Contextual Re-Ranking with Behavior Aware Transformers

Chen Qu[†], Chenyan Xiong[*], Yizhe Zhang[*], Corby Rosset[*], W. Bruce Croft[†], and Paul Bennett[*]
University of Massachusetts Amherst[†], Microsoft AI & Research [*]
{chenqu, croft}@cs.umass.edu; {chenyan.xiong, yizhe.zhang, corosset, pauben}@microsoft.com

ABSTRACT

In this work, we focus on the contextual document ranking task, which deals with the challenge of user interaction modeling for conversational search. Given a history of user feedback behaviors, such as issuing a query, clicking a document, and skipping a document, we propose to introduce behavior awareness to a neural ranker, resulting in a *Hierarchical Behavior Aware Transformers* (HBA-Transformers) model. The hierarchy is composed of an *intra-behavior attention* layer and an *inter-behavior attention* layer to let the system effectively distinguish and model different user behaviors. Our extensive experiments on the AOL session dataset [3] demonstrate that the hierarchical behavior aware architecture is more powerful than a simple combination of history behaviors. Besides, we analyze the conversational property of queries. We show that coherent sessions tend to be more *conversational* and thus are more demanding in terms of considering history user behaviors.

KEYWORDS

Conversational Search; Neural-IR; Behavior Aware Transformers

ACM Reference Format:
Chen Qu, Chenyan Xiong, Yizhe Zhang, Corby Rosset, W. Bruce Croft, and Paul Bennett. 2020. Contextual Re-Ranking with Behavior Aware Transformers. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3397271.3401276

1 INTRODUCTION

Due to the recent emergence of intelligent personal assistants, information seeking activities are gradually moving to a conversational interface. In such a scenario, history user interactions are vital to understand the user’s information need. Thus, we study modeling user interactions in the search history via the contextual ranking task, as a crucial step towards conversational search.

Conversational search and ad-hoc retrieval share the same backbone of document ranking. To fulfill a complex information need with a search engine, users typically need to search for multiple turns. Temporally connected turns are referred to as a *session* or *task* [11, 14]. In each turn, the user issues a query, browses search engine result pages (SERPs), and clicks on documents for further investigation. This iterative information seeking process bears a strong resemblance to conversational search. The current turn is *contextual* in terms of history user behaviors. Compared to conversational search, the contextual ranking task enjoys better data availability and maturer evaluation methods. Thus, we work on contextual ranking as the groundwork towards conversational search.

The idea of modeling context for document ranking dates back to early works [4, 5, 9, 13, 15]. Typical techniques include query expansion and learning to rank. In addition to using the current session only, White et al. [15] also leverages similar tasks from search logs. Due to the lack of large-scale session datasets, most approaches are mainly limited to non-parametric or feature-based models. These models may not be able to capture the complex user intent dynamics in a real search session. With the recent emergence of neural-IR, researchers have begun to revisit this topic with deep models. For example, Ahmad et al. [2, 3] employ hierarchical recurrent structures to model queries and clickthrough information across history turns. They use multi-task learning to optimize for both document ranking and query suggestion.

In this paper, we introduce behavior awareness to a neural ranker. Users’ past behaviors in a session may contribute differently when determining the relevance of a candidate document. For example, behaviors in an adjacent turn could be more informative than those in a distant turn. Moreover, clicked and skipped documents could provide distinct clues of the information need. It is vital for the ranker to distinguish history behaviors. Thus, we design the Hierarchical Behavior Aware Transformers (HBA-Transformers) on top of BERT and enable flexible incorporation of the session history.

We conduct extensive experiments on the AOL session data [3]. We first show that a BERT based ranker without any context information is able to outperform a recent context aware recurrent model [3]. We further show that BERT is capable of modeling session history by simply prepending history user behaviors to the current query. Moreover, we demonstrate that our hierarchical behavior attention mechanism is more powerful in this scenario than a simple concatenation. Finally, we conduct an in-depth analysis on the conversational property of queries with the Microsoft Generic Intent Encoder [17]. We show that coherent sessions tend to be more *conversational* as they are more demanding in considering the user behaviors in the session history.

2 BEHAVIOR AWARE TRANSFORMERS

In this section, we first set up the task of contextual re-ranking and then describe our Hierarchical Behavior Aware Transformers.

2.1 Task Definition

We are given search logs organized in sessions. A session is denoted as $\{T^i\}_{i=1}^N$, where T^i is the i-th turn and N is the total number of

SIGIR ’20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401276

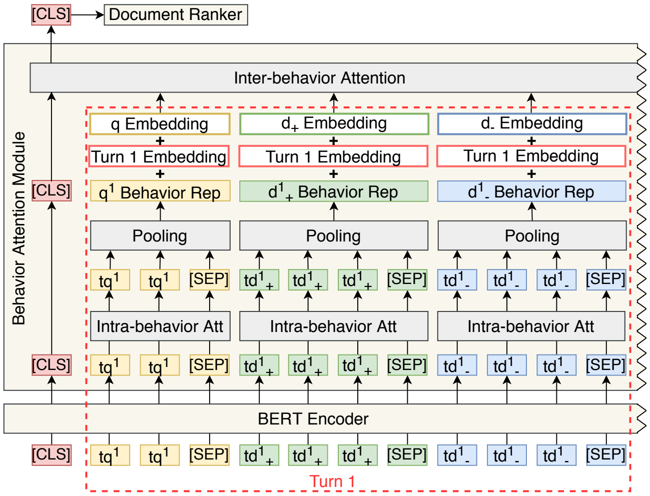


Figure 1: We illustrate how the hierarchy works on the first turn. The rest of the concatenated sequence is omitted.

turns. T^i is further denoted as $\{q^i, D^i, Y^i\}$, where q^i is the query, and $D^i = \{d^i_j\}_{j=1}^k$ is the retrieved top-k documents for q^i. $Y^i = \{y^i_j\}_{j=1}^k$ is a set of binary relevance labels for D^i. The click label $y^i_j = 1$ means the user makes a "satisfied click" on d^i_j after issuing q^i. D^i and Y^i reveal the *clicked* document d^i_+ and the *skipped* document d^i_- for T^i. The skipped document d^i_- is identified by the commonly used *Skip Above and Skip Next* strategy [1]. When there are multiple skipped documents, we take the first one. Therefore, we use T^i as three user behaviors: $\{q^i, d^i_+, d^i_-\}$. Given all available history turns $H^n = \{T^i\}_{i=1}^{n-1}$ up to turn n, we extract a sequence of *history user behaviors* denoted as $H^n_* = \{q^1, d^1_+, d^1_-, q^2, d^2_+, d^2_-, \ldots, q^{n-1}, d^{n-1}_+, d^{n-1}_-\}$. We further define a set of *session user behaviors* as $H^n_* \cup \{q^n\}$, to include the current query q^n. Given $H^n_* \cup \{q^n\}$, the task is to rerank D^n so that the clicked documents are ranked as high as possible.

2.2 Model

Our HBA-Transformers model is presented in Figure 1. The *BERT encoder* encodes input tokens into contextualized representations. Then the *hierarchical behavior attention module* first generates behavior representations and then attends to these representations with a holistic view of the session. Finally, the *document ranker* predicts a relevance score of the candidate document. This hierarchical architecture enables flexible integration of the session history and thus can be more effective in contextual re-ranking.

2.2.1 ***BERT Encoder.*** The encoder is shown in the lower part of Figure 1. Previous works [6, 12] apply BERT for ranking in a manner of sequence pair classification. We extend this scheme by prepending history user behaviors in a window size of w to the query segment. Specifically, the input sequence is "$\texttt{[CLS]}\ q^{n-w}\ \texttt{[SEP]}\ d_+^{n-w}\ \texttt{[SEP]}\ d_-^{n-w}\ \texttt{[SEP]} \cdots \texttt{[SEP]}\ q^n\ \texttt{[SEP]}\ d\ \texttt{[SEP]}$". Let $\mathbf{t}_m \in \mathbb{R}^h$ denotes the embedding of the m-th token in the input sequence and h denotes the hidden size. The input sequence is denoted as $\{\mathbf{t}_m\}_{m=1}^M$, where M is the sequence length. BERT outputs a contextualized token representation $\hat{\mathbf{t}}_m$ for every token by attending to all tokens in the input sequence. In other words, $\hat{\mathbf{t}}_m$ is contextualized in terms of the entire available session and the candidate document. This is the advantage of modeling the concatenation of the session over modeling each behavior individually. The BERT encoder is fine-tuned during training.

2.2.2 ***Hierarchical Behavior Attention Module.*** This module learns to attend to user behaviors and the current document to produce a history-enhanced query-document representation for ranking. The hierarchy in this module is composed of an intra-behavior attention layer and an inter-behavior attention layer.

Intra-behavior Attention. As shown in the middle part of Figure 1, the input of the intra-behavior attention layer is $\{\hat{\mathbf{t}}_m\}_{m=1}^M$, and the output is behavior-level representations for all behaviors. Given a specific session user behavior (or the current document), we isolate token representations within this behavior as $\{\hat{\mathbf{t}}_m\}_{m=s}^e$, where s and e are the start and end of this behavior. The `[CLS]` of the input sequence and the trailing `[SEP]` of this behavior are also considered as within-behavior tokens. We then apply an intra-behavior attention layer to $\{\hat{\mathbf{t}}_m\}_{m=s}^e$, followed by an average pooling on the dimension of sequence length, to get the behavior representation $\mathbf{r}$ as $\mathbf{r} = \text{AvgPool}\left(\text{IntraBehaviorAtt}(\{\hat{\mathbf{t}}_m\}_{m=s}^e)\right)$. The intra-behavior attention layer has three sub-layers: a multi-head attention layer, a feed-forward intermediate layer, and a feed-forward output layer, which are identical to those in a BERT layer. The encoding of different behaviors are vectorized to produce a set of behavior representations $\mathbf{R} = \{\mathbf{r}_{q^{n-w}}, \mathbf{r}_{d_+^{n-w}}, \mathbf{r}_{d_-^{n-w}}, \cdots, \mathbf{r}_{q^n}, \mathbf{r}_d\}$.

Inter-behavior Attention. As shown in the upper part of Figure 1, given $\mathbf{R}$ produced by the last layer, the inter-behavior attention layer considers each behavior as a whole and compute their attention to each other. We further introduce *behavior awareness* to this layer with *behavior aware embeddings*. Since the position and type are two properties that can uniquely identify a user behavior in a session, we design two sets of behavior aware embeddings, namely, the *behavior position embeddings* and the *behavior type embeddings*, to let the model be aware of these properties.

For behavior position embeddings, the position refers to the relative position in terms of the current query. We use different embeddings for behaviors at different positions. Behavior type embeddings work in a similar way. The vocabulary of behavior types is defined as $\{q, d_+, d_-, d_*\}$, where d_* denotes the current document. The behavior aware embeddings are randomly initialized and learned. These embeddings are added to the behavior representations, followed by a layer normalization (LN). For example, an enhanced behavior representation for $\mathbf{r}_{d_+^{n-w}}$ is $\hat{\mathbf{r}}_{d_+^{n-w}} = \text{LN}\left(\mathbf{r}_{d_+^{n-w}} + \mathbf{e}_{n-w} + \mathbf{e}_{d_+}\right)$, where $\mathbf{e}_{n-w} \in \mathbb{R}^h$ is the behavior position embeddings for turn $n - w$ and $\mathbf{e}_{d_+} \in \mathbb{R}^h$ is the behavior type embeddings for d_+. The `[CLS]` representation is prepended to $\mathbf{R}$ so that we can use the same pooling strategy in BERT [7]. The behavior aware embeddings for `[CLS]` are set to the same ones as the current document.

We then apply the same sub-layers as in the intra-behavior attention to $\mathbf{R}$ to obtain $\tilde{\mathbf{R}} = \{\tilde{\mathbf{r}}_{\texttt{[CLS]}}, \tilde{\mathbf{r}}_{q^{n-w}}, \tilde{\mathbf{r}}_{d_+^{n-w}}, \tilde{\mathbf{r}}_{d_-^{n-w}}, \cdots, \tilde{\mathbf{r}}_{q^n}, \tilde{\mathbf{r}}_d\}$. Different from the intra-behavior attention layer that computes attention on a token level, these transformations are now on a behavior level for bidirectional session modeling. Lastly, we obtain $\mathbf{r}^*_{\texttt{[CLS]}}$ by taking a linear projection of $\tilde{\mathbf{r}}_{\texttt{[CLS]}}$. $\mathbf{r}^*_{\texttt{[CLS]}}$ is considered as a history-enhanced query-document representation for ranking.

Table 1: Data Statistics.

Data Split	Train	Valid	Test
# Sessions	219,748	34,090	29,369
# Queries	566,967	88,021	76,159
# Avg. Queries per Session	2.58	2.58	2.59
Avg. Query Length	2.86	2.85	2.9
Avg. Doc Length	7.27	7.29	7.08
# Avg. Doc per Query	5	5	50
# Avg. Click per Query	1.08	1.08	1.11
# Min/Med/Max Queries per Session	2/2/10	2/2/10	2/2/10

2.2.3 ***Document Ranker.*** This is the last module in Figure 1. We follow the previous BERT-ranking setting [12] and use binary classification with cross-entropy loss.

3 EXPERIMENTS

3.1 Experimental Setup

3.1.1 ***Dataset.*** We use the AOL session data in Ahmad et al. [3]. that has synthetic skipped documents [3]. They use the document title as its content following previous works [8, 10]. While this is a limitation, Zamani et al. [16] indicates that the title field is very informative in re-ranking. We present data statistics in Table 1.

3.1.2 ***Competing Methods.*** Ahmad et al. [3] has shown their method CARS significantly outperforms both classical and neural ad-hoc retrieval models. Therefore, we first compare with their method, the best performing approach in this task. We then build another context aware neural model based on BERT as an even stronger baseline. To be specific, the competing methods are:

- **CARS** [3]: This uses recurrent structures to encode queries and clicks in a session. It is trained with a multi-task learning setting.
- **BERT** [12]: This model applies BERT to document ranking by considering the current query only and the candidate document.
- **BERT-Concat**: This is the HBA-Transformers without hierarchical behavior attention. It has variations that only consider specific previous user behaviors, namely, **-Q**, **-QC**, and **-QCS**. "Q", "C", "S" denote queries, clicked, and skipped documents.
- **HBA-Transformers**: This refers to our full model described in Section 2. This also has variations of **-QC** and **-QCS**.

3.1.3 ***Evaluation Metrics.*** We follow Ahmad et al. [3] and use mean reciprocal rank (MRR@all) and normalized discounted cumulative gain (nDCG@1, 3, 10) for evaluation.

3.1.4 ***Implementation Details.*** We use the BERT-Base (uncased) model and the same training scheme for all our models. We set the max sequence length to 128, the training batch size to 512, the number of training epochs to 10, and the learning rate to 1e-4. The warm up portion of the learning rate is 10% of the total steps. We set the gradient accumulation steps for BERT/BERT-Concat and HBA-Transformers to 1 and 4 respectively. The history window size is set to 3. We use half precision and train the models with 8 NVIDIA GeForce RTX 2080 Ti GPUs. We save checkpoints every 5,000 steps and evaluate on 5,000 validation sessions (about 15% of the validation data) to select the best model for the test set.

Table 2: Main evaluation results. ‡ means statistically significant improvement over the strongest baseline with $p < 0.05$.

Models	MRR	nDCG		
		@1	@3	@10
CARS[1] [3]	0.4538	0.2940	0.4249	0.5109
BERT [12]	0.5198	0.3592	0.4984	0.5813
BERT-Concat-Q	0.5196	0.3596	0.4977	0.5806
BERT-Concat-QC	0.5340	0.3759	0.5149	0.5934
BERT-Concat-QCS	0.5366	0.3787	0.5174	0.5954
HBA-Transformers-QC	**0.5450**‡	**0.3866**‡	**0.5291**‡	**0.6021**‡
HBA-Transformers-QCS	0.5446‡	0.3850‡	0.5268‡	0.6012‡

Table 3: Impact of the skipped documents and history window size. We report MRR on the test set.

Models	history=1	history=2	history=3
BERT-Concat-QC	**0.5376**	0.5345	0.5340
BERT-Concat-QCS	0.5343	0.5346	**0.5366**
HBA-Transformers-QC	**0.5593**	0.5429	0.5450
HBA-Transformers-QCS	0.5399	**0.5496**	0.5446

3.2 Main Evaluation Results

We report the evaluation results in Table 2. We observe that even though CARS leverages the context information and is trained with multi-task learning, the BERT ranker *without* any context outperforms CARS. This verifies the capability of BERT in single-turn ranking. Besides, BERT-Concat-QC(S) boosts the performance for BERT substantially, suggesting history behaviors can benefit ranking. It also shows that concatenating history turns in a BERT based model is effective despite its simplicity. However, it seems that history queries are less informative or harder to exploit than clicks since BERT-Concat-Q performs on par with BERT. Finally, HBA-Transformers outperforms CARS by a large margin. Moreover, even though BERT-Concat is a very strong baseline, our method demonstrates statistically significant improvement over it with $p < 0.05$ tested by the Student's paired t-test. This shows the strength of our approach in modeling user behaviors.

3.3 Ablation Analysis

In this section, we conduct an ablation analysis to investigate the contributions of model components and design choices.

3.3.1 ***Impact of Skipped Documents and History Window Size.*** We present the investigation results in Table 3. Although we show that history user behaviors are informative in Section 3.2, we observe that providing different amount of history to the models does not show major differences. This could be explained by a property of the AOL data that many sessions are not made of a sequence of strictly evolving queries. More analysis on this property is presented in Section 3.4. In terms of the skipped documents, we see no consistent impacts. This could be due to the fact that the skipped document in the AOL data is synthetic instead of being recorded from real SERPs [3]. This is a limitation of the AOL data.

3.3.2 ***Impact of Hierarchical Behavior Attention and Behavior Aware Embeddings.*** Our method without hierarchical behavior attention is essentially the same with BERT-Concat. In Table 4, we observe that our models with behavior aware embeddings have better performance than those without, although the statistical

[1] The number discrepancies of CARS come from different tie-breaking strategies in evaluation. We use trec_eval while Ahmad et al. [3] uses an author-implemented evaluation.

Table 4: Ablation. ‡ and † means statistically significant *decrease* compared to the previous line with $p < 0.05$ and 0.1.

Models	-QC	-QCS
Full Model	**0.5450**	**0.5446**
Without Behavior Aware Embeddings	0.5399‡	0.5432†
Without Hierarchical Behavior Attention	0.5340‡	0.5366‡

significance is not strong in "-QCS". The hierarchical behavior attention, on the other hand, contributes statistically significant improvement to the performance with and without the skipped documents. These results show that although behavior aware embeddings are helpful, the hierarchical architecture of the behavior attention mechanism is the primary source of effectiveness.

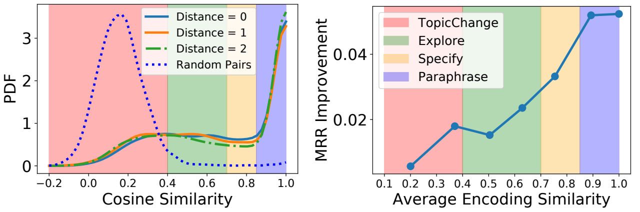
3.4 Analysis of Conversational Properties

We analyze the conversational properties of queries to show the implications of our research in conversational search. We use the MS GEN Encoder[2] to encode the queries into representations. Authors of GEN Encoder build a taxonomy for query relations, defined by the cosine similarity of the representations of query pairs [17]:

- **Topic Change** (≤ 0.4): the two queries talk about different issues.
- **Explore** $(0, 4, 0.7]$: the second query explores around the first.
- **Specify** $(0.7, 0.85]$: the second query drills down the first.
- **Paraphrase** $(0.85, 1]$: the two queries share the same intent.

We conduct two parts of studies. First, we follow Zhang et al. [17] to analyze the distribution of cosine similarities between query pairs as shown in Figure 2a. Queries within each pair can be adjacent, separated by one or two other queries in the session, or paired randomly. We observe that random query pairs often talk about different topics, as expected, while queries in the same session are much more related. Besides, for same-session queries, many of them are similar to each other, despite having different distances. Compared with Bing queries [17], AOL data has more paraphrasing queries and less queries that can form an information seeking chain. This supports our finding in Section 3.3.1 that introducing different amount of history has similar contribution to the performance.

Second, we inspect the effect of our method on queries with different conversational properties. Given the current query, we compute the cosine similarity between the representations of this query and each previous query within the history window. These similarity scores are averaged to provide a characterization of how *conversational* a query is. We adopt the same guideline to interpret the conversational property and analyze its correlation with the improvement of performance. The results are shown in Figure 2b. The MRR improvement is computed as the performance gain of HBA-Transformers-QC over BERT in Table 2. The average encoding similarity scores are grouped by quantiles. We merge the last four quantiles since their similarity scores are very similar. The MRR improvement comes from the history user behaviors that are considered by our model. We observe that the MRR improvement gets progressively larger as the session becomes more and more coherent, suggesting that coherent sessions are more demanding in considering the history user behaviors in the session. For paraphrasing queries, the clicked/skipped documents could be different and thus can contribute to the performance. This analysis indicates that coherent sessions are more conversational and should be targeted by behavior aware models.



(a) Distribution of CosSim. **(b) MRR Improvement.**

Figure 2: Analysis of Conversational Properties.

4 CONCLUSIONS AND FUTURE WORK

In this paper, we propose a hierarchical behavior attention mechanism for contextual document ranking. We show that although a simple concatenation of history user behaviors is effective, a behavior aware hierarchical architecture is more powerful. Moreover, we show that coherent sessions tends to be more conversational and thus are more demanding in considering history user behaviors. Future works will consider more diverse user behaviors and will use a larger range of test collections.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval. 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] E. Agichtein, E. Brill, S. T. Dumais, and R. Ragno. Learning User Interaction Models for Predicting Web Search Result Preferences. In *SIGIR*, 2006.
[2] W. U. Ahmad, K.-W. Chang, and H. Wang. Multi-Task Learning for Document Ranking and Query Suggestion. In *ICLR*, 2018.
[3] W. U. Ahmad, K.-W. Chang, and H. Wang. Context Attentive Document Ranking and Query Suggestion. In *SIGIR*, 2019.
[4] P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In *SIGIR*, 2012.
[5] B. Carterette, P. D. Clough, M. M. Hall, E. Kanoulas, and M. Sanderson. Evaluating Retrieval over Sessions: The TREC Session Track 2011-2014. In *SIGIR*, 2016.
[6] Z. Dai and J. P. Callan. Deeper Text Understanding for IR with Contextual Neural Language Modeling. In *SIGIR*, 2019.
[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, 2018.
[8] J. Gao, X. He, and J.-Y. Nie. Clickthrough-based Translation Models for Web Search: from Word Models to Phrase Models. In *CIKM*, 2010.
[9] C. Van Gysel, E. Kanoulas, and M. de Rijke. Lexical Query Modeling in Session Search. In *ICTIR*, 2016.
[10] J. Huang, W. Zhang, Y. Sun, H. Wang, and T. Liu. Improving Entity Recommendation with Search Log and Multi-Task Learning. 2018.
[11] R. Jones and K. L. Klinkner. Beyond the Session Timeout: Automatic Hierarchical Segmentation of Search Topics in Query Logs. In *CIKM*, 2008.
[12] R. Nogueira and K. Cho. Passage Re-ranking with BERT. *ArXiv*, abs/1901.04085, 2019.
[13] X. Shen, B. Tan, and C. Zhai. Context-Sensitive Information Retrieval Using Implicit Feedback. In *SIGIR*, 2005.
[14] H. Wang, Y. Song, M.-W. Chang, X. He, R. W. White, and W. Chu. Learning to Extract Cross-session Search Tasks. In *WWW*, 2013.
[15] R. W. White, W. Chu, A. H. Awadallah, X. He, Y. Song, and H. Wang. Enhancing Personalized Search by Mining and Modeling Task Behavior. In *WWW*, 2013.
[16] H. Zamani, B. Mitra, X. Song, N. Craswell, and S. Tiwary. Neural ranking models with multiple document fields. In *WSDM*, 2017.
[17] H. Zhang, X. Song, C. Xiong, C. Rosset, P. N. Bennett, N. Craswell, and S. Tiwary. Generic Intent Representation in Web Search. In *SIGIR*, 2019.

[2] https://msturing.org/