RePair My Queries: Personalized Query Reformulation via Conditional Transformers

Yogeswar Lakshmi Narayanan and Hossein Fani(✉)

University of Windsor, Windsor, ON, Canada
{lakshmiy,hfani}@uwindsor.ca

Abstract. Search engines have difficulty searching into knowledge repositories because they are not tailored to the users' differing information needs. User queries are, more often than not, under-specified or contain ambiguous terms that also retrieve irrelevant documents. Personalized query reformulation aims to refine queries per user, enhancing the relevance of search results while avoiding semantic drift. This task remains challenging due to the inadequate number of queries in user's search sessions, let alone a query itself often suffers from ambiguity and is too brief. Existing methods have employed session history or click-throughs to enrich the query context, though one crucial cue has been overlooked: the user herself. In this paper, we propose leveraging conditional transformers such as the text-to-text transfer transformer (`t5`) to incorporate a user-tailored pretext to the input sequence as prior conditions to generate personalized reformulation of queries in the output sequence. Our experiments on the `aol` query log demonstrated the effectiveness of `t5` in personalized query reformulation, without any loss of generality to other conditional transformers. The codebase to support the reproducibility of our research is available at https://github.com/fani-lab/RePair/tree/uid-wise24.

Keywords: Query Reformulation · Personalized Information Retrieval

1 Introduction

Extracting relevant information presents significant challenges to search engines as user queries are often short and unclear, leading to the retrieval of *ir*relevant information. Query reformulation, also known as query suggestion, aims to transform the user's *original* query into a new *reformulated* version that more accurately reflects the user's intent (information need) and, therefore, enhances the relevance of search results. Existing works utilize machine learning to learn *better* reformulations of an original query [1,5] using web retrieval datasets like `aol` [1] or `msmarco` [20] following the *weak* assumption that input queries improve gradually within a search session, i.e., the last query in the search session is the *best* reformulation of the original query [5]. However, users' intents may undergo gradual evolution or sudden changes within search sessions [27], resulting in a loss of sequential context between queries, known as *query drift*. Recently, new

M. Barhamgi et al. (Eds.): WISE 2024, LNCS 15436, pp. 219–229, 2025.
https://doi.org/10.1007/978-981-96-0579-8_16

Table 1. A query from `aol` [15] by different users with diverging intents.

q	user-less q^*	`map`	uid	clicked url	user intent	user-based q^*	`map`
		0.142	05183534	www.tanforless.com	airbrush	*'tanforless com'*	1.000
'tanning bed lotions'	*'sunless tanning bed'*	0.014	13149841	www.suntanning.com	salons	*'tanning salons'*	0.083
		0.000	09469379	www.bodyconcept.com	bodybuilding	*'fitness tech'*	0.500

research efforts have been put into producing standard benchmark datasets that are free of query drifts and designed specifically to train and evaluate the efficacy of query reformulation methods for web or *non*-web information retrieval systems [2,27,32]. Nonetheless, existing query reformulation methods are based on *objective* relevance judgement, assuming the same query from different users has the same retrieval intents, which is ill-posed and discounting in real-world scenarios. From Table 1, whether the query *'tanning bed lotions'* is submitted by an *athlete* or *celebrity* would ideally yield diverging and *subjective* sets of relevant information. As seen, the former user clicks on a *'fitness'*-related webpage while the latter clicks on a *'skin care'*-related one. Therefore, should a query reformulation method yield a better version of the submitted query by such users to retrieve more relevant webpages, it needs to consider the user.

Various methods have been proposed to personalize the information retrieval processes to each user by including word embeddings [33], and creating a unified information access system using dense retrieval [31]. However, little to no work has studied the positive effect of personalized query reformulation except that of works done by Zhang et al. [33], where they exploited word embedding to learn the semantics of queries for personalization, semantic dependency of terms, and topic dependency. However, a major drawback to their work is that they handle words with multiple meanings in a single representation, and the out-of-vocabulary problem is commonly present in vector embedding. Their work also requires significant computational resources to generate and evaluate candidate queries.

In this paper, we propose to utilize conditional transformers for *personalized* query reformulation. Specifically, (1) for each user and her submitted query, we form the *subjective* set of relevant webpages per query per user using the clicked urls by the user. (2) We then train a conditional transformer, e.g., text-to-text-transfer transformer (`t5`) [24], to learn the transformation from the user's subjective relevant webpages to her query using her identifier (`uid`) as the condition, as shown in Fig. 1. (3) During inference, given an unseen query submitted by a user, we feed it to the trained transformer along with the user's identifier as the condition and consider the generated stream of tokens in the output as the personalized reformulated query for the user.

Conditional transformers have already been extensively employed in conversational information retrieval [30], question answering [29], and query reformulation in particular [2]. For instance, Nogueira et al. [22] proposed `doc2query` to expand documents to semantically related keywords to address vocabulary mismatch between the original queries and relevant documents. They trained

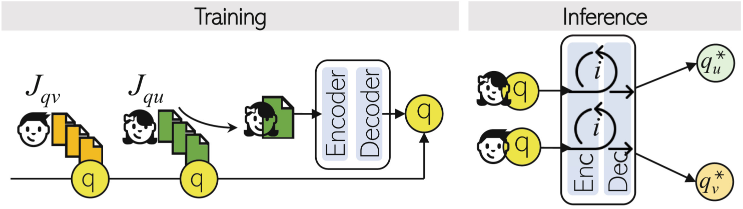

Fig. 1. Utilizing conditional transformers for personalized query reformulation.

the `Transformer` [28] to translate relevant documents of a query into the query. Shortly after, Nogueira et al. proposed `doc-t5-query` by replacing the `Transformer` with `t5` [24] observing a substantial improvement in retrieval effectiveness [21], followed by Arabzadeh et al. [2] who used `doc-t5-query` to generate training datasets for supervised query reformulation methods at a large scale. Mitra et al. [19] utilized bert [6] to find relevant questions to a query that featured as *'people also asked'* for question suggestion in search engines. However, no work has studied the application of conditional transformers for personalization in query reformulation, and we are the first to study the search efficacy of the personalized reformulations of queries compared to when the user is overlooked, referred to as *user-less* reformulations. In a host of sparse and dense information retrieval methods including `bm25` [25] and the recent `colbert` [12], and evaluation metrics such as mean average precision (`map`) and mean reciprocal rank (`mrr`), our experiments demonstrate that incorporating user when reformulating queries through transformers like `t5` enhances the efficacy of reformulated queries in retrieving more relevant information.

The main contributions of our work can be summarized as follows:

1. We propose personalized query reformulation to incorporate user context into the user's query reformulations.
2. We show that conditional transformers can be seamlessly tapped into for the task of personalized query reformulation.
3. We conduct experiments on `aol` [15], the well-known benchmark web retrieval dataset, to show the synergistic impact of incorporating user context compared to the lack thereof in query reformulations.

2 Background: Conditional Transformer

We tap into the conditional transformers to personalize query reformulations. Herein, we briefly formalize conditional language modeling [8] along with its modern implementations, the transformers [6,13,24], omitting details of architectures. We refer readers to Vaswani et al. [28] for more details.

An *un*conditional language model assigns a probability $p(q)$ to a sequence of words $q = [w_1, w_2, ..., w_{|q|}]$, which can be decomposed using the chain rule as

$p(q) = \prod_{i=1} p(w_{i+1}|w_1, ..., w_i)$. Intuitively, a language model predicts the probability of the next word, given the history of preceding words. A simplifying strategy to develop a language model is to make a Markov assumption, i.e., to forget the distant past and only consider the last few n words, referred to as n-gram language models, i.e., $p(q) = \prod_{i=1} p(w_{i+1}|w_{i-n}, ..., w_i)$. To eschew the Markov assumption, Mikolov et al. [18] proposed a recurrent neural network (rnn) for language modeling that encodes the entire history to a dense vector of fixed dimensionality $\mathbf{h}$ via a single layer neural architecture such that $p(w_{i+1}|w_1, w_2, ..., w_i) \approx p(w_{i+1}|\mathbf{h}_{i-1}, w_i)$. Later, Sutskever et al. [26] and others [3,4,9,10,14] proposed seq-to-seq encoder-decoder neural architectures to learn translation from an input sequence to a target sequence using two recurrent neural networks where the first one encodes an input sequence of words $[w_1, w_2, ..., w_{i-1}]$ onto $\mathbf{h}_{i-1}$ (encoder rnn), and the second one decodes (generates) the rest of the sequence $[w_i, ..., w_{|q|}]$ from the $\mathbf{h}_{i-1}$ (decoder rnn). Finally, Luong et al. [14] proposed *global* attention where the probability of generating a word at decoder rnn is conditioned not only on $\mathbf{h}_{i-1}$ but also on all $\mathbf{h}_{<i-1}$ of the encoder rnn. This allows the decoder to attend over all words in the input sequence selectively. To reduce the computational complexity of recurrence at encoder and decoder recurrent neural networks, Vaswani et al. [28] proposed the `Transformer`, a seq-2-seq model with *no* recurrent connections to enable parallel calculation of $\mathbf{h}_{<i}$ at the encoder and $\mathbf{h}_{>=i}$ at the decoder models, which yielded promising performance on machine language translation and led to a large body of research on transformer-based language modeling [6,13,24].

Meanwhile, such models paved the way for *conditional* language modelling, where probabilities are assigned to sequences of words given a pretext that encodes a condition. Formally, a conditional language model estimates $p(q|\mathtt{X}) = \prod_{i-1} p(w_{i+1}|\mathtt{X}, w_1, ..., w_i)$, that is, the probability of the next word, given the history of previously generated words and conditioning context `X`, which is kept fixed during the encoding and decoding phases. For instance, Keshar et al. [11] proposed `ctrl`, a transformer that can condition on a predefined set of keywords that influence the *genre* the generated sequence of words in the output for the same input sequence of words (prompt); e.g., p(*'sharp'*|`X`=`review`, *'knife'*) vs. p(*'tool'*|`X`=`wiki`, *'knife'*). Subsequently, conditional transformers have been employed as a single universal (generalist) model to perform multiple natural language processing tasks, from question-answering to document summarization to sentiment analysis, relieving the need for several models [11,17,23]. Inspired by `ctrl` and other conditional transformers, we aim to apply conditional transformers to personalize query reformulation for users, `X` being the user identifier, to control query reformulations on a per-user basis, as explained hereafter.

3 Problem Definition

Let $u, v \in \mathcal{U}$ be two users who submitted an original query q for their varying intents and $\mathcal{J}_{qu}$ and $\mathcal{J}_{qv}$ are the *subjective* set of relevant webpages obtained from their differing clicked urls. Given an information retrieval metric $m_r(\cdot : \mathcal{J})$, which measures the quality of retrieved information for an input query $\cdot$ with respect to a reference set of relevant webpages $\mathcal{J}$ under an information retrieval method r, the goal of personalized query reformulation is to generate user-based changes q_u^* and q_v^* for the same original query q for all users who submitted q, such as u or v, in order to retrieve more relevant webpages concerning each user's intent compared to when the same original query q or its *user-less* reformulated version q^* *ir*respective of the users is used. Formally:

$$\wedge_{u \in \mathcal{U}} \{m_r(q_u^* : \mathcal{J}_{qu}) > m_r(q^* : \mathcal{J}_{qu}) > m_r(q : \mathcal{J}_{qu})\} \tag{1}$$

where $\wedge$ is the logical **and** that should hold *true* for all users $u \in \mathcal{U}$ who has submitted the original query q. For instance, from Table 1, users who submitted the same original query $q =$ *'tanning bed lotions'* have different intents that would be satisfied by different webpages. A query reformulation method that disregards who submitted the query would reformulate the query to $q^* =$ *'sunless tanning bed'* whose **map** is 0.142 for one user but drops to 0.000 for the other user, falling short of suiting *all* users' diverging intents. However, personalized query reformulation generates user-based changes whose per-user **map** are higher, as seen in the last column, such that the users are *independently* satisfied.

4 Personalized Query Reformulation

Inspired by Keshar et al. [11], given a conditional transformer τ, like ctrl [11] or t5 [24], we train τ to learn the distribution of queries given each user's subjective relevant webpages (user-based click-throughs) $\mathcal{J}_{qu}$ conditioned on the user's identifier uid, i.e., $p(q|\texttt{X}{=}\texttt{uid}, \mathcal{J}_{qu})$ such that the user identifier provides a point of personalization control over the generation process for her query q.

During training, given a user u, her submitted query q, and the set of clicked webpages by the user as $\mathcal{J}_{qu}$, we concatenate the webpages into a single document as the input sequence, prepending the user identifier, that is, $[\texttt{uid}{:}\mathcal{J}_{qu}]$ and pair it with the query q as the output sequence: $[\texttt{uid} : \mathcal{J}_{qu}] \rightarrow q$, as shown in Fig. 1 (left). Note that for a different user v who submitted the same original query q, we would have $[\texttt{vid}{:}\mathcal{J}_{qv}] \rightarrow q$. We expect a conditional transformer to learn different pathways based on users to generate the same query q. In other words, we want the transformer to learn the many-to-one mappings between many personalized intents and the same submitted query. During inference, given an *unseen* query q submitted by the user u, we feed the trained transformer τ with $[\texttt{uid}{:}q]$ to take u's personalized pathway to generate sequence of words as the reformulation q_u^* for user u, i.e., $[\texttt{uid}{:}q] \rightarrow q_u^*$. Similarly, for v, we feed $[\texttt{vid}{:}q]$ to transformer τ to generate v's reformulation of q conditioned on v, i.e., $[\texttt{vid}{:}q] \rightarrow q_v^*$.

Modern transformers apply *beam* search at their decoders to generate the most probable stream of words [26]. However, beam search tends to produce common phrases and repetitive text from the training set. Fan et al. [7], proposed to randomly select the next token of output stream from the top-k most probable tokens, which has shown to be substantially more effective [7,22]. Top-k random selection, however, yields non-deterministic output generation during inference given the same input, i.e., $[u : \mathcal{J}_{qu}] \rightarrow q^*_{iu}$. To ensure diverse and novel reformulated queries for the same query of a user, we further employ random selection upon beam search to generate a collection of reformulated queries per user's query as opposed to a single reformulated query.

We argue that user-less query reformulation, where a reformulation has been generated for a query q irrespective of the user who submitted the query, results in lower search efficacy since it falls short of satisfying *all* users' subjective relevant information, as seen in Table 1 for $q =$*'tanning bed lotions'* and its user-less reformulation q^*= *'sunless tanning bed'* whose `map` is `0.1429` for one user but drops for others, compared to personalized reformulations whose `map` are generally higher.

5 Experiments

In this section, we lay out the details of our experiments and findings toward answering our main research question:

RQ: *Does personalized query reformulation via conditional transformers improve the search experience for each user compared to when users are overlooked?*

5.1 Setup

Conditional Transformer. We performed our experiments on the pre-trained `t5-base` [24] with 220 million parameters as our conditional transformer τ and used beam search decoding with top-k = `10` random sampling. We chose `t5` for its easy installation and built on the google cloud and tpu utilization for fast model training and inference. We set the maximum input and output sequence lengths to `512` and `64`, respectively. We performed fine-tuning for `4,000` epochs using a batch size of `256` and a learning rate of `0.001`.

Dataset. We utilized the reconstructed version of the web retrieval dataset `aol` by Sean et al. [15], which includes `9,966,939` timestamped queries submitted by `650,000` users, users' clicked urls and the crawled webpages of the urls. To construct the pairs [`uid`: $\mathcal{J}_{qu}$] $\rightarrow q$ for all users, we employed the `ir-dataset` api [16]. Due to the computational complexity of processing the webpages' textual content, we decided to consider either the `title` or the concatenation of the url and title (`url.title`) as the representatives for relevant information.

Evaluation Methodology. We evaluated the performance of personalized reformulated queries q^*_u compared to reformulations of queries irrespective of users q^* as well as original queries q in retrieving at most top-`100` user-based relevant webpages $\mathcal{J}_{qu}$ in terms of mean average precision (`map`) and mean reciprocal rank (`mrr`) under `bm25` [25] and `colbert` [12] as sparse and dense information retrieval methods, respectively. We refer to a reformulated query q^* as an *oracle* query, should it obtain the highest retrieval effectiveness of `1.0` for `map` or `mrr`.

Table 2. Search efficacy of personalized vs. user-less query reformulations in aol.

	aol.title		aol.title.url	
	−user	+user	−user	+user
#q	4,459,613	7,348,389	4,672,506	8,020,979
avg #words in q	3.5849	3.0245	3.5817	2.9766
avg #words in q^* vs. q_u^*	3.0543	2.0527	3.4778	3.0717
$\texttt{bm25.mrr}(\cdot : \mathcal{J}_{qu})$				
avg $\texttt{mrr}(q)$	0.0297		0.0364	
avg $\texttt{mrr}(q^*)$	0.6670	**0.7937**	0.6807	**0.8172**
#$\texttt{mrr}(q^*) > \texttt{mrr}(q)$	2,276,965	**4,503,321**	2,845,642	**4,891,602**
%	51%	**61%**	60%	**61%**
#oracle q^*	1,069,531	**1,132,741**	1,037,103	**1,347,612**
$\texttt{bm25.map}(\cdot : \mathcal{J}_{qu})$				
avg $\texttt{map}(q)$	0.0241		0.0308	
avg $\texttt{map}(q^*)$	0.4702	**0.5693**	0.4783	**0.5562**
#$\texttt{map}(q^*) > \texttt{map}(q)$	2,583,023	**4,491,856**	2,421,347	**4,884,799**
%$\texttt{map}(q^*) > \texttt{map}(q)$	58%	**61%**	52%	**61%**
#oracle q^*	649,764	**686,682**	591,001	**855,355**
$\texttt{colbert.mrr}(\cdot : \mathcal{J}_{qu})$				
avg $\texttt{mrr}(q)$	0.0807		0.0802	
avg $\texttt{mrr}(q^*)$	0.2790	**0.2989**	0.2224	**0.310**
#$\texttt{mrr}(q^*) > \texttt{mrr}(q)$	2,200	**2,288**	2,281	**3,216**
%$\texttt{mrr}(q^*) > \texttt{mrr}(q)$	11%	**11.4%**	11.4%	**16%**
#oracle q^*	0	0	0	0
$\texttt{colbert.map}(\cdot : \mathcal{J}_{qu})$				
avg $\texttt{map}(q)$	0.0661		0.0603	
avg $\texttt{map}(q^*)$	0.1357	**0.215**	0.0990	**0.1967**
#$\texttt{map}(q^*) > \texttt{map}(q)$	2,334	**3,355**	2,715	**3,321**
%$\texttt{map}(q^*) > \texttt{map}(q)$	11.6%	**16.7%**	13.57%	**16.60%**
#oracle q^*	0	0	0	0

Baselines. To answer whether personalized query reformulations yield better search efficacy than when users are overlooked, we run our choice of the transformer, t5, conditioned on the users (+user) and lack thereof (−user).

5.2 Results

Table 2 reports statistics, including the number of queries, and the average number of words in original queries q, personalized reformulated queries (+user) and user-less reformulated queries (−user), as well as the average mrr and map improvements for each of the variants based on bm25 and colbert.

Sparse Retrieval (bm25). Looking at Table 2, we can observe personalized reformulated queries (+user) yield substantial improvements for bm25 across aol.title and aol.title.url in terms of mrr and map compared to user-less reformulated queries that are oblivious to the users (−user) as well as original queries. We also see that the number of personalized reformulated queries is almost double that of user-less reformulated queries across metrics. Further, with respect to the number of oracle queries (map=1.0 or mrr=1.0), personalized query reformulation could outperform user-less query reformulations in terms of both metrics, cementing our view on the personalization for query reformulation to improve the search experience per user.

Dense Retrieval (colbert). On the one hand, unlike bm25, dense retrieval methods like colbert are tied to extreme resource consumption for a large-scale set of original queries as in aol. On the other hand, although bm25 could yield a substantial amount of personalized query reformulations with better search efficacy compared to the original queries and user-less reformulations, there are still 39% of users' original queries whose reformulations fell short due to them being *hard* for bm25 to fetch more relevant webpages and improve the metrics. To show the search efficacy of personalized query reformulation under dense retrieval while saving computational cost, we ran colbert on 20,000 randomly sampled hard original queries, their personalized reformulations, and user-less reformulations. From Table 2, we observe superior performance gain by colbert for personalized reformulations of original queries compared to user-less reformulations across aol.title and aol.title.url in terms of mrr and map. Also, while colbert could bring more relevant webpages for bm25's hard queries using their personalized or user-less reformulations, it fell short of reaching the maximum search efficacy for *any* reformulations, i.e., the number of oracle reformulated queries is 0 for both personalized and user-less reformulations, highlighting the challenge of finding oracle reformulated queries for hard queries, be it via the personalized or user-less approaches.

6 Concluding Remarks and Future Work

In this paper, we leveraged conditional transformers to generate personalized reformulation of queries by adding user identifiers to the input sequence as prior conditions. Our experiments on the aol query log using t5 demonstrated the effectiveness of personalized query reformulation compared to user-less query reformulation where users' subjective relevant pages are overlooked. For future work, we will conduct a comparative study on various conditional transformers on domain-specific datasets to ensure the reproducibility of our work.

References

1. Ahmad, W.U., Chang, K., Wang, H.: Context attentive document ranking and query suggestion. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, pp. 385–394. ACM (2019)
2. Arabzadeh, N., Bigdeli, A., Seyedsalehi, S., Zihayat, M., Bagheri, E.: Matches made in heaven: toolkit and large-scale datasets for supervised query reformulation. In: CIKM 2021: The 30th ACM International Conference on Information and Knowledge Management, pp. 4417–4425. ACM (2021)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR (2015)
4. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 1724–1734. ACL (2014)
5. Dehghani, M., Rothe, S., Alfonseca, E., Fleury, P.: Learning to attend, copy, and generate for session-based query suggestion. In: 2017 ACM on Conference on Information and Knowledge Management, pp. 1747–1756 (2017)
6. Devlin, J., Chang, M., 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, NAACL-HLT, pp. 4171–4186. Association for Computational Linguistics (2019)
7. Fan, A., Lewis, M., Dauphin, Y.N.: Hierarchical neural story generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL, pp. 889–898. Association for Computational Linguistics (2018)
8. Graves, A.: Generating sequences with recurrent neural networks. CoRR **abs/1308.0850** (2013)
9. Jean, S., Cho, K., Memisevic, R., Bengio, Y.: On using very large target vocabulary for neural machine translation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL, pp. 1–10. The Association for Computer Linguistics (2015)
10. Kalchbrenner, N., Blunsom, P.: Recurrent continuous translation models. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 1700–1709. ACL (2013)
11. Keskar, N., McCann, B., Varshney, L.R., Xiong, C., Socher, R.: CTRL: a conditional transformer language model for controllable generation. CoRR **abs/1909.05858** (2019)
12. Khattab, O., Zaharia, M.: 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, SIGIR, pp. 39–48. ACM (2020)
13. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019)
14. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 1412–1421. The Association for Computational Linguistics (2015)

15. MacAvaney, S., Macdonald, C., Ounis, I.: Reproducing personalised session search over the AOL query log. In: Hagen, M., et al. (eds.) ECIR 2022. LNCS, vol. 13185, pp. 627–640. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99736-6_42
16. MacAvaney, S., Yates, A., Feldman, S., Downey, D., Cohan, A., Goharian, N.: Simplified data wrangling with IR_datasets. In: SIGIR 2021: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2429–2436. ACM (2021)
17. McCann, B., Keskar, N.S., Xiong, C., Socher, R.: The natural language decathlon: multitask learning as question answering. CoRR **abs/1806.08730** (2018)
18. Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., Khudanpur, S.: Recurrent neural network based language model. In: 11th Annual Conference of the International Speech Communication Association, INTERSPEECH, pp. 1045–1048. ISCA (2010)
19. Mitra, R., Gupta, M., Dandapat, S.: Transformer models for recommending related questions in web search. In: CIKM 2020: The 29th ACM International Conference on Information and Knowledge Management, pp. 2153–2156. ACM (2020)
20. Nguyen, T., et al.: MS MARCO: a human generated machine reading comprehension dataset. In: Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS. CEUR Workshop Proceedings, vol. 1773. CEUR-WS.org (2016)
21. Nogueira, R., Lin, J., Epistemic, A.: From doc2query to docTTTTTquery. Online preprint **6** (2019)
22. Nogueira, R.F., Yang, W., Lin, J., Cho, K.: Document expansion by query prediction. CoRR **abs/1904.08375** (2019)
23. Radford, A., et al.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
24. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020)
25. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inf. Retr. **3**(4), 333–389 (2009)
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, pp. 3104–3112 (2014)
27. Tamannaee, M., Fani, H., Zarrinkalam, F., Samouh, J., Paydar, S., Bagheri, E.: ReQue: a configurable workflow and dataset collection for query refinement. In: CIKM 2020: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, 19–23 October 2020, pp. 3165–3172. ACM (2020)
28. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, pp. 5998–6008 (2017)
29. Yang, H., Zhao, X., Wang, Y., Li, M., Chen, W., Huang, W.: DGQAN: dual graph question-answer attention networks for answer selection. In: SIGIR 2022: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1230–1239. ACM (2022)
30. Ye, C., Liao, L., Feng, F., Ji, W., Chua, T.: Structured and natural responses co-generation for conversational search. In: SIGIR 2022: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 155–164. ACM (2022)

31. Zeng, H., Kallumadi, S., Alibadi, Z., Nogueira, R.F., Zamani, H.: A personalized dense retrieval framework for unified information access. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, pp. 121–130. ACM (2023)
32. Zerveas, G., Zhang, R., Kim, L., Eickhoff, C.: Brown university at TREC deep learning 2019. In: Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC. NIST Special Publication, vol. 1250. National Institute of Standards and Technology (NIST) (2019)
33. Zhang, X.: Improving personalised query reformulation with embeddings. J. Inf. Sci. **48**(4), 503–523 (2022)