Performance Prediction for Non-Factoid Question Answering

Helia Hashemi, Hamed Zamani, and W. Bruce Croft
Center for Intelligent Information Retrieval
University of Massachusetts Amherst
Amherst, MA 01003
{hhashemi,zamani,croft}@cs.umass.edu

ABSTRACT

Estimating the quality of a result list, often referred to as query performance prediction (QPP), is a challenging and important task in information retrieval. It can be used as feedback to users, search engines, and system administrators. Although predicting the performance of retrieval models has been extensively studied for the ad-hoc retrieval task, the effectiveness of performance prediction methods for question answering (QA) systems is relatively unstudied. The short length of answers, the dominance of neural models in QA, and the re-ranking nature of most QA systems make performance prediction for QA a unique, important, and technically interesting task. In this paper, we introduce and motivate the task of performance prediction for non-factoid question answering and propose a neural performance predictor for this task. Our experiments on two recent datasets demonstrate that the proposed model outperforms competitive baselines in all settings.

1 INTRODUCTION

The goal of query performance prediction (QPP) in information retrieval (IR) is predicting the effectiveness of a retrieval model for a given query [1]. QPP has been extensively explored in the context of ad-hoc retrieval [11, 13, 16, 18, 22, 23] and web search. We argue that QPP for QA is fundamentally different from QPP for ad-hoc retrieval. This is due to the shorter length of answers, the dominance of neural models in QA, and the re-ranking nature of most QA systems.[1] These fundamental differences and the important role of this task in current information access systems have motivated us to introduce the task of predicting the performance of *retrieval-based* question answering systems,[2] which is relatively unstudied. In particular, we study the task of performance prediction for *non-factoid* question answering. Non-factoid questions are considered as open-ended questions and require complex answers, like descriptions, opinions, or explanations, like, "what is the reason for life?". We believe this type of questions have a pivotal role in question answering systems, since their technologies are not as mature as factoid questions, which seek for precise facts, like "At what age did Rossini stop writing opera?".

We further propose a neural network architecture for predicting the performance of non-factoid QA systems. Our model utilizes retrieval scores and the contents of the question and the top ranked answers to estimate the performance of the result list. In addition, unlike most existing performance predictors, our model consists of a natural language understanding component by making use of bidirectional encoder representations from Transformers (BERT) [5].

We evaluate the proposed model on two recent non-factoid QA datasets that contain reasonable numbers of queries for training neural models: (1) WikiPassageQA [2] that consists of 3332 training questions with an average of 1.7 relevant passages from Wikipedia. (2) ANTIQUE [6] which is a non-factoid dataset with 2,426 training questions collected from a community question answering website. Our experiments suggest that the proposed model outperforms competitive baselines in predicting the performance of various retrieval models, including neural ranking models.

2 RELATED WORK

Query performance prediction, also known as quality estimation and query difficulty prediction, has been widely studied for ad-hoc retrieval and web search [1, 3, 7, 13, 16–18, 23]. The task of query performance prediction is defined as predicting the retrieval effectiveness of a search engine given an issued query with no implicit or explicit relevance information.

Query performance prediction approaches can be partitioned into two disjoint sets: pre-retrieval and post-retrieval approaches. Pre-retrieval QPP approaches predict the performance of each query based on the content and the context of the query in addition to the corpus statistics. Pre-retrieval predictors are often derived from linguistic or statistical information. Part-of-speech tags, as well as syntactic and morphological features of query terms are among the linguistic features used for query performance prediction. Inverse document frequency [3] and average query term coherence [8] are examples of statistical information used for this task. Hauff et. al [7] provided a through overview of the pre-retrieval QPP approaches.

Alternately, post-retrieval QPP approaches estimate query performance by analyzing the result list returned by the retrieval engine in response to the query. Carmel and Yom-Tov [1] categorized post-retrieval predictors into the following three categories. (1) Clarity-based approaches [3] estimate the query performance by measuring the coherence (clarity) of the result list with respect to the collection. (2) Robustness-based approaches [23] predict the query performance by estimating the *robustness* of the result list. (3) A variety of post-retrieval approaches predict the query performance by analyzing the retrieval score distribution [11, 18, 23], and are commonly referred to as score-based approaches.

[1] See Section 3.1 for in detail differences of QPP for QA and ad-hoc retrieval.
[2] Other QA settings, such as machine reading comprehension, that involve selecting a specific short span within a sentence, selecting answer from predefined choices, or predicting a blanked-out word of a sentence, are not the focus of this work.

SIGIR '19, July 21–25, 2018, Paris, France

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

There is also a line of research that combines multiple predictors from multiple categories, e.g., the utility estimation framework [16].

Krikon et al. [9] studied QPP in the context of passage retrieval with a focus on factoid questions. In more detail, they estimated the performance of passage retrieval as the first retrieval phase in factoid QA. However, in this paper, we focus on non-factoid QA with is fundamentally different [21]. We study this method as a baseline. In addition, Roitman [13] proposed a QPP method for ad-hoc retrieval by utilizing passage information, which is out of the scope of this paper.

Liu et al.[10] addressed the question difficulty estimation in community question answering websites based on the skills of users. Their approach is independent of the question and answer contents, and is orthogonal to our work. Shah and Pomerantz [15] predicted the quality of an answer in response to a question in a CQA system in terms of 13 criteria, and users' profile data. Unlike this work which focused on measuring the correlation between user satisfaction and an answer's quality criteria like politeness, readability, conciseness, etc., our work focuses on predicting the performance of a result list in response to a question.

3 MOTIVATION

Similar to ad-hoc retrieval, accurate and real-time performance predictors could potentially be used in triggering a specific action in the retrieval system, such as selecting an index traversal algorithm at query time, choosing the correct number of documents to process in a cascaded multistage retrieval system, choosing the most effective ranking function per query, or selecting the best variant from multiple query reformulations [22]. In addition, we believe performance prediction for non-factoid questions can potentially play a vital role in the current modern information access systems. The emergence of new generation of search interfaces including conversational search systems and intelligent assistant services (e.g., Siri, Cortana, and Google assistant) intensifies the importance of an effective and efficient performance prediction method. To elaborate more on these examples, consider a conversational search scenario in which the system must decide whether it can address the user's information need, or go through follow up and clarifying questions to get a better understanding of the information need. This is even more important for the systems with a voice-only interface, such as Amazon's Alexa. Lack of features such as auto correction, auto completion, and different levels of English fluency among users, all in all, introduce new obstacles for query understanding. On the other hand, since the output of QA systems, given their voice or text interface, is mostly a single answer that should address the user's information need leaves almost no room for error. This is where an accurate QPP method could have a significant impact.

3.1 QPP for Ad-hoc Retrieval vs. QA

We claim that the task of performance prediction in question answering is fundamentally different from performance prediction in ad-hoc retrieval and web search, because:

- QPP methods in ad-hoc have been mostly designed to predict recall-oriented metrics. However, in QA systems the main metrics are precision-oriented, e.g., mean reciprocal rank.
- A number of state-of-the-art QPP methods for ad-hoc retrieval are based on term distribution in the top retrieved documents, e.g., [3, 23]. Unlike ad-hoc document retrieval, in QA, candidate answers are often short, e.g., sentence-level or passage-level, and they often have a little term overlap with each other as well as the question.
- The notion of relevance in QA is different from ad-hoc retrieval. In QA systems a relevant passage or sentence must directly answer the question, however in ad-hoc retrieval, annotations are done based on topical relevance. Many existing QPP methods for ad-hoc retrieval, e.g., [3, 23], distinguish topically similar documents from off-topic documents, which cannot perform effectively for QA.
- Many existing QPP methods predict query performance using the retrieval scores assigned to the top retrieved documents. However, given the dominance of neural network approaches in QA systems, the scale and distribution of retrieval scores returned by different neural models are significantly different. This may have a major impact on the effectiveness and robustness of score-based methods.

4 METHODOLOGY

In this section, we introduce NQA-QPP, our neural model for predicting the performance of non-factoid question answering. The model utilizes both retrieval scores and question/answer text to estimate the performance of a question answering system. Similar to [22], we design a component-based neural model as follows:

Component I: score-based component. The first component learns a representation from the scores produced by the QA system for each candidate answer. Let R be the retrieval scores for the top k retrieved answers in descending order. Inspired by the score-based QPP approaches that successfully utilizes the standard deviation of retrieval scores, such as [11, 18], we create a vector S with the size of $k-1$ such that $S[i] = \text{stdev}(R[1:i+1])$, where stdev denotes the standard deviation. In other words, the i^{th} element of S represents the standard deviation of the retrieval scores from the beginning to the rank $i+1$. We finally obtain a d-dimensional representation from the retrieval scores as $\phi_I(\hat{R}|S)$, where $\hat{R}$ is the retrieval scores R normalized using z-score normalization, and | means concatenation. The function $\phi_I : \mathbb{R}^{2k-1} \rightarrow d$ is a fully-connected feed-forward network with two hidden layers. Details of the network architecture are mentioned later in this section.

Component II: question-only component. The second component learns a representation suitable for query performance prediction from the question content, without having access to the retrieval list. This is motivated by pre-retrieval QPP methods, e.g., [7, 8]. To model this component, we use Bidirectional Encoder Representations from Transformers (BERT) [5] that recently achieved state-of-the-art performance in a wide range of natural language understanding tasks. BERT provides token-level representation for each sentence or a pair of sentences. The representation learned for the first token by BERT (i.e., `[CLS]`) can be seen as a representation for the whole sentence. We feed this representation to a fully-connected network as follows: $\phi_{II}(\phi^{\text{BERT}}_{[\text{CLS}]}(q))$, where $\phi_{II} : \mathbb{R}^l \rightarrow d$ is a fully-connected network and l denote the representation dimensionality of BERT. We use the pre-trained small model in which $l = 768$.[3]

Component III: question-answer component. The third component takes the content of the top k retrieved answers and learns a d-dimensional representation. To maintain our consistency, we

[3]Pre-trained BERT models: https://github.com/google-research/bert.

Table 1: Data statistics.

	WikiPassageQA	ANTIQUE
# training/validation/test queries	3332/417/416	2183/243/200
Average qrel per query	1.7	8.5

again use BERT for representing each question-answer pair. In more detail, our third component is as follows:

$$\phi_{III}\left(\phi'(\phi^{\text{BERT}}_{[\text{CLS}]}(q,a_1)) \mid \phi'(\phi^{\text{BERT}}_{[\text{CLS}]}(q,a_2)) \mid \cdots \mid \phi'(\phi^{\text{BERT}}_{[\text{CLS}]}(q,a_k))\right)$$

where ϕ' and ϕ_{III} are two fully-connected networks. In fact, ϕ_{III} takes the representation for all top k answers.

Aggregation. We aggregate the representations learned by each of the above components as follows:

$$\psi(\hat{R}|S|\phi_I|\phi^{\text{BERT}}_{[\text{CLS}]}(q)|\phi_{II}|\phi_{III}) \tag{1}$$

where ψ is a fully-connected network that produces a single real value. In addition to the output of individual components, ψ also takes their inputs (except for the question-answer component).

In all of the mentioned feed-forward networks, we use ReLU as the hidden layer activation. We employ dropout in all hidden layers to avoid overfitting. We train NQA-QPP using maximum likelihood maximization, which is equivalent to a cross-entropy loss.

5 EXPERIMENTS

Data. We evaluate our models on the following non-factoid QA datasets.[4] (1) The **WikiPassageQA** dataset [2] was created using Amazon's Mechanical Turk platform. Crowd workers were asked to create non-factoid questions based on a Wikipedia article, and indicate the location of their respective answer passages within the document. (2) **ANTIQUE** [6] is a dataset that have recently created through crowdsourcing.[5] ANTIQUE is a sample of non-factoid questions from Yahoo! Webscope L6, which is a community question answering data. Table 1 shows statistics of datasets.

Experimental Setup. We implemented our model using TensorFlow. In all experiments, the network parameters were optimized using the Adam optimizer. For hyper-parameter optimization, we performed grid search, and chose the hyper-parameters based on the Pearson's correlation on the validation set. The learning rate was selected from $\{1\times10^{-5}, 5\times10^{-4}, 1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}\}$. The batch size was selected from $\{32, 64, 128\}$. The dropout keep probability was selected from $\{0.5, 0.8, 0.9, 1.0\}$. The number of hidden layers in the dense network and their output sizes were selected from $\{1, 2\}$ and $\{10, 20, 50, 100\}$, respectively.

Evaluation Metrics. To evaluate the models, we compute the correlation between the predicted performances and the actual query performance in terms of reciprocal rank (RR). Following prior work on QPP [3, 16, 18, 22, 23], we use Pearson's correlation (P-ρ), Spearman's correlation (S-ρ), and Kendall's correlation (K-τ) coefficients. P-ρ is a linear correlation metric that is sensitive to the actual predicted performance values; while, S-ρ and K-τ are rank-based correlation metrics that only take the order of the questions into account. The correlations with a p-value of less than 0.01 and 0.001 are marked with † and ‡, respectively.

[4]We omit the WebAP dataset in our experiments, due to its small number of queries (i.e., 82), and MS MARCO dataset due to its incomplete relevance judgments.

[5]The data is publicly available at https://ciir.cs.umass.edu/downloads/Antique.

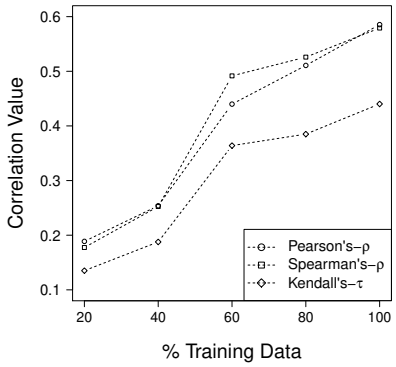

Figure 1: Learning curve for NQA-QPP on WikiPassageQA.

Results and Discussion. As mentioned earlier in Section 1, we are not aware of any performance prediction method for non-factoid question answering. Therefore, we compare our method against several query performance prediction methods that produce competitive results for ad-hoc and passage retrieval. Our baselines ranges from score-based models, i.e., σ_k [11], NQC [18], NQC.NEQT [9], WIG [23], SMV [19], and RSD [14], to clarity-based models, i.e., Clarity [3], to robustness-based models, i.e., QF [23], to combining models, i.e., UEF [16], LTRoq [12], and NeuralQPP [22]. The last baseline is a state-of-the-art QPP method for ad-hoc retrieval based on neural network. For the details about the baselines, we refer the reader to the associated articles. Please note that we tune all the hyper-parameters of all the baselines using the same procedure taken for our model. We use WIG and Pearson's correlation to implement UEF.[6]

Note that NQC, SMV, and WIG require a normalization factor. Previous work on QPP for ad-hoc retrieval concatenated all the documents in the collection and computed its score by the retrieval model, which is not possible for most neural retrieval models. Therefore, we compute this normalization factor for the neural models as the average retrieval score of all candidate answers for the question. We kept the concatenation approach for predicting the performance of the BM25 model.

In our first set of experiments, we consider three retrieval model: BM25 and two neural ranking models including aNMM (an attention-based QA model) [20] and Conv-KNRM [4]. Table 2 reports the QPP performance for the proposed method and the baselines. The neural models re-rank 100 answers retrieved by BM25. According to the results, NQA-QPP outperforms all the baselines in all settings. Interestingly, the score-based baselines perform poorly in predicting the performance of Conv-KNRM. This happens because the scale and distribution of the scores produced by neural models are different. Predicting the performance of BM25 is still easier for NQA-QPP, compared to the other retrieval models. It is worth noting that NeuralQPP is developed for ad-hoc retrieval and is based on the bag-of-words assumption, however, NQA-QPP takes advantage of a more sophisticated language modeling representation and performs better.

Table 3 shows the performance of NQA-QPP for predicting different ranking metrics. For the sake of space, in this experiment

[6]To improve reproducibility, we release our implementation and hyper-parameter tuning for all the models.

Table 2: The results for predicting the performance of different retrieval models, in terms of reciprocal rank (RR).

	QPP	BM25			aNMM			Conv-KNRM		
	Method	P-ρ	S-ρ	K-τ	P-ρ	S-ρ	K-τ	P-ρ	S-ρ	K-τ
WikiPassageQA Dataset	σ_k	0.4573‡	0.5218‡	0.3822‡	0.2481‡	0.1852‡	0.1286‡	0.0335	0.0447	0.0299
	NQC	0.4711‡	0.5179‡	0.3768‡	0.0466	0.0286	0.0180	0.0158	0.0452	0.0302
	WIG	0.1421†	0.2525‡	0.1784‡	0.1537	0.1724	0.1201	0.0181	0.0777	0.0535
	SMV	0.4601‡	0.5190‡	0.3776‡	0.0351	0.0617	0.0414	0.0060	0.0505	0.0323
	RSD	0.4672‡	0.5337‡	0.4005‡	0.2516‡	0.1946‡	0.1320‡	0.0219	0.0381	0.0401
	Clarity	0.4129‡	0.4204‡	0.3011‡	0.2764‡	0.3463‡	0.2395‡	0.1264†	0.1333†	0.0892†
	QF	0.0194	0.0389	0.0308	0.0876	0.0700	0.0509	0.0588	0.1055	0.0733
	NQC.NEQT	0.4811‡	0.5281‡	0.3821‡	0.0921	0.0514	0.0191	0.0321	0.0631	0.0758
	UEF	0.2109‡	0.3356‡	0.2361‡	0.2696‡	0.3698‡	0.2545‡	0.1843‡	0.2286‡	0.1465‡
	LTRoq	0.4921‡	0.5088‡	0.3472‡	0.2749‡	0.2112‡	0.1973‡	0.1621‡	0.2371‡	0.1281‡
	NeuralQPP	0.5112‡	0.4980‡	0.2801‡	0.2411‡	0.1819‡	0.1255‡	0.1714‡	0.2104‡	0.1359‡
	NQA-QPP	**0.5854**‡	**0.5791**‡	**0.4402**‡	**0.3436**‡	**0.3731**‡	**0.2640**‡	**0.2069**‡	**0.2490**‡	**0.1671**‡
ANTIQUE Dataset	σ_k	0.0966	0.2889‡	0.2120‡	0.2777‡	0.2624‡	0.1852‡	-0.0455	-0.0236	-0.0162
	NQC	0.2224†	0.2693‡	0.1949‡	0.0450	-0.0007	0.0018	-0.0021	0.0175	0.0143
	WIG	0.1456	0.2258†	0.1658†	0.0461	0.1206	0.0822	0.0143	0.1312	0.0899
	SMV	0.1557	0.2265†	0.1646†	0.0382	-0.0038	-0.0018	-0.0207	-0.0239	-0.0135
	RSD	0.1044	0.3041‡	0.2517‡	0.2816‡	0.2773‡	0.2146‡	0.0043	0.0176	0.0081
	Clarity	0.1300	0.0780	0.0561	0.2196‡	0.2559‡	0.1771‡	0.0493	0.0807	0.0547
	QF	0.0025	0.0570	0.0425	0.1771†	0.0528	0.0426	-0.0178	-0.0866	-0.0658
	NQC.NEQT	0.2315‡	0.2800‡	0.1891‡	0.0504	0.0031	0.0116	0.0513	0.0358	0.0423
	UEF	0.1649	0.3351	0.2421	0.3230‡	0.3007‡	0.2293‡	0.1304	0.1119	0.0980
	LTRoq	0.2810‡	0.2992‡	0.2572‡	0.3346‡	0.3125‡	0.2917‡	0.1915‡	0.1621‡	0.1348‡
	NeuralQPP	0.2711‡	0.3111‡	0.2384‡	0.3211‡	0.2968‡	0.2263‡	0.1644‡	0.1512‡	0.1031‡
	NQA-QPP	**0.4118**‡	**0.4428**‡	**0.3291**‡	**0.3708**‡	**0.4202**‡	**0.3013**‡	**0.2736**‡	**0.2446**†	**0.1757**†

Table 3: Results of NQA-QPP for predicting the performance in terms of different ranking metrics.

Metric	RR	AP	P@1	P@3	P@10
P-ρ	0.5854‡	0.5327‡	0.5508‡	0.5273‡	0.4136‡
S-ρ	0.5791‡	0.5358‡	0.5295‡	0.5434‡	0.4660‡
K-τ	0.4402‡	0.3920‡	0.4402‡	0.4512‡	0.3738‡

we only focus on predicting the performance of BM25 on WikiPassageQA. As shown in the table, NQA-QPP is robust in predicting different ranking metrics. The only metric with significant drop is P@10 and the reason is that there is on average only 1. relevant passages per query in the WikiPassageQA dataset (see Table 1).

Figure 1 plots the learning curve for NQA-QPP on predicting the performance of BM25 on WikiPassageQA. According to the plot, the performance of NQA-QPP is not yet saturated. This suggests that our model can perform better given more training data.

6 CONCLUSIONS

In this paper, we introduced and motivated the task of performance prediction for non-factoid question answering. Furthermore, we proposed NQA-QPP, a neural model for predicting the performance of a retrieval model for non-factoid questions. We conducted our experiments on two diverse non-factoid QA datasets. Our results showed that NQA-QPP outperforms all the baselines in different retrieval settings. The learning curve demonstrated the potential of the model to perform better given more training data.

7 ACKNOWLEDGMENT

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-1715095. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

[1] D. Carmel and E. Yom-Tov. 2010. *Estimating the Query Difficulty for Information Retrieval* (1st ed.). Morgan and Claypool Publishers.
[2] D. Cohen, L. Yang, and W. B. Croft. 2018. WikiPassageQA: A Benchmark Collection for Research on Non-factoid Answer Passage Retrieval. In *SIGIR '18*.
[3] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. 2002. Predicting Query Performance. In *SIGIR '02*. 299–306.
[4] Z. Dai, C. Xiong, J. Callan, and Z. Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *WSDM '18*.
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).
[6] H. Hashemi, M. Aliannejadi, H. Zamani, and W. B. Croft. 2019. ANTIQUE: A Non-Factoid Community Question Answering Benchmark. In *arXiv preprint 1905.08957*.
[7] C. Hauff, D. Hiemstra, and F. de Jong. A Survey of Pre-retrieval Query Performance Predictors. In *CIKM '08*.
[8] J. He, M. Larson, and M. de Rijke. 2008. Using Coherence-based Measures to Predict Query Difficulty. In *ECIR '08*. 689–694.
[9] E. Krikon, D. Carmel, and O. Kurland. 2012. Predicting the Performance of Passage Retrieval for Question Answering. In *CIKM '12*. 2451–2454.
[10] J. Liu, Q. Wang, C. Lin, and H. Hon. Question Difficulty Estimation in Community Question Answering Services. In *EMNLP '13*.
[11] J. Pérez-Iglesias and L. Araujo. 2010. Standard Deviation As a Query Hardness Estimator. In *SPIRE '10*. 207–212.
[12] F. Raiber and O. Kurland. 2014. Query-performance Prediction: Setting the Expectations Straight. In *SIGIR '14*. 13–22.
[13] H. Roitman. 2018. An Extended Query Performance Prediction Framework Utilizing Passage-Level Information. In *ICTIR '18*.

[14] H. Roitman, S. Erera, and B. Weiner. 2017. Robust Standard Deviation Estimation for Query Performance Prediction. In *ICTIR '17*. 245–248.
[15] C. Shah and J. Pomerantz. 2010. Evaluating and Predicting Answer Quality in Community QA. In *SIGIR '10*. ACM.
[16] A. Shtok, O. Kurland, and D. Carmel. 2010. Using Statistical Decision Theory and Relevance Models for Query-performance Prediction. In *SIGIR*. 259–266.
[17] A. Shtok, O. Kurland, and D. Carmel. 2016. Query Performance Prediction Using Reference Lists. *TOIS* 34, 4 (June 2016), 19:1–19:34.
[18] A. Shtok, O. Kurland, D. Carmel, F. Raiber, and G. Markovits. 2012. Predicting Query Performance by Query-Drift Estimation. *TOIS* 30, 2 (May 2012), 11:1–11:35.
[19] Y. Tao and S. Wu. 2014. Query Performance Prediction By Considering Score Magnitude and Variance Together. In *CIKM '14*. 1891–1894.
[20] L. Yang, Q. Ai, J. Guo, and W. B. Croft. 2016. aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. In *CIKM '16*. 287–296.
[21] L. Yang, Q. Ai, D. Spina, R. Chen, L. Pang, W. B. Croft, J. Guo, and F. Scholer. 2016. Beyond Factoid QA: Effective Methods for Non-factoid Answer Sentence Retrieval. In *ECIR '16*. 115–128.
[22] H. Zamani, W. B. Croft, and J. S. Culpepper. 2018. Neural Query Performance Prediction using Weak Supervision from Multiple Signals. In *SIGIR '18*.
[23] Y. Zhou and W. B. Croft. 2007. Query Performance Prediction in Web Search Environments. In *SIGIR '07*. 543–550.