Deep Query Likelihood Model for Information Retrieval*

Shengyao Zhuang[0000−0002−6711−0955], Hang Li[0000−0002−5317−7227], and Guido Zuccon[0000−0003−0271−5563]

The University of Queensland, Brisbane, St. Lucia
{*shengyao.zhuang*, *hang.li*, *g.zuccon*}@uq.edu.au

Abstract. The query likelihood model (QLM) for information retrieval has been thoroughly investigated and utilised. At the basis of this method is the representation of queries and documents as language models; then retrieval corresponds to evaluate the likelihood that the query could be generated by the document. Several approaches have arisen to compute such probability, including by maximum likelihood, smoothing and considering translation probabilities from related terms.

In this paper, we consider estimating this likelihood using modern pre-trained deep language models, and in particular the text-to-text transfer transformer (T5) – giving rise to the QLM-T5. This approach is evaluated on the passage ranking task of the MS MARCO dataset; empirical results show that QLM-T5 significantly outperforms traditional QLM methods, as well as a recent ad-hoc methods that exploits T5 for this task.

1 Introduction

Language modelling has been introduced in Information Retrieval (IR) in the late '90s to score documents for a query [18, 5] and as alternative to other popular methods such as TF-IDF and BM25. The most basic and popular form of language model used in IR is unigram language model, which defines a probability distribution over the words in the collection. A common way to exploit language models in IR is within the query likelihood model (QLM) [18], on which we base the method in this paper; alternative approaches include the relevance model of Lavrenko&Croft [9] and the risk minimization framework of Zhai&Lafferty [8].

QLM scores a document for retrieval by considering the likelihood that the query could be generated by the document. The basic form of QLM uses the maximum likelihood estimator (MLE) to compute the query likelihood; this however exposes the method to issues due to data sparseness [24], e.g., the estimated probability of a query term that does not appear in the document will be zero, rendering the overall score of the document to be zero. To overcome this issue, smoothing has been commonly used. Smoothing transfers probability mass from the probability associated with a query term appearing in the document to the

* Shengyao Zhuang and Hang Li contributed equally to this work.

probability associated with that query term appearing in the collection. Extensively used smoothing methods include Jelinek–Mercer and Dirichlet smoothing [24], which interpolate, in a parametric manner, the likelihood of the term in the document with that associated with the term appearing in the collection. The optimal parameter values for these smoothing techniques are collection and application dependent [24]. Alternatives to these form of smoothing are methods that transfer probabilities across related terms (translation language models [1]) and others that use clusters and nearest neighbours [11, 7].

Recent advances in natural language processing have seen the introduction of deep language models [2, 12, 19, 20]; pre-trained versions of these models have been applied to search tasks demonstrating promising results [10]. Specifically, the common trend in IR is to obtain deep language models that have been pre-trained on a large text corpus and convert them to ranking models via fine-tuning on ranking tasks. An example is the work from Nogueira and Cho [16], where the raw text from a query-document pair is provided as input to the pre-trained deep language model BERT, which in turn outputs a relevance score. A notable benefit of using such deep language models is that no language preprocessing pipeline such as stemmers and stoppers is required. For example, different morphological variations are automatically handled by these deep language models by exploiting the knowledge gained from the pre-train and fine-tune steps.

In this paper we build upon the QLM tradition in IR, and create a novel QLM ranking method based on a specific deep language model. Our method, called QLM-T5, uses the text-to-text transfer transformer language model (T5) deep language model [20] in place of the MLE estimation in QLM; and, unlike in traditional QLM, it does so effectively without the need for further smoothing. T5 is an encoder-decoder model that has been shown effective for an array of natural language processing tasks. Our experimental results on the MS MARCO passage ranking task [15] show that QLM-T5 significantly outperforms traditional QLM methods, demonstrating the benefit of deep language models used within a QLM approach to IR.

2 T5 Query Language Model

The query likelihood model calculates the probability $P(Q|D)$ of generating the query Q from a given document D. Traditional approaches in IR use the maximum likelihood estimation (MLE) and smoothing methods to compute this probability [24]. Recent autoregressive deep language models such as generative pre-trained (GPT) [19] and text-to-text transfer transformer (T5) [20] can alternatively be used to calculate the likelihood of generating a target text given an input text using the teacher forcing inference mechanism: instead of taking the generated token as the input to the next time step, the target token is passed as the next input. The likelihood of generating an entire sequence of target tokens is then computed by the product of the sampling probabilities of the next target tokens from the output probability distributions of each time step.

In this work we focus on using the T5 deep language model, which has been already exploited in previous work in IR, but in an alternative form, i.e., to

generate possible query variations to append to the document representation, which is then used for retrieval (doc2query-T5 method) [17].

The T5 model is an encoder-decoder architecture. When using the teacher forcing mechanism, the document text tokens $d_0, d_1 ... d_n \in D$ are provided as input to the encoder, while the target query text tokens $q_0, q_1 ... q_{|Q|} \in Q$ plus a decoder start of sentence token $< bos >$ at the beginning of the sequence are provided as input to the decoder. At each time step t, the decoder outputs the probability $P_{T5}(q_{t+1})$ of sampling the next target query token:

$$T5_t(Encoder(d_0, d_1 ... d_n), Decoder(< bos >, q_0, q_1 ... q_t)) = P_{T5}(q_{t+1}) \quad (1)$$

It is important to note that the probability of sampling the next query token is conditioned to the document text and all previous query tokens[1]:

$$P_{T5}(q_{t+1}) = P_{T5}(q_{t+1}|D, < bos >, q_0, q_1 ... q_t) \quad (2)$$

This is differ from the traditional unigram QLM, where the sampling probabilities of each token only depend on the document text, but somewhat resemble dependence language models [4, 13] that provide a similar mechanism.

We take a similar approach to the traditional QLM to exploit T5 for retrieval. Specifically, we compute the query (log) likelihood for Q given the document D as

$$log(P_{QLM-T5}(Q, D)) = log(P_{T5}(< bos >)) + \sum_{i=0}^{|Q|-1} log(P_{T5}(q_i)) \quad (3)$$

3 Empirical Evaluation

We are interested to empirically verify the effectiveness of QLM-T5, compared to traditional forms of QLM; we further compare QLM-T5 to a recent method that also exploits T5 for ranking (doc2query-T5 [17]), but without casting T5 in the QLM framework. For this, we use the development portion of the MS MARCO Passage Ranking Dataset [15]. This portion consists of ≈8.8 million passages and 6980 unique queries; on average, each query has one relevant passage only.

Passages were indexed with Anserini [23] using the default parameters. Anserini was also used to produce runs for BM25 ($k1 = 0.82$ and $b = 0.68$), Query Language Models with Dirichlet (QLM-D, $\mu = 1,000$) and Jelinek Mercer (QLM-JM, $\lambda = 0.1$) smoothing, and Sequential Dependence Model using QLM-JM [13] (QLM-JM-SDM), retrieving the top 1,000 passages for each query. These form our first-stage retrieval baselines. We used QLM-JM-SDM to inform us regarding whether it may have been the inclusion of query term dependencies, rather than the actual deep language model, that produced gains over QLM-D/JM.

Because the inference stage of T5 is computationally expensive, in our experiments we used QLM-T5 as a second-stage re-ranker, with BM25 used as the first-stage ranker. We then also created runs where QLM-D and QLM-JM were

[1] The first query token q_0 only depends on the document text D plus the ¡bos¿ token.

used as second-stage re-ranker on top of BM25. For completeness, we also ran our QLM-T5 using QLM-D and QLM-JM as first-stage rankers. Although the aim of our experiments is to study the effectiveness of QLM-T5 with respect to other methods in the QLM framework, we also reproduced the doc2query-T5 model [17] to provide further context for the interpretation of our results. The doc2query-T5 model also relies on T5; furthermore, the same fine-tuned model was used[2]. However it does so by leveraging T5 to source possible query candidates that may be asked regarding a target document (passage in the case of these experiments). These query candidates are appended to the document to enhance its representation – retrieval is then performed with BM25 operated on the new representation of the documents.

As evaluation metrics, we use MRR@10, nDCG and INST. MRR@10[3] was used despite remarks that this is an unstable metric (Fuhr's argument [3], but perhaps more importantly Zobel&Rashidi's findings [25]) because this is the only metric used in the MS MARCO leaderboard, to which we want to allow comparison for further contextualisation of the results reported here. The use of nDCG for this task is less controversial (though note only binary relevance and mostly single-relevant documents for each query). The cut-offs considered were at 1 to model the use of the method for selecting an answer in context of e.g., a conversational search agent; at 3 and 10 to model a typical web search scenario; and at 1,000 to provide an evaluation of the complete ranking. We also computed INST [14] using the publicly available implementation from Koopman&Zuccon [6]. INST is a weighted precision metric where the probability of a user assessing a result at a specific rank depends on the rank position, the expected number of relevant documents T, and the actual number of relevant documents encountered up to that rank. This metric suits well the MS MARCO task, which is a question-answer based task with $T = 1$ (we use this value). Statistical analysis of results is performed using two-tailed paired t-test.

4 Results

Empirical results are reported in Table 1. The first four rows in the table show BM25 is superior to QLM-D, QLM-JM and QLM-JM-SDM on MS MARCO (differences statistical significant, $p < 0.01$); the superiority of BM25 with regards to QLM-D and QLM-JM is consistent with previous findings on other collections [21, 22]. The next pair of rows shows that the traditional QLM methods are not effective second-stage rankers either.

We now focus on the effectiveness of the proposed QLM-T5, which is used to re-rank results from a first-stage ranker (all results up to rank 1,000). We find that the use of QLM-T5 (irrespective of the first-stage method QLM-T5 uses) significantly outperforms first-stage retrieval runs, other re-rankers, and

[2] T5 model for MS MARCO from Nogueira et al. [17], fine-tuned to maximize query likelihood.

[3] i.e. the reciprocal rank value (averaged across all queries) up to rank 10 if a relevant document has been retrieved by then, otherwise zero.

Method	ndcg@1	ndcg@3	ndcg@10	ndcg@1000	INST	MRR@10
BM25	0.1042	0.1736	0.2340	0.3161	0.0916	0.1874
QLM-JM	0.0960	0.1586	0.2181	0.2955	0.0849	0.1740
QLM-D	0.0831	0.1371	0.1874	0.2752	0.0730	0.1491
QLM-JM-SDM	0.1044	0.1674	0.2271	0.3032	0.0900	0.1825
BM25+QLM-JM	0.0960	0.1586	0.2181	0.3006	0.0849	0.1741
BM25+QLM-D	0.0831	0.1371	0.1875	0.2795	0.0730	0.1492
QLM-JM+QLM-T5	0.1765	0.2786	0.3577	0.4086	0.1485	0.2948
QLM-D+QLM-T5	0.1769	0.2790	0.3595	0.4123	0.1489	0.2960
BM25+QLM-T5	**0.1784**	**0.2823**$^{\lozenge}$	**0.3647**$^{\lozenge\S}$	**0.4215**$^{\lozenge\S}$	**0.1506**$^{\lozenge}$	**0.2997**$^{\lozenge\S}$
doc2query-T5+BM25	0.1653	0.2600	0.3377	0.4139	0.1389	0.2768

Table 1. Effectiveness of first-stage and rerank methods. BM25+QLM-T5 is statistically significant better ($p < 0.01$) than all first-stage rankers, including doc2query-T5+BM25. BM25+QLM-T5 is statistically significant better ($p < 0.01$) than QLM-JM+QLM-T5 on metrics indicated by $\lozenge$, and BM25+QLM-T5 is statistically significant better ($p < 0.01$) than QLM-D+QLM-T5 on metrics indicated by §.

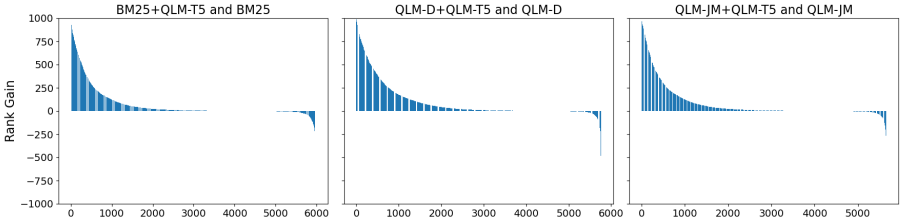

Fig. 1. Rank position gains/losses per query for QLM-T5 re-ranker compared to the respective first-stage retrieval method.

the doc2query-T5 model, which also relies on the T5 language model, on several evaluation metrics. Among all QLM-T5 runs, we find that the one that uses BM25 as first-stage ranker outperforms the others, and differences are statistically significant ($p < 0.01$) on several evaluation metrics.

Furthermore, in Figure 1 we present the ranks gained (or lost) by QLM-T5 with respect to BM25, QLM-D and QLM-JM. Specifically, we measure how many rank positions the relevant passages have gained (lost) compared to the corresponding first-stage ranker method. Figure 1 indicates that QLM-T5 reranker sensibly improves rankings (movements of up to 991 ranks) for more than 50% of the queries for BM25, QLM-JM, and QLM-D, with more than $\approx 1,500$ queries exhibiting gains of over 100 rank positions. The method does however produce some losses: a small amount of queries appear to have rank losses for QLM-T5. Similar findings are obtained when nDCG was used in place of rank position.

To better understand when QLM-T5 worked and when it failed, we further analyzed the queries with the maximum (991) and minimum (-495) rank gains/losses between BM25+QLM-T5 and BM25. For query *"what does it mean*

when you dream about babies", QLM-T5 achieved the maximum rank gain of 991: the relevant passage is pushed from BM25's rank position 993 up to rank 2. We note that the passage placed by BM25+QLM-T5 at rank 1 also appears relevant to us: *"... Dreams that include babies are positive signs. Dreaming about interacting with a baby or simply seeing a baby in a dream can mean that pleasant surprises and fortuitous occurrences are about to occur in your life..."*,

For query *"how many tables can sql server join"*, QLM-T5 had the largest rank loss (-495): BM25 placed the relevant passage at rank 255 while BM25+QLM-T5 at rank 750. We further note, however, that the top passage by BM25+QLM-T5 is *"... A SQL Server JOIN is performed whenever two or more tables are joined in a SQL statement."*, which appears to us to be relevant to the query[4].

These examples suggest that (1) QLM-T5 can successfully capture the semantic meaning of queries and passages, and produce a good match; (2) losses observed for QLM-T5 might be because of unjudged passages in MS MARCO, (3) results on MS MARCO should be considered very carefully as the dataset does not contain information about unjudged documents (thus rendering impossible the computation of residuals, e.g., for INST) and assessments appear to be very shallow and primarily based on BM25.

5 Conclusion and Future Work

In this paper, we have adapted the T5 deep language model within the query likelihood model to rank passages. Results on the MS MARCO benchmark dataset show that QLM-T5 significantly outperforms traditional QLM methods, quantifying the benefits of using deep language models within QLM in place of MLE and smoothed estimators. We also show that QLM-T5 more effectively models query dependencies than sequential dependence models.

A drawback of QLM-T5 is its computational efficiency. The method, being based on a transformer based neural network, requires considerable running time at inference. In addition, unlike traditional QLM methods (but akin to sequential dependence models), the calculation of the likelihood of each query term is conditioned on all previous query terms: pre-computing and storing query term likelihoods independently of the query is then not possible. This makes it reasonable to execute the QLM-T5 as a second-stage reranker, but it is infeasible to use it as a first-stage ranker instead. However, we believe that this issue could be partially alleviated by storing outputs of the encoder layer of T5 in the index so that at runtime the only inference needed is at the decoder level. Compared to other strong neural re-ranker baselines, such as BERT-based re-ranker [16], our model is outperformed in terms of MRR@10 (BERT-Large: 0.365 vs. QLM-T5: 0.300). Future work will explore this direction along with alternative avenues to improve the efficiency of QLM-T5, e.g., so that it becomes reasonable to apply it to document ranking tasks, besides the considered passage ranking.

[4] The passage marked relevant in MS MARCO for this query is *"... A JOIN clause is used to combine rows from two or more tables, based on a related column between them..."*.

Acknowledgements. Hang Li is funded by the Grain Research and Development Corporation (GRDC), project AgAsk (UOQ2003-009RTX). Associate Professor Guido Zuccon is the recipient of an Australian Research Council DECRA Research Fellowship (DE180101579) and a Google Faculty Award.

References

1. Berger, A., Lafferty, J.: Information retrieval as statistical translation. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 222–229 (1999)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4171–4186 (2019)
3. Fuhr, N.: Some common mistakes in ir evaluation, and how they can be avoided. In: ACM SIGIR Forum. vol. 51, pp. 32–41. ACM New York, NY, USA (2018)
4. Gao, J., Nie, J.Y., Wu, G., Cao, G.: Dependence language model for information retrieval. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 170–177 (2004)
5. Hiemstra, D.: A linguistically motivated probabilistic model of information retrieval. In: International Conference on Theory and Practice of Digital Libraries. pp. 569–584. Springer (1998)
6. Koopman, B., Zuccon, G.: A test collection for matching patients to clinical trials. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 669–672. SIGIR '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2911451.2914672, https://doi.org/10.1145/2911451.2914672
7. Kurland, O., Lee, L.: Corpus structure, language models, and ad hoc information retrieval. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 194–201 (2004)
8. Lafferty, J., Zhai, C.: Document language models, query models, and risk minimization for information retrieval. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 111–119. SIGIR '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/383952.383970, https://doi.org/10.1145/383952.383970
9. Lavrenko, V., Croft, W.B.: Relevance based language models. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 120–127. SIGIR '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/383952.383972, https://doi.org/10.1145/383952.383972
10. Lin, J., Nogueira, R., Yates, A.: Pretrained transformers for text ranking: Bert and beyond. arXiv preprint arXiv:2010.06467 (2020)
11. Liu, X., Croft, W.B.: Cluster-based retrieval using language models. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 186–193 (2004)
12. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

13. Metzler, D., Croft, W.B.: A markov random field model for term dependencies. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 472–479 (2005)
14. Moffat, A., Bailey, P., Scholer, F., Thomas, P.: Inst: An adaptive metric for information retrieval evaluation. In: Proceedings of the 20th Australasian Document Computing Symposium. ADCS '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2838931.2838938, https://doi.org/10.1145/2838931.2838938
15. Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., Deng, L.: Ms marco: A human-generated machine reading comprehension dataset (2016)
16. Nogueira, R., Cho, K.: Passage re-ranking with bert. arXiv preprint arXiv:1901.04085 (2019)
17. Nogueira, R., Lin, J., Epistemic, A.: From doc2query to doctttttquery. Online preprint (2019)
18. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. pp. 275–281 (1998)
19. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
20. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019)
21. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Now Publishers Inc (2009)
22. Speriosu, M., Tashiro, T.: Comparison of okapi bm25 and language modeling algorithms for ntcir-6. Justsystems Corporation **14** (2006)
23. Yang, P., Fang, H., Lin, J.: 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. pp. 1253–1256 (2017)
24. Zhai, C., Lafferty, J.: A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS) **22**(2), 179–214 (2004)
25. Zobel, J., Rashidi, L.: Corpus bootstrapping for assessment of the properties of effectiveness measures. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 1933–1952. CIKM '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3340531.3411998, https://doi.org/10.1145/3340531.3411998