has also been mixed; for example, Billerbeck and Zobel (2005) explore both query expansion and document expansion in the same framework and conclude that the former is consistently more effective.

A new generation of neural ranking models offer solutions to the vocabulary mismatch problem based on continuous word representations and the ability to learn highly non-linear models of relevance; see recent overviews by Onal et al. (2018) and Mitra and Craswell (2019a). However, due to the size of most corpora and the impractical-

ity of applying inference over every document in response to a query, nearly all implementations today deploy neural networks as re-rankers over initial candidate sets retrieved using standard inverted indexes and a term-based ranking model such as BM25 (Robertson et al., 1994). Our work fits into this broad approach, where we take advantage of neural networks to augment document representations prior to indexing; term-based retrieval then happens exactly as before. Of course, retrieved results can still be re-ranked by a state-of-the-art neural model (Nogueira and Cho, 2019), but the output of term-based ranking already appears to be quite good. In other words, our document expansion approach can leverage neural networks without their high inference-time costs.

## 3 Method: Doc2query

Our proposed method, which we call “Doc2query”, proceeds as follows: For each document, the task is to predict a set of queries for which that document will be relevant. Given a dataset of (query, relevant document) pairs, we use a sequence-to-sequence transformer model (Vaswani et al., 2017) that takes as an input the document terms and produces a query. The document and target query are segmented using BPE (Sennrich et al., 2015) after being tokenized with the Moses tokenizer.<sup>1</sup> To avoid excessive memory usage, we truncate each document to 400 tokens and queries to 100 tokens. Architecture and training details of our transformer model are described in Appendix A.

Once the model is trained, we predict 10 queries using top-*k* random sampling (Fan et al., 2018) and append them to each document in the corpus. We do not put any special markup to distinguish the original document text from the predicted queries. The expanded documents are indexed, and we retrieve a ranked list of documents for each query using BM25 (Robertson et al., 1994). We optionally re-rank these retrieved documents using BERT (Devlin et al., 2018) as described by Nogueira and Cho (2019).

## 4 Experimental Setup

To train and evaluate the models, we use the following two datasets:

**MS MARCO** (Bajaj et al., 2016) is a passage

<sup>1</sup> <http://www.statmt.org/moses/>

	TREC-CAR MAP Test	MS MARCO MRR@10 Test Dev	Retrieval Time ms/query
Single Duet v2 (Mitra and Craswell, 2019b)	-	24.5 24.3	650*
Co-PACRR <sup>♠</sup> (MacAvaney et al., 2017)	14.8	- -	-
BM25	15.3	18.6 18.4	50
BM25 + RM3	12.7	- 16.7	250
BM25 + Doc2query (Ours)	18.3	21.8 21.5	90
BM25 + Doc2query + RM3 (Ours)	15.5	- 20.0	350
BM25 + BERT (Nogueira and Cho, 2019)	34.8	35.9 36.5	3400 <sup>†</sup>
BM25 + Doc2query + BERT (Ours)	<b>36.5</b>	<b>36.8 37.5</b>	3500 <sup>†</sup>

Table 1: Main results on TREC-CAR and MS MARCO datasets. \* Our measurement, in which Duet v2 takes 600ms per query, and BM25 retrieval takes 50ms. <sup>♠</sup> Best submission of TREC-CAR 2017. <sup>†</sup> We use Google’s TPUs to re-rank with BERT.

**re-ranking dataset** with 8.8M passages<sup>2</sup> obtained from the top-10 results retrieved by the Bing search engine (from 1M queries). The training set contains approximately 500k pairs of query and relevant documents. Each query has one relevant passage, on average. The development and test sets contain approximately 6,900 queries each, but relevance labels are made public only for the development set.

**TREC-CAR** (Dietz et al., 2017) is a dataset where the input query is the concatenation of a Wikipedia article title with the title of one of its sections. The ground-truth documents are the paragraphs within that section. The corpus consists of all English Wikipedia paragraphs except the abstracts. The released dataset has five predefined folds, and we use the first four as a training set (approx. 3M queries), and the remaining as a validation set (approx. 700k queries). The test set is the same as the one used to evaluate submissions to TREC-CAR 2017 (approx. 2,250 queries).

We evaluate the following ranking methods:

**BM25:** We use the Anserini open-source IR toolkit (Yang et al., 2017, 2018)<sup>3</sup> to index the original (non-expanded) documents and BM25 to rank the passages. During evaluation, we use the top-1000 re-ranked passages.

**BM25 + Doc2query:** We first expand the documents using the proposed Doc2query method. We then index and rank the expanded documents exactly as in the BM25 method above.

**RM3:** To compare document expansion with query expansion, we applied the RM3 query expansion technique (Abdul-Jaleel et al., 2004). We apply query expansion to both unexpanded documents (BM25 + RM3) as well as the expanded documents (BM25 + Doc2query + RM3).

**BM25 + BERT:** We index and retrieve documents as in the BM25 condition and further re-rank the documents with BERT as described in Nogueira and Cho (2019).

**BM25 + Doc2query + BERT:** We expand, index, and retrieve documents as in the BM25 + Doc2query condition and further re-rank the documents with BERT.

To evaluate the effectiveness of the methods on MS MARCO, we use its official metric, mean reciprocal rank of the top-10 documents (MRR@10). For TREC-CAR, we use mean average precision (MAP).

## 5 Results

Results on both datasets are shown in Table 1. BM25 is the baseline. Document expansion with our method (BM25 + Doc2query) improves retrieval effectiveness by  $\sim 15\%$  for both datasets. When we combine document expansion with a state-of-the-art re-ranker (BM25 + Doc2query + BERT), we achieve the best-known results to date on TREC CAR; for MS MARCO, we are near the state of the art.<sup>4</sup> Our full re-ranking condition (BM25 + Doc2query + BERT) beats BM25 + BERT alone, which verifies that the contribution

<sup>2</sup> <https://github.com/dfcf93/MSMARCO/tree/master/Ranking>

<sup>3</sup> <http://anserini.io/>

<sup>4</sup> The top leaderboard entries do not come with system descriptions, and so it is not possible to compare our approach with theirs.

Input Document:	July is the hottest month in Washington DC with an average temperature of 27C (80F) and the coldest is January at 4C (38F) with the most daily sunshine hours at 9 in July. The wettest month is May with an average of 100mm of rain.
Predicted Query:	weather in washington dc
Target query:	what is the temperature in washington
Input Document:	The Delaware River flows through Philadelphia into the Delaware Bay. It flows through and aqueduct in the Roundout Reservoir and then flows through Philadelphia and New Jersey before emptying into the Delaware Bay.
Predicted Query:	what river flows through delaware
Target Query:	where does the delaware river start and end
Input Document:	sex chromosome - (genetics) a chromosome that determines the sex of an individual; mammals normally have two sex chromosomes chromosome - a threadlike strand of DNA in the cell nucleus that carries the genes in a linear order; humans have 22 chromosome pairs plus two sex chromosomes.
Predicted Query:	what is the relationship between genes and chromosomes
Target Query:	which chromosome controls sex characteristics

Table 2: Examples of query predictions on MS MARCO compared to real user queries.

of Doc2query is indeed orthogonal to that from post-indexing re-ranking.

Where exactly are these better scores coming from? We show in Table 2 examples of queries produced by our Doc2query model trained on MS MARCO. We notice that the model tends to copy some words from the input document (e.g., Washington DC, River, chromosome), meaning that it can effectively perform term re-weighting (i.e., increasing the importance of key terms). Nevertheless, the model also produces words not present in the input document (e.g., weather, relationship), which can be characterized as expansion by synonyms and other related terms.

To quantify this analysis, we measured the proportion of predicted words that exist (copied) vs. not-exist (new) in the original document. Excluding stop words, which corresponds to 51% of the predicted query words, we found that 31% are new while the rest (69%) are copied. If we expand MS MARCO documents using only new words and retrieve the development set queries with BM25, we obtain an MRR@10 of 18.8 (as opposed to 18.4 when indexing with original documents). Expanding with copied words gives an MRR@10 of 19.7. We achieve a higher MRR@10 of 21.5 when documents are expanded with both types of words, showing that they are complementary.

Further analyses show that one source of improvement comes from having more relevant documents for the re-ranker to consider. We find that the Recall@1000 of the MS MARCO development set increased from 85.3 (BM25) to 89.3 (BM25 + Doc2query). Results show that BERT is indeed able to identify these correct answers from the improved candidate pool and bring them to the

top of the ranked list, thus improving the overall MRR.

As a contrastive condition, we find that query expansion with RM3 hurts in both datasets, whether applied to the unexpanded corpus (BM25 + RM3) or the expanded version (BM25 + Doc2query + RM3). This is a somewhat surprising result because query expansion usually improves effectiveness in document retrieval, but this can likely be explained by the fact that both MS MARCO and CAR are precision oriented. This result shows that document expansion can be more effective than query expansion, most likely because there are more signals to exploit as documents are much longer.

Finally, for production retrieval systems, latency is often an important factor. Our method without a re-ranker (BM25 + Doc2query) adds a small latency increase over baseline BM25 (50 ms vs. 90 ms) but is approximately seven times faster than a neural re-ranker that has a three points higher MRR@10 (Single Duet v2, which is presented as a baseline in MS MARCO by the organizers). For certain operating scenarios, this trade-off in quality for speed might be worthwhile.

## 6 Conclusion

We present the first successful use of document expansion based on neural networks. Document expansion holds substantial promise for neural models because documents are much longer and thus contain richer input signals. Furthermore, the general approach allows developers to shift the computational costs of neural network inference from retrieval to indexing.

Our implementation is based on integrating



three open-source toolkits: OpenNMT (Klein et al., 2017), Anserini, and TensorFlow BERT. The relative simplicity of our approach aids in the reproducibility of our results and paves the way for further improvements in document expansion.

## Acknowledgments

KC thanks support by NVIDIA and CIFAR and was partly supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI) and Samsung Electronics (Improving Deep Learning using Latent Structure). JL thanks support by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

## References

- Nasreen Abdul-Jaleel, James Allan, W. Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Donald Metzler, Mark D. Smucker, Trevor Strohman, Howard Turtle, and Courtney Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004)*. Gaithersburg, Maryland.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2016. MS MARCO: A Human Generated MACHine Reading COMprehension Dataset. *arXiv:1611.09268* (2016).
- Adam Berger and John Lafferty. 1999. Information Retrieval as Statistical Translation. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*. 222–229.
- Bodo Billerbeck and Justin Zobel. 2005. Document Expansion versus Query Expansion for Ad-hoc Retrieval. In *Proceedings of the 10th Australasian Document Computing Symposium*. 34–41.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the Association for Information Science* 41, 6 (1990), 391–407.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805* (2018).
- Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. 2017. TREC Complex Answer Retrieval Overview. In *Proceedings of the Twenty-Sixth Text REtrieval Conference (TREC 2017)*.
- Miles Efron, Peter Organisciak, and Katrina Fenlon. 2012. Improving Retrieval of Short Texts Through Document Expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2012)*. 911–920.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. *arXiv:1805.04833* (2018).
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980* (2014).
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proc. ACL*. <https://doi.org/10.18653/v1/P17-4012>
- Sean MacAvaney, Andrew Yates, and Kai Hui. 2017. Contextualized PACRR for complex answer retrieval. In *Proceedings of TREC*.
- Bhaskar Mitra and Nick Craswell. 2019a. An Introduction to Neural Information Retrieval. *Foundations and Trends in Information Retrieval* 13, 1 (2019), 1–126.
- Bhaskar Mitra and Nick Craswell. 2019b. An Updated Duet Model for Passage Re-ranking. *arXiv:1903.07666* (2019).
- Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv:1901.04085* (2019).
- Kezban Dilek Onal, Ye Zhang, Ismail Sengor Altıngöve, Md Mustafizur Rahman, Pinar Karagoz, Alex Braylan, Brandon Dang, Heng-Lu Chang, Henna Kim, Quinten Mcnamara, Aaron Angert, Edward Banner, Vivek Khetan, Tyler McDonnell, An Thanh Nguyen, Dan Xu, Byron C. Wallace, Maarten Rijke, and Matthew Lease. 2018. Neural Information Retrieval: At the End of the Early Years. *Information Retrieval* 21, 2-3 (June 2018), 111–182.
- Jeremy Pickens, Matthew Cooper, and Gene Golovchinsky. 2010. Reverted Indexing for Feedback and Expansion. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010)*. 1049–1058.
- Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*. Gaithersburg, Maryland, 109–126.
- Joseph John Rocchio. 1971. Relevance Feedback in Information Retrieval. In *The SMART Retrieval System—Experiments in Automatic Document Processing*, Gerard Salton (Ed.). Prentice-Hall, Englewood Cliffs, New Jersey, 313–323.

- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural Machine Translation of Rare Words with Subword Units. *arXiv:1508.07909* (2015).
- Amit Singhal and Fernando Pereira. 1999. Document Expansion for Speech Retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*. 34–41.
- Tao Tao, Xuanhui Wang, Qiaozhu Mei, and ChengXiang Zhai. 2006. Language Model Information Retrieval with Document Expansion. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. 407–414.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*. 5998–6008.
- Ellen M. Voorhees. 1994. Query Expansion Using Lexical-Semantic Relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994)*. 61–69.
- Jinxi Xu and W. Bruce Croft. 2000. Improving the Effectiveness of Information Retrieval with Local Context Analysis. *ACM Transactions on Information Systems* 18, 1 (2000), 79–112.
- Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017)*. 1253–1256.
- Peilin Yang, Hui Fang, and Jimmy Lin. 2018. Anserini: Reproducible Ranking Baselines Using Lucene. *Journal of Data and Information Quality* 10, 4 (2018), Article 16.

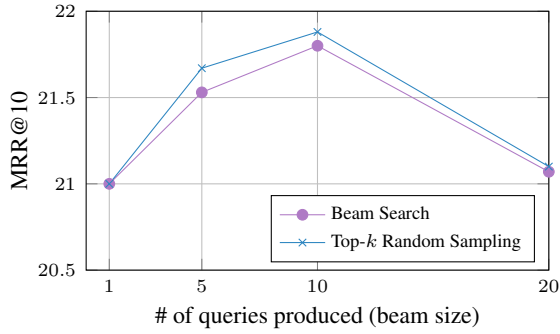


Figure 2: Retrieval effectiveness on the development set of MS MARCO when using different decoding methods to produce queries. On the *x*-axis, we vary the number of predicted queries that are appended to the original documents.

## Appendix A Architecture and Training Details

The architecture of our transformer model is identical to the *base* model described in Vaswani et al. (2017), which has 6 layers for both encoder and decoder, 512 hidden units in each layer, 8 attention heads and 2048 hidden units in the feed-forward layers. We train with a batch size of 4096 tokens for a maximum of 30 epochs. We use Adam (Kingma and Ba, 2014) with a learning rate of  $10^{-3}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.998$ , L2 weight decay of 0.01, learning rate warmup over the first 8,000 steps, and linear decay of the learning rate. We use a dropout probability of 0.1 in all layers. Our implementation uses the OpenNMT framework (Klein et al., 2017); training takes place on four V100 GPUs. To avoid overfitting, we monitor the BLEU scores of the training and development sets and stop training when their difference is larger than four points.

## Appendix B Evaluating Various Decoding Schemes

Here we investigate how different decoding schemes used to produce queries affect the retrieval effectiveness. We experiment with two decoding methods: beam search and top-*k* random sampling with different beam sizes (number of generated hypotheses). Results are shown in Figure 2. Top-*k* random sampling is slightly better than beam search across all beam sizes, and we observed a peak in the retrieval effectiveness when 10 queries are appended to the document. We conjecture that this peak occurs because too few queries yield insufficient diversity (fewer semantic

matches) while too many queries introduce noise and reduce the contributions of the original text to the document representation.