Multi-step Entity-centric Information Retrieval for Multi-Hop Question Answering

Ameya Godbole[1*], **Dilip Kavarthapu**[1*], **Rajarshi Das**[1*],
Zhiyu Gong[1], **Abhishek Singhal**[1], **Hamed Zamani**[1],
Mo Yu[2], **Tian Gao**[2], **Xiaoxiao Guo**[2], **Manzil Zaheer**[3] **and Andrew McCallum**[1]
[1]University of Massachusetts Amherst,
[2]IBM Research, [3]Google Research

Abstract

Multi-hop question answering (QA) requires an information retrieval (IR) system that can find *multiple* supporting evidence needed to answer the question, making the retrieval process very challenging. This paper introduces an IR technique that uses information of entities present in the initially retrieved evidence to learn to '*hop*' to other relevant evidence. In a setting, with more than **5 million** Wikipedia paragraphs, our approach leads to significant boost in retrieval performance. The retrieved evidence also increased the performance of an existing QA model (without any training) on the HOTPOTQA benchmark by **10.59** F1.

1 Introduction

Multi-hop QA requires finding multiple supporting evidence, and reasoning over them in order to answer a question (Welbl et al., 2018; Talmor and Berant, 2018; Yang et al., 2018). For example, to answer the question shown in figure 1, the QA system has to retrieve two different paragraphs and reason over them. Moreover, the paragraph containing the answer to the question has very little lexical overlap with the question, making it difficult for search engines to retrieve them from a large corpus. For instance, the accuracy of a BM25 retriever for finding *all* supporting evidence for a question decreases from 53.7% to 25.9% on the 'easy' and 'hard' subsets of the HOTPOTQA training dataset.[1]

We hypothesize that an effective retriever for multi-hop QA should have the "*hopiness*" built into it, by design. That is, after retrieving an initial set of documents, the retriever should be able to "hop" onto other documents, if required. We note that, many supporting evidence often share common

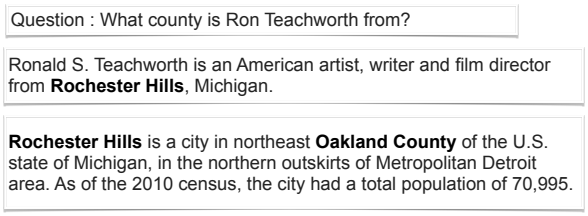

Figure 1: Multi-hop questions require finding multiple evidence and the target document containing the answer has very little lexical overlap with the question.

(*bridge*) entities between them (e.g. "Rochester Hills" in figure 1). In this work, we introduce a model that uses information about entities present in an initially retrieved paragraph to jointly find a passage of text *describing* the entity (*entity-linking*) and also determining whether that passage would be relevant to answer the multi-hop query.

A major component of our retriever is a re-ranker model that uses contextualized entity representation obtained from a pre-trained BERT (Devlin et al., 2018) language model. Specifically, the entity representation is obtained by feeding the query and a Wikipedia paragraph describing the entity to a BERT model. The re-ranker uses representation of both the initial paragraph and the representation of all the entities within it to determine which evidence to gather next.

Essentially, our method introduces a new way of *multi-step* retrieval that uses information about intermediate entities. A standard way of doing multi-step retrieval is via *pseudo-relevance feedback* (Xu and Croft, 1996; Lavrenko and Croft, 2001) in which relevant terms from initial retrieved documents are used to reformulate the initial question. A few recent works learn to reformulate the query using task specific reward such as document recall or performance on a QA task (Nogueira and Cho, 2017; Buck et al., 2018; Das et al., 2019). However, these methods do not necessarily use the information about entities present in the evidence as they might not be the more frequent/salient terms in it.

* Equal contribution. Correspondence to {agodbole, rajarshi}@cs.umass.edu

[1]According to Yang et al. (2018), the easy (hard) subset primarily requires single (multi) hop reasoning. We only consider queries that have answers as spans in at least one paragraph.

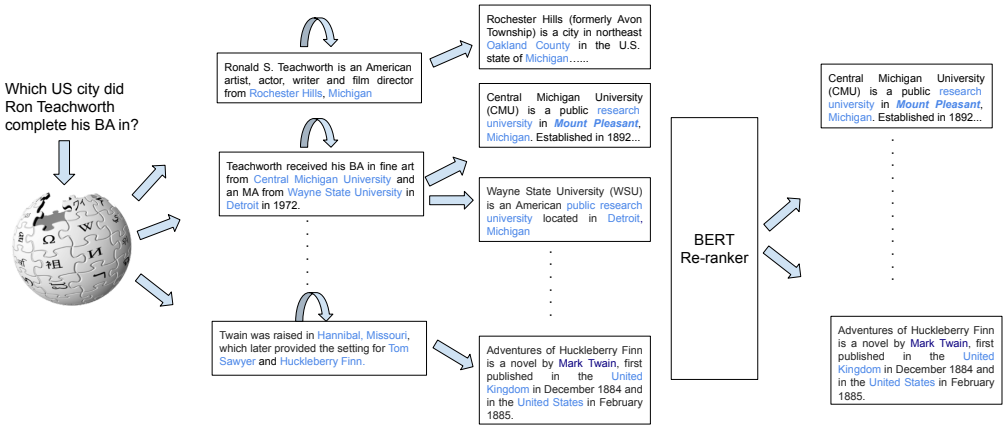

Figure 2: Overview of our approach. We use the entity mentions present in the initially retrieved paragraphs to link to paragraphs describing them. Next, the BERT-based re-ranker scores the chain of initial and the entity-describing paragraph. Note the presence of self-loop from the initial paragraphs to accommodate for questions that do not require 'hopping' to a new paragraph. Finally, the paragraph at the end of every chain is reported in the order in which the chain it belongs to is ranked.

Empiricially, our method outperforms all of these methods significantly for multi-hop QA. Our work is most closely related to the recently proposed BERT re-ranker model of Nogueira and Cho (2019). However, unlike us, they do not model the chains of evidence paragraphs required for a multi-hop question. Secondly, they also do not have a entity linking component to identify the relevant paragraphs. Our model out-performs them for multi-hop QA.

To summarize, this paper presents an entity-centric IR approach that jointly performs entity linking and effectively finds relevant evidence required for questions that need multi-hop reasoning from a large corpus containing millions of paragraphs. When the retrieved paragraphs are supplied to the baseline QA model introduced in Yang et al. (2018), it improved the QA performance on the hidden test set by 10.59 F1 points.[2]

2 Methodology

Our approach is summarized in Figure 2. The first component of our model is a standard IR system that takes in a natural language query 'Q' and returns an initial set of evidence. For our experiments, we use the popular BM25 retriever, but this component can be replaced by any IR model. We assume that all spans of entity mentions have been identified in the paragraph text by a one-time pre-processing, with an entity tagger.[3]

[2]Code, pre-trained models and retrieved paragraphs are released — https://github.com/ameyagodbole/entity-centric-ir-for-multihop-qa

[3]We plan to explore joint learning of entity tagging with linking and retrieval as future work.

Entity Linking The next component of our model is an entity linker that finds an introductory Wikipedia paragraph describing the entity, corresponding to each entity mention. Several IR approaches (Xiong et al., 2016; Raviv et al., 2016) use an off-the-shelf entity linker. However, most entity linking systems (Ganea and Hofmann, 2017; Raiman and Raiman, 2018) have been trained on Wikipedia data and hence using an off-the-shelf linker would be unfair, since there exists a possibility of test-time leakage. To ensure strictness, we developed our own simple linking strategy. Following the standard approach of using mention text and hyper-link information (Cucerzan, 2007; Ji and Grishman, 2011), we create a mapping (alias table) between them. The alias table stores mappings between a mention string (e.g. "Bill") and various entities it can refer to (e.g. Bill Clinton, Billy Joel, etc). The top-40 documents returned by the BM25 retriever on the dev and test queries are also ignored while building the alias table. At test time, our re-ranker considers all the candidate entity paragraphs that a mention is linked to via the alias table. Although simple, we find this strategy to work well for our task and we plan to use a learned entity linker for future work.

Re-ranker The next component of our model is a BERT-based re-ranker that ranks the chains of paragraphs obtained from the previous two components of the model. Let Q denote the query, D denote a paragraph in the initial set of paragraphs returned by the BM25 retriever. Let e denote an entity mention present in D and E be the linked

document returned by the linker for e. If there are multiple linked documents, we consider all of them. Although our retriever is designed for multi-hop questions, in a general setting, most questions are not multi-hop in nature. Therefore to account for questions that do not need hopping to a new paragraph, we also add a 'self-link' (Figure 2) from each of the initial retrieved paragraph, giving the model the ability to stay in the same paragraph.

To train the re-ranker, we form *query-dependent* passage representation for both D and E. The query Q and the paragraph E are concatenated and fed as input to a BERT encoder and the corresponding [CLS] token forms the entity representation **e**. Similarly, the document representation **d** is set to the embedding of the [CLS] token obtained after encoding the concatenation of Q and D. The final score that the entity paragraph E is relevant to Q is computed by concatenating the two query-aware representation **d** and **e** and passing it through a 2-layer feed-forward network as before. It should be noted, the final score is determined by both the evidence paragraphs D and E and as we show empirically, not considering both leads to decrease in performance.

During training, we mark a chain of paragraphs as a positive example, if the last paragraph of the chain is present in the supporting facts, since that is a chain of reasoning that led to a relevant paragraph. All other paragraph chains are treated as negative examples. In our experiments, we consider chains of length 2, although extending to longer chains is straightforward. The training set had on an avg. 6.35 positive chains per example suggesting a multi-instance multi-label learning training setup (Surdeanu et al., 2012). However, for this work, we treat each chain independently. We use a simple binary cross-entropy loss to train the network.

3 Experiments

For all our experiment, unless specified otherwise, we use the open domain corpus[4] released by Yang et al. (2018) which contains over 5.23 million Wikipedia abstracts (introductory paragraphs). To identify spans of entities, we use the implementation of the state-of-the-art entity tagger presented in Peters et al. (2018).[5] For the BERT encoder, we use the BERT-BASE-UNCASED model.[6] We use the implementation of widely-used BM25 retrieval available in Lucene.[7]

Model	ACCURACY @2	@5	@10	@20	MAP
BM25	0.093	0.191	0.259	0.324	0.412
PRF-TFIDF	0.088	0.157	0.204	0.258	0.317
PRF-RM	0.083	0.175	0.242	0.296	0.406
PRF-TASK	0.097	0.198	0.267	0.330	0.420
BERT-re-ranker	0.146	0.271	0.347	0.409	0.470
QUERY+E-DOC	0.101	0.223	0.301	0.367	0.568
Our Model	**0.230**	**0.482**	**0.612**	**0.674**	**0.654**

Table 1: Retrieval performance of models on the HOTPOTQA benchmark. A successful retrieval is when *all* the relevant passages for a question are retrieved from more than 5 million paragraphs in the corpus.

3.1 IR for MultiHop QA

We introduce a new way of doing multi-step retrieval. A popular way of doing it in traditional IR systems is via pseudo-relevance feedback (PRF). The PRF methods assume that the top retrieved documents in response to a given query are relevant. Based on this assumption, they expand the query in a weighted manner. PRF has been shown to be effective in various retrieval settings (Xu and Croft, 1996). We compare with two widely used PRF models — The Rocchio's algorithm on top of the TF-IDF retrieval model (PRF-TFIDF) (Rocchio, 1971) and the relevance model (RM3) based on the language modeling framework in information retrieval (PRF-RM) (Lavrenko and Croft, 2001). Following prior work (Nogueira and Cho, 2017), we use query likelihood retrieval model with Dirichlet prior smoothing (Zhai and Lafferty, 2001) for first retrieval run.

Nogueira and Cho (2017) proposed a new way of query reformulation — incorporating reward from a document-relevance task (PRF-TASK) and training using reinforcement learning. Recently, Nogueira and Cho (2019) proposed a BERT based passage re-ranker (BERT-re-ranker) that has achieved excellent performance in several IR benchmarks. But, its performance has not been evaluated on multi-hop queries till now. For a fair comparison with our model which looks at paragraphs corresponding to entities, we use top 200 paragraphs retrieved by the initial IR model for BERT-re-ranker instead of 25 for our model.[8]

Table 1 reports the accuracy(@k) of retrieving

[4] https://hotpotqa.github.io/wiki-readme.html
[5] https://allennlp.org/models
[6] https://github.com/google-research/bert
[7] https://lucene.apache.org/
[8] There were 2.725 entities in a paragraph on average. We wanted to make sure to give the BERT-re-ranker baseline atleast 25 $\times$ 2.275 paragraphs.

all[9] the relevant paragraphs required for answering a question in HOTPOTQA[10] within the top k paragraphs. We also report the mean average precision score (MAP) which is a strict metric that takes into account the relative position of the relevant document in the ranked list (Kadlec et al., 2017). As we can see, our retrieval technique vastly outperforms other existing retrieval systems with an absolute increase of **26.5**% (accuracy@10) and **18.4**% (MAP), when compared to BERT-re-ranker. The standard PRF techniques do not perform well for this task. This is primarily because the PRF methods rely on statistical features like frequency of terms in the document, and fail to explicitly use information about entities, that may not be frequently occurring the paragraph. In fact, their performance is a little behind the standard retrieval results of BM25, suggesting that this benchmark dataset needs entity-centric information retrieval. The PRF-TASK does slightly better than other PRF models, showing that incorporating task-specific rewards can be beneficial. However, as we find, RL approaches are slow to converge[11] as rewards from a down-stream tasks are sparse and action space in information retrieval is very large.

Ablations. We investigate whether modeling the chain of paragraphs needed to reach the final paragraph is important or not. As an ablation, we ignore the representation of the initial retrieved document D_1 and only consider the final document representing the entity (QUERY+E-DOC). Table 1 shows that, indeed modeling the chain of documents is important. This makes intuitive sense, since to answer questions such as the county where a person is from (figure 1), modeling context about the person, should be helpful. We also evaluate, if our model performs well on single-hop questions as well. This evaluation is a bit tricky to do in HOTPOTQA, since the evaluataion set only contains questions from 'hard' subset (Yang et al., 2018). However, within that hard subset, we find the set of question, that has the answer span present in *all* the supporting passages (SINGLE-HOP (HARD)) and only in *one* of the supporting passages (MULTI-HOP (HARD))[12]. The intuition is that if there are multiple evidence containing the answer spans then it might be a little easier for a downstream QA model to identify the answer span. Figure 3 shows that our model performs equally well on both type of queries and hence can be applied in a practical setting.

[9]This is different from the usual hits@k metric where at least one relevant evidence is required to be present in the top-k retrieved evidence.

[10]Since, the supporting passage information is only present for train & validation set, we consider the validation set as our hidden test set and consider a subset of train as validation set.

[11]Training took ∼2 weeks for comparable performance.

[12]There were 1184 SINGLE-HOP (HARD) and 4734 MULTI-HOP (HARD) queries.

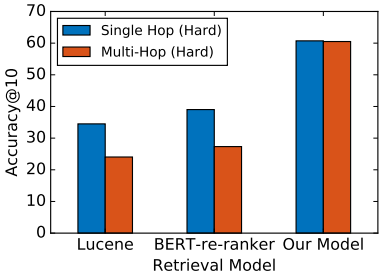

Figure 3: Our retrieval model works equally well for single-hop queries. This can be attributed to the presence of self-loops in the model which can make the model not hop to a different paragraph, if not required.

Model	EM	F1
Baseline Reader (Yang et al., 2018)	23.95	32.89
Our re-implementation	26.06	35.67
+ retrieved result	**35.36**	**46.26**

Table 2: Performance on QA task on hidden test set of HOTPOTQA after adding the retrieved paragraphs

3.2 Performance on HOTPOTQA

Table 2 shows the performance on the QA task. We were able to achieve better scores than reported in the baseline reader model of Yang et al. (2018) by using Adam (Kingma and Ba, 2014) instead of standard SGD (our re-implementation). Next, we use the top-10 paragraphs retrieved by our system from the entire corpus and feed it to the reader model. We achieve a **10.59** absolute increase in F1 score than the baseline. It should be noted that we use the simple baseline reader model and we are confident that we can achieve better scores by using more sophisticated reader architectures, e.g. using BERT based architectures. Our results show that retrieval is an important component of an open-domain system and equal importance should be given to both the retriever and reader component.

3.3 Zero-shot experiment on Wikihop

We experiment if our model trained on HOTPOTQA can generalize to another multi-hop dataset – WIKIHOP (Welbl et al., 2018), without any training. In the WIKIHOP dataset, a set of candidate introductory Wikipedia paragraphs are given per question. Hence, we do not need to use our initial BM25 retriever.

We assign the first entity mention occurring in a

Model	acc@2	acc@5
BM25	0.06	0.30
BERT-re-ranker (zs)	0.08	0.27
Our Model (zs)	**0.10**	**0.41**

Table 3: Zero-shot (zs) IR results on WIKIHOP.

paragraph as the textual description of that entity. For instance, if the first entity mention in the paragraph is 'Mumbai', we assign that paragraph as the textual description for the entity 'Mumbai'. This assumption is often true for the introductory paragraphs of a Wikipedia article. Next, we perform entity linking of mentions by just simple string matching (i.e. linking strings such as 'mumbai' to the previous paragraph). After constructing a small subgraph from the candidate paragraphs, we apply our model trained on HOTPOTQA. Since the dataset does not provide explicit supervision for which paragraphs are useful, we mark a paragraph as 'correct' if it contains the answer string. The baseline models we compare to are a BM25 retriever and a BERT-re-ranker model of (Nogueira and Cho, 2019) that ranks all the candidate supporting paragraphs for the question. Table 3 shows our model outperforms both models in zero-shot setting.

4 Related Work

Document retrieval using entities. Analysis of web-search query logs has revealed that there is a large portion of entity seeking queries (Liu and Fang, 2015). There exists substantial work on modeling documents with entities occurring in them. For example, Xiong et al. (2016) represents a document with bag-of-entities and Raviv et al. (2016) use entity-based language modeling for document retrieval. However, they depend on an off-the-shelf entity tagger, where as we jointly perform linking and retrieval. Moreover, we use contextualized entity representations using pre-trained LMs which have been proven to be better than bag-of-words approaches. There has been a lot of work which leverages knowledge graphs (KGs) to learn better entity representations (Xiong and Callan, 2015; Xiong et al., 2017; Liu et al., 2018) and for better query reformulation (Cao et al., 2008; Dalton et al., 2014; Dietz and Verga, 2014). Our work is not tied to any specific KG schema, instead we encode entities using its text description.

Neural ranking models have shown great potential and have been widely adopted in the IR community (Dehghani et al., 2017; Guo et al., 2019; Mitra et al., 2017; Zamani et al., 2018, inter-alia). Bag-of-words and contextual embedding models, such as word2vec and BERT, have also been explored extensively for various IR tasks, from document to sentence-level retrieval (Padigela et al., 2019; Zamani and Croft, 2016, 2017).

5 Conclusion

We introduce an entity-centric approach to IR that finds relevant evidence required to answer multi-hop questions from a corpus containing millions of paragraphs leading to significant improvement to an existing QA system.

Acknowledgements

This work is funded in part by the Center for Data Science and the Center for Intelligent Information Retrieval, and in part by the National Science Foundation under Grant No. IIS-1514053 and in part by the International Business Machines Corporation Cognitive Horizons Network agreement number W1668553 and in part by the Chan Zuckerberg Initiative under the project Scientific Knowledge Base Construction. 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

Christian Buck, Jannis Bulian, Massimiliano Ciaramita, Wojciech Gajewski, Andrea Gesmundo, Neil Houlsby, and Wei Wang. 2018. Ask the right questions: Active question reformulation with reinforcement learning. In *ICLR*.

Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. 2008. Selecting good expansion terms for pseudo-relevance feedback. In *SIGIR*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*.

Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *SIGIR*.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *ICLR*.

Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. 2017. Neural ranking models with weak supervision. In *SIGIR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Laura Dietz and Patrick Verga. 2014. Umass at trec web 2014: Entity query feature expansion using knowledge base links. Technical report, MASSACHUSETTS UNIV AMHERST.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *EMNLP*.

Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. 2019. A deep look into neural ranking models for information retrieval. *arXiv preprint arXiv:1903.06902*.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *ACL*.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. 2017. Knowledge base completion: Baselines strike back. *arXiv preprint arXiv:1705.10744*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Victor Lavrenko and W Bruce Croft. 2001. Relevance-based language models. In *SIGIR*.

Xitong Liu and Hui Fang. 2015. Latent entity space: a novel retrieval approach for entity-bearing queries. *Information Retrieval Journal*.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *ACL*.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*.

Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. In *EMNLP*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Harshith Padigela, Hamed Zamani, and W. Bruce Croft. 2019. Investigating the successes and failures of BERT for passage re-ranking. *arXiv preprint arXiv:1905.01758*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Jonathan Raphael Raiman and Olivier Michel Raiman. 2018. Deeptype: multilingual entity linking by neural type system evolution. In *AAAI*.

Hadas Raviv, Oren Kurland, and David Carmel. 2016. Document retrieval using entity-based language models. In *SIGIR*.

J. J. Rocchio. 1971. Relevance feedback in information retrieval. In *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. In *TACL*.

Chenyan Xiong and Jamie Callan. 2015. Esdrank: Connecting query and documents through external semi-structured data. In *CIKM*.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2016. Bag-of-entities representation for ranking. In *ICTIR*.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017. Word-entity duet representations for document ranking. In *SIGIR*.

Jinxi Xu and W Bruce Croft. 1996. Quary expansion using local and global document analysis. In *SIGIR*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Hamed Zamani and W. Bruce Croft. 2016. Embedding-based query language models. In *ICTIR*.

Hamed Zamani and W. Bruce Croft. 2017. Relevance-based word embedding. In *SIGIR*.

Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *CIKM*.

Chengxiang Zhai and John Lafferty. 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR*.