Towards Neural Ranking for Mixed-Initiative Conversational Search

Hinrik Snær Guðmundsson
University of Amsterdam
Amsterdam, Netherlands
hinriksnaer@gmail.com

David Biertimpel
University of Amsterdam
Amsterdam, Netherlands
david.biertimpel@student.uva.nl

Abstract

Recent research has shown that integrating clarifying questions and answers into ranking models offers the potential to better understand users' information needs and improve document ranking. However, previous approaches only used naive ranking models (i.e. QL and BM25) so far and neural rankers remain unexplored. At the same time, neural ranking models dominate leaderboards for single-shot query tasks and bring interesting features that should also be advantageous in a conversational setup. In this work we explore how neural rankers can be extended to effectively represent clarifying question and answer in addition to the initial user query. To this end, we first try to extend conventional neural ranking models ConvKNRM and PACRR by naively aggregating FastText word embeddings. We then investigate whether contextualized word embeddings given by BERT are able to incorporate clarifying questions and answers more effectively and outperform these baselines. Lastly, we analyze how our models perform on different answer polarities (affirmation, negation, I don't know and other).

Keywords: Mixed-initiative, conversational search, information retrieval, document ranking, Qulac

ACM Reference Format:
Hinrik Snær Guðmundsson and David Biertimpel. 2018. Towards Neural Ranking for Mixed-Initiative Conversational Search. In *Information Retrieval 2, September, October 2020,Amsterdam, NL.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 Introduction

When searching the Web, users usually iterate through formulating single-shot queries and evaluating the results, until the desired outcome is found. In this scenario, users are on their own to formulating queries that best meet their information needs. This lead to a lengthy and redundant search process due to the ambiguous and multifaceted nature of queries.

In recent years, Search Engine Results Pages (SERPs) received more interactive features and voice-based assistants such as Amazon Alexa and Google Assistant gained popularity in search related tasks. Both hold the potential of rendering the users search experience to be more interactive and intuitive.

One way to achieve this is to let the system proactively interact with the users by asking clarifying questions about the previous query. In this way, the system can guide users through their search by gradually gathering more information about their information needs. This turns the *single-initiative* search into a *mixed-initiative* conversational search.

When it comes to retrieving documents, the ability for the system to respond with a clarifying question and include the user's response in the search can improve the ranking of documents [9]. Using this feedback can however be challenging due to the noisy nature of natural language in mixed-initiative conversations. Also, clarifying questions are non-trivial to generate or retrieve [1, 22]. Moreover, questions that only include partially relevant information are likely to receive negative user feedback or user feedback that provides no additional information [9]. Recently, with the release of the *Qulac* data set, Aliannejadi et. al. [1] introduced an offline evaluation framework for this task, by augmenting each query with a matching clarifying question and answer. This eliminates the aspect of free clarification question generation which vastly reduces the complexity of the task.

In the last years, deep neural networks have seen significant adoption in a wide range of NLP and IR related tasks. Neural ranking models like DRMM [5], ConvKNRM [3], and PACRR [7] showed some success capturing relevance features and integrating context. Currently, information retrieval leaderboards are dominated by ranking models using transformer-based contextualized word embeddings like ELMo [17] or BERT [4]. BERT's architecture enables a more meaningful representation of the relevance between query and document as it is context-aware and can establish higher level relationships through its multi-level attention mechanisms. Despite these developments, the problem of utilizing

Information Retrieval 2, September, October 2020, Amsterdam, NL

ACM ISBN 978-1-4503-XXXX-X/18/06...$0.00
https://doi.org/10.1145/1122445.1122456

clarifying questions and answers for *mixed-initiative* conversational search is far less explored. Neural models with their capabilities should find more elaborated representations of query, question and answer and thus have a positive impact on ranking performance. In particular, BERT's ability to model context should permit to handle complex semantics like negation better than conventional neural models.

In this paper, we investigate how clarifying question and answer can improve the performance of neural ranking methods using the *Qulac* dataset. We focus on finding effective representations of initial query, question and answer so that neural rankers can make best use of this additional information. We explore the benefits of contextual word representations like BERT, and compare it to conventional embeddings such as Word2Vec [14], Glove [16] and FastText [12], which are used by more conventional neural ranking methods such as ConvKNRM [3] and PACRR [7]. To this end, we compare different neural ranking methods, both with and without clarifying question and answer. Overall we aim to answer the following research questions:

- Does integrating clarifying question and answer aid ranking performance of neural rankers ConvKNRM and PACRR?
- Do we see performance differences across the answer polarities *Yes*, *No*, "I don't know" (*Idk*) and *Other*?
- What is an effective joint representation for initial query, clarifying question and answer for document ranking?
- Can transformer-based contextual language representations like BERT improve ranking performance over conventional neural rankers like ConvKNRM and PACRR?
- Does an aggregation function operating on BERT's [CLS] embeddings improve over a token level aggregation?

2 Related Work

Neural document ranking. In document ranking, the goal is to find a function that maps query-document pairs to a relevance score, which can be used to put documents into a specific order. While most NLP tasks require capturing the semantic properties of text, the challenge in document ranking is that relevance is the primary interest. In recent years, neural ranking approaches have been introduced to better capture relevance features to create more meaningful rankings for the user.

So called interaction based models try to achieve this by formulating query-document pairs as translation matrices that can be transformed to relevance scores. For extracting relevance scores from these matrices Guo et. al. introduce DRMM [5] that is based on a histogram pooling technique. Xiong et. al. present with KNRM [21] a similar approach, where a matching histogram is approximated with a kernel pooling method enabling end-to-end learning. However, these approaches are based on matching individual words and do not incorporate context. Dai et. al. [3] aim to mitigate this problem by matching n-grams instead of individual words using CNNs to combine adjacent word embeddings into an n-gram representation. PACRR [7] expands on this motivation and creates position-aware features by using recurrent layers to capture interactions between n-grams. Despite improvements, context is still not captured natively, as it is the case with the recently introduced transformer-based contextual language representations like ELMo [17] or BERT [4]. Here, identical words may have different embeddings depending on their context e.g. *firm* (company) vs. a *firm* handshake. To address this MacAvaney et. al. introduce CEDR [10] that integrates context sensitive BERT embeddings in non-context aware methods like the previously discussed DRMM and KNRM using BERTs [CLS] token. While CEDR is rather an extension for existing methods, Nogueira & Cho [15] re-purposed BERT for query-based passage re-ranking achieving state of the art performance on the on the MS MARCO passage re-ranking task.

Document retrieval with multi-query search. The primary limitation of single-shot queries for document retrieval is that the user's information need is limited by how well he can express it through a single query. Due to this limitation, researchers have introduced methods that incorporate more information into a search to provide better results for the user. These methods take into account previous queries and/or other context [2, 11]. This can in some sense be considered the first steps for the developing field of conversational search since the system is able to provide better search results by allowing the user to interact with the system. This allows the system to achieve a better understanding of the user's information need over time, which is based on information that the system acquires beyond a single-shot query.

Mixed-initiative document retrieval with clarifying questions. Real conversations are however more complex and include mixed initiative [20] which can include assertions, commands, questions and prompts. Theoretical frameworks have been introduced for conversational search, for example in [18], researchers studied conversational approaches to information retrieval, presenting a theory and model of information interaction in a chat setting. This framework would utilize multi-turn interaction with a user to narrow down their specific information need. In [1], researchers introduced a framework consisting of three components: question retrieval, question selection, and document retrieval. For our work, we will be focusing primarily on the document retrieval component to provide the best possible results given the results that would be acquired from the previous two components along with the original query. Other independently implemented question retrieval and question selection components can then be used together with our document retrieval component.

In [9], researchers sought out to enable mixed-initiative by giving a system the ability to ask clarifying questions to the user, the answer and the clarifying question would then be used along with the initial query to provide a document ranking. The paper provided insights on the task of document ranking with clarification-based conversations by introducing a heuristic ranking model. This ranker was however very simple and the researchers proposed that deep neural models may be able to provide improved performance.

3 Neural Rankers for Conversational Search

In this section we describe our approaches for document ranking using initial query Q_0, clarifying question Q and answer A. First, we discuss the challenges of finding a reasonable aggregation method for Q_0, Q and A. We then describe how we adapt the conventional neural rankers ConvKNRM and PACRR to the conversational setting. Afterwards, we discuss two different approaches for extending BERT to model Q_0, Q and A. Next to a bottom-up approach where we aggregate Q_0, Q and A on a token level, we introduce a novel higher-level aggregation procedure on the BERT [CLS] embeddings using 1D convolutions.

3.1 Finding a joint representation of Q_0, Q and A

The task of finding a meaningful representation for the initial query Q_0, question Q and answer A has not been explored much. The challenge is to incorporate the additional information provided by Q and A into a ranking model such that ranking performance is improved. The neural rankers ConvKNRM and PACRR use FastText word embeddings as input, which causes us to create such representations for Q_0, Q and A. Naive aggregations such as averaging and linear combinations are likely to lose valuable information as the embeddings of Q_0, Q and A are reduced to a point estimate. However, Krasakis et. al. [9] made the observation that the initial query Q_0 (i.e. topic) is more important for ranking than Q and A and thus should be given a higher weight. This can be modelled by a linear combination to make the point estimate more expressive. The use of BERT promises to provide a more elaborate joint representation of Q_0, Q and A. However, it is non-trivial to adapt the BERT architecture to enforce a meaningful aggregation. Therefore, with simple aggregations like concatenation of Q_0, Q and A, we rely on BERT to find a meaningful representation by itself. This is most likely suboptimal, since it does not allow us to include our qualitative knowledge in the aggregation process.

3.2 Conventional Neural Models

With the conventional neural rankers ConvKNRM and PACRR we investigate which performance can be achieved when aggregating Q_0, Q and A in a rather naive manner. ConvKNRM [3] (Convolutional Kernel-based Neural Ranking Model) models n-gram soft matches for ad-hoc search. A standard ConvKNRM takes a query Q_0 and document D as inputs, both of which are embedded using FastText, which has been showed to provide improved word embeddings by utilizing subword information more effectively than previous methods. Convolutional layers create n-gram representations of these inputs to incorporate context. The results are then passed through a cross-match layer where document and query n-grams are matched. Afterwards, kernel pooling is applied and passed through a learning to rank layer . PACRR [7] (Position-Aware Convolutional Recurrent Relevance) consists of two main components: a relevance matching component that converts each query-document pair into a similarity matrix, and an architecture combining convolutional and recurrent modules that create position-aware features to capture interactions between input sequences. PACRR is similar to ConvKNRM as they are both interaction based models and focus on incorporating context.

These rankers should serve as a frame of reference to the more complex BERT-based approaches. To find a simple representation E_{qqa} for Q_0, Q and A, we perform naive aggregations of their input word embeddings. We employ a *mean aggregation* where the word embeddings for Q_0, Q and A are summed together and averaged. We will also introduce *weighted aggregation*, where we compute a linear combination of word embeddings:

$$E_{qqa} = \lambda_1 Q_0 + \lambda_2 Q + \lambda_3 A, \tag{1}$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Note that the *mean aggregation* is just a special case of the *weighted aggregation*. We then use the new embedding E_{qqa} as the input for ConvKNRM and PACRR.

3.3 BERT Based Neural Rankers

Contextual language representations like BERT capture the context of each word. Bert uses transformers, an attention mechanism that learns contextual relations between words in a text. It is comprised of an encoder and a decoder layer where the encoder reads text input and the decoder produces a prediction for a task. The encoder reads an entire sequence of words at once which allows the model to learn the context of a word based on its surrounding words. While Word2Vec, FastText and Glove will encode identical words with identical embeddings, BERT generates word embeddings based on each word's context. This means that BERT can encode the same word with different embeddings as long as the context where the word is used is different. Using the same example as in section 2, the word "firm" is represented differently when the context relates to a company compared to when it refers to something being solid or rigid. This motivates using BERT for our conversational search setup as we assume BERT to better capture affirmation, negation or other contextual subtleties, than conventional word embeddings.

A general problem with using BERT for document ranking is that it's pre-trained on a semantic matching task. However, previous research has shown that in order to perform well in document ranking, the model must learn features that are appropriate for relevance matching [5]. To learn these features, BERT needs to be fine-tuned on a document ranking task. To achieve this we use the approach used by MacAvaney et. al. [10] which use BERT's [CLS] embeddings as input for a ranking function. This way the gradients from the relevance prediction backpropagate through BERT, fine-tuning the produced word embeddings.

To transform this approach to our conversational setting, we propose two approaches. First, we aggregate Q_0, Q and A on token level by performing concatination with the [SEP] token in between.

$$T_{qqa} = T_{Q_0} \times \texttt{[SEP]} \times T_Q \times \texttt{[SEP]} \times T_A \tag{2}$$

where T_{Q_0}, T_Q and T_A are the tokens of Q_0, Q and A and × stands for concatenation. We can now use T_{qqa} like previous BERT based ranking approaches use T_{Q_0}. This means we first combine T_{qqa} with the document tokens T_{doc},

$$T_{qqa,doc} = T_{qqa} \times \texttt{[SEP]} \times T_{doc} \tag{3}$$

and use $T_{qqa,doc}$ as the input for BERT. Note that due to BERT's limited input size of 512 tokens, most documents are too long to be processed as a whole. This is circumvented by splitting the documents, separately applying the forward pass on the splits and finally averaging the resulting [CLS] embeddings. The final output [CLS] embedding is then processed by a single linear layer predicting relevance scores y_{rel}. With aggrigation for Q_0, Q and A on a token level, BERT learns an implicit joint representation in a bottom up manner, steered by the relevance gradients. In this approach we introduce as little human bias as possible and let BERT find the optimal representation of Q_0, Q and A by itself.

In contrast to this, we explore if it's beneficial to model and learn a more explicit aggregation using higher level features. To achieve this, we separately embed Q_0, Q, A and the document D using BERT and then find a joint representation using 1D convolutions. Separately encoding Q_0, Q, A and D provides us with $Q_{0,cls}$, Q_{cls}, A_{cls} and D_{cls} which are the respective BERT [CLS] embeddings. To find a joint representation for Q_0, Q and A we concatenate $Q_{0,cls}$, Q_{cls}, A_{cls} and process it with a function f_{qqa} that consists of three $3x1$ 1D convolutions

$$E_{qqa} = f_{qqa}\left(Q_{0,cls} \times Q_{cls} \times A_{cls}\right) \tag{4}$$

The resulting joint representation E_{qqa} has the same spatial dimensions as one of $Q_{0,cls}$, Q_{cls} and A_{cls} due to spatial downsampling in f_{qqa}, but has an increased depth. With this we aim to create a more expressive representation that better captures relevance features. Simultaneously, D_{cls} is processed by f_{doc} to create a document embedding E_{doc} that matches the dimensions of E_{qqa}.

$$E_{doc} = f_{doc}\left(D_{cls}\right) \tag{5}$$

f_{doc} consists of two $3x1$ 1-dimensional convolutions that preserve the spatial dimensions of D_{cls}. Afterwards, we concatenate E_{qqa} and E_{doc} and use a function f_{ranker} to predict the relevance of Q_0, Q, A and document D.

$$y_{rel} = f_{ranker}\left(E_{qqa} \times E_{doc}\right) \tag{6}$$

f_{ranker} consists of two $3x1$ 1-dimensional convolutions and one linear layer. The two convolutional layers find an representation for E_{qqa} and document E_{doc}. The linear layer uses this to predict a relevance score y_{rel}. Here relevance gradients train the functions f_{qqa}, f_{doc} and f_{ranker} but also backpropagate through the [CLS] embeddings to fine-tune BERT. This procedure explores whether it is beneficial to apply an aggregation learned from relevance information to higher level BERT embeddings compared to the lower level aggregation described above.

4 Experiments

In this section, we first describe the Qulac dataset that we use to incorporate clarifying question Q and answer A into our models. Afterwards, we introduce our baselines which we compare against our models as well as covering our experimental setup. We then quantitatively assess all our introduced models and analyse performance differences across different answer polarities. Finally, we critically compare the performance of our neural models with our naive BM25 baseline.

4.1 Dataset

We use the Qulac dataset to train and evaluate our models. Qulac is built on top of the TREC Web Track 09-12 collections and has been extended to include clarifying questions Q and answers A next to the original query Q_0 (i.e. topic). This addition enables the modelling of a mixed-initiative conversational setup.

For our experiments we only use a small subset of the documents in the TREC Web Track 09-12 collections. This subset contains 198 topics with $10k$ documents per topic. Due memory issues we rank these $10k$ documents using a simple query likelihood and only use the top $1k$ document for our ranking task. With this we effectively perform a document pre-ranking before our actual ranking task.

For our training we split up the dataset into a training and validation set. For testing, as with the setup in [9], we create a test split which we divide into four answer polarities:

- *Yes* - affirmation
- *No* - negation
- *Idk* - I don't know
- *Other* - other polarities

The answers of positive polarity (i.e. *yes*), these answers confirm either completely or to a certain extent that the

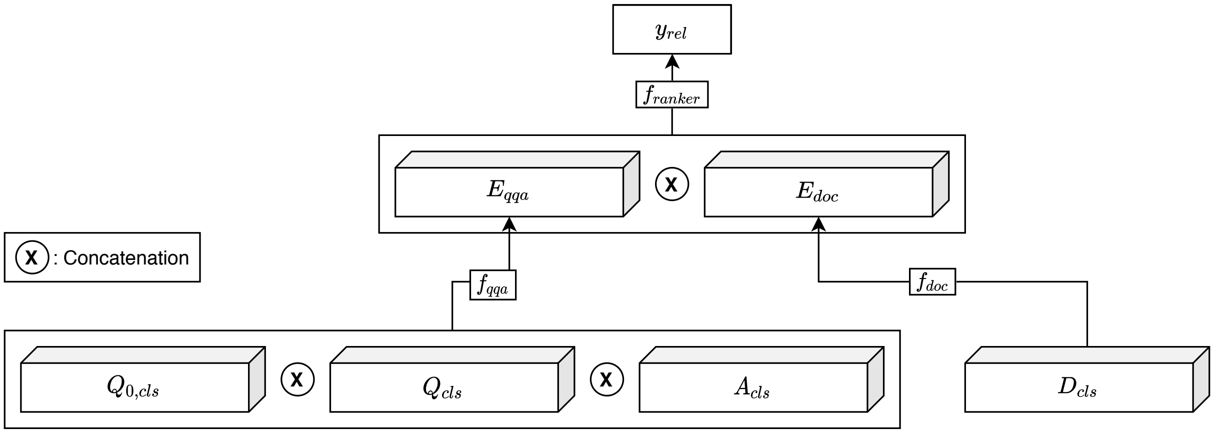

Figure 1. High level architecture for the 1D-convolution approach for intergrating clarifying question and answer into BERT.

information asked for in the clarifying question is actually close to the user's information needs. Answers of negative polarity (i.e. no) express fully or to a degree that the information that is being asked about in the clarifying question does not fit the user's information need. Answers in the *idk* polarity include "I don't know", where the user expresses either uncertainty that the information that is being asked about in the clarifying question fits the user's information need, or that the information being asked about in the clarifying question is too personal. The remaining answers are put in *Other* and do not fit the previous three polarities.

4.2 Baselines

To check if incorporating clarifying question Q and answer A leads to an improvement, we train baseline versions for all our models (i.e. ConvKNRM, PACRR and BERT). These baselines will only take the initial query Q_0 as input and do not consider clarifying question Q and answer A. For our BERT baseline, we concatenate initial query Q_0 and document D at token level using the separator token [SEP]

$$T_{Q_0,doc} = T_{Q_0} \times \texttt{[SEP]} \times T_{doc} \tag{7}$$

Further, as a naive baseline we employ BM25.

4.3 Experimental Setup

Task. We perform a re-ranking task, where BM25 provides us with an initial top-k ranking and the neural ranking models performs document re-ranking. During training, validation and testing we set k to 100, 20 and 50, respectively.

Metrics. During training we use nDCG@20 as our validation metric. We employ early stopping and terminate training if no improvement is achieved after 20 consecutive epochs. For testing we use nDCG@1 and nDCG@20.

Training Details. All of our models use are trained using the Adam optimizer [8] with a learning rate of 0.001. During training the batch size is set to 16 and one epoch holds 32 batches. ConvKNRM and PACRR have a testing batch size of 64, BERT has a testing batch size of 1. We use a maximal query length of 30 and document length of 475. For training we use a conventional cross-entropy loss. Negative sampling is used with a positive/negative ratio of $1 : 1$. To create vector representations for ConvKNRM and PACRR, we use FastText [13] (i.e. `wiki-news-300d-1M` - One million word vectors trained on Wikipedia 2017, UMBC webbase corpus [6]). For the BERT encoder we use a learning rate of $2 \cdot 10^{-5}$.

4.4 Performance of Neural Rankers in Conversational Search

In this section we analyse and discuss the results achieved by our neural ranking models. In table 1 we see all nDCG@20 results on the full test set and across different answer polarities. In table 2 the nDCG@1 results of the full test set are presented. To determine whether performance differences are significant, we perform t-tests between interesting model pairs. We do not calculate significance tests if the differences are very clear or if the test would not provide interesting insights. The significance tests are presented in table 3.

ConvKNRM. Looking at the results for ConvKNRM we see that both *mean aggregation* and *weighted aggregation* achieve improvements over the baseline. The improvements of the model with *mean aggregation* are rather small and in table 3 we see that the difference are not significant. However, the model using *weighted aggregation* shows large improvements over its baseline which are statistically significant. Of all models, ConvKNRM achieved the strongest relative improvements compared to its baseline. Another considerable improvement can be seen when looking at the nDCG@1 performance in table 2, where both the *mean aggregation* and *weighted aggregation* show a clear increase. table 3 shows that both improvements are significant. Given the current frame of reference of the Q_0 baseline, these results suggest that ConvKNRM can benefit from including Q and A when *weighted aggregation* is used.

			Answer Polarity			
Models	aggr.	*all*	*yes*	*no*	*idk*	*other*
BM25	Q_0	**0.444**	**0.428**	**0.457**	**0.385**	**0.429**
ConvKNRM	Q_0	0.233	0.235	0.231	0.208	0.243
	μ	0.245	0.231	0.241	0.236	0.275
	w	0.296	0.282	0.308	0.214	0.283
PACRR	Q_0	0.286	0.272	0.290	0.308	0.286
	μ	0.294	0.264	0.305	0.279	0.295
	w	0.275	0.252	0.281	0.262	0.277
BERT	Q_0	0.412	0.377	0.441	0.262	0.374
	Bottom-up	0.425	0.389	0.449	0.291	0.406
	1D-conv	0.272	0.249	0.279	0.224	0.280

Table 1. Experimental results for all tested models, table shows nDCG@20 results for all of the test sets.

Models	Q_0	$Q_0 + Q + A$
BM25	0.267	-
ConvKNRM	0.047	μ : 0.129 w : 0.148
PACRR	0.1407	μ : 0.133 w : 0.142
BERT: Bottom-up	0.232	0.283
BERT aggr. cls	-	0.108

Table 2. nDCG@1 performance of all rankers, Q_0 column highlights performance where the ranker was excusively trained on the original query while $Q_0 + Q + A$ shows performance after incorporating question and answer.

PACRR. In contrast to ConvKNRM, with PACRR we don't see considerable improvements over the baseline for both aggregations. Although PACRR with *mean aggregation* shows slight improvements, the differences are not significant (see table 3). *weighted aggregation* is even slightly worse than the baseline. For this reason, we cannot see a coherent trend in the performance of the aggregation methods. It is further interesting to observe that the baseline of PACRR is considerable higher than ConvKNRM. When looking at the nDCG@1 performance in table 2, we again see no considerable improvements over the baseline. From this we can conclude that for PACRR including Q and A with naive aggregations does not lead to better document rankings.

BERT: Bottom-up. The BERT model with bottom-up aggregation outperforms both ConvKNRM and PACRR by a large margin. Since the differences are clear, we omit significance tests to avoid cluttering table 3. While showing strong improvements over ConvKNRM and PACRR, we see only marginal improvements over its Q_0 baseline. In the table 3 we see that these differences do not reach statistical significance. Table 2 shows how the nDCG@1 score gains noticeable improvements over its Q_0 baseline. In contrast to the nDCG@20 scores, these differences reach statistical significance. Further, we also see strong improvements over ConvCNRM and PACRR with nDCG@1. These results suggest that despite performing better than conventional neural ranking models, BERT does likely not achieve to find a meaningful representation of Q_0, Q and A by itself that lead to stable ranking improvements. However, considering the nDCG@1 improvements we see that the top document in the ranking is more likely to be relevant.

1D Convolution BERT. BERT with 1D convolutions on the BERT `[CLS]` embeddings massively drops in performance compared to the bottom-up aggregation approach. As a consequence it also performs considerably worse than its Q_0 baseline. It further performs slightly worse than the conventional neural models ConvKNRM and PACRR. The results clearly show that 1D-convolutions are not able to capture and aggregate the rich BERT `[CLS]` embeddings. Since the approach not only fails to utilize the additional information from Q and A, but also causes a significant performance drop, it is most likely that the 1D convolutions discard much of the information in the `[CLS]` embeddings.

4.5 Analyzing the Effects of Answer Polarity

In the following we assess how the performance of our neural ranking approaches vary across different answer polarities. In table 1 we see nDCG@20 scores of all answer polarities in full detail. Figure 2 gives a more higher level overview only comparing the best approach of each model.

"I don't know" (*idk*) test results indicate that answers containing "I don't know" are the most difficult answers to provide accurate ranking for. This can be partially attributed to methodologies that were applied during data collection. Here the individuals were instructed to reply to personal or irrelevant questions with "I don't know". This causes the *idk* test set to include questions and answers that provided no additional information useful for the ranking.

As shown in figure 2 ConvKNRM with *weighted aggregation* and BERT with bottom-up aggregation perform notably worse on the *idk* answers than on other classes. Only the

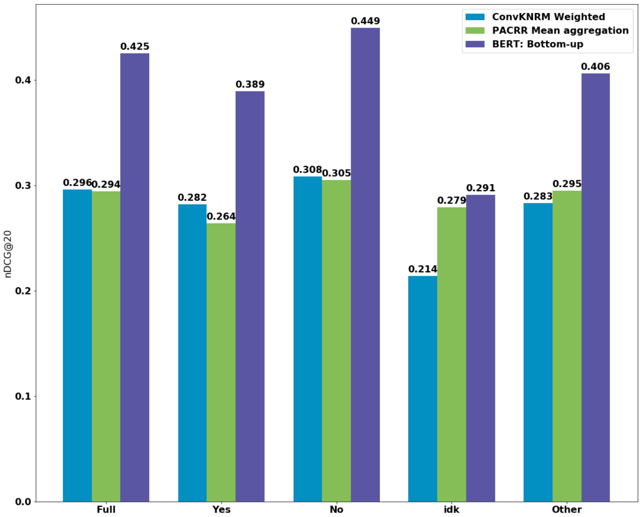

Figure 2. Results showing the best performing ConvKNRM, PACRR and BERT based approach using nDCG@20 for evaluation.

shown PACRR model does not seem to largely effected by these answer types.

Remaining polarities. While we see a clear trend in the *idk* responses, we do not see any significant tendencies for the other polarities. *Yes*, *No* and *Other* all seem to lead to similar ranking performance. Because of different experimental setups, it is difficult to compare these results to the findings of our related work. Besides the distinction between answer polarities, Krasakis et. al [9] also distinguish answer lengths. Depending on the answer length, the answer polarity has a different impact on ranking performance. For example, performance of the *No* polarity improves when the answer is longer than a simple "no". It is possible that similar trends exist in our models, but are not visible because the results are not fine-grained enough.

Correlation of polarities. Results indicate that fine tuning λ_1, λ_2 and λ_3 can impact the performance but lack of consistent improvements over all tests seems to show that weight distribution will not guarantee improvements for all possible answer types. This can be observed in the results for ConvKNRM in table 1, where *weighted aggregation* provides the best *Yes* results, while *mean aggregation* provides the best *idk* results.

BERT: Bottom-up. The model performed best out of all the neural models that incorporated question and answer when testing on the *idk* test set. The BERT: Bottom-up model was also the only neural model whose overall best performance had the best performance for every individual test set, showing more stability than other rankers. The BERT: Bottom-up approach showed the most increase in performance compared to the baseline when applied on *Other* and *idk* polarities. Such an increase in both test sets is not observed in other models and it shows how performance can be improved using clarifying question and answer despite answers having polarities that do not explicitly state affirmation or negation.

model pair	**p-value**	**t-statistic**
nDCG@20		
ConvKNRM Q_0 vs. ConvKNRM μ	0.2073	1.2612
ConvKNRM Q_0 vs. **ConvKNRM w**	$9.32 \cdot 10^{-10}$	6.1512
PACRR Q_0 vs. PACRR μ	0.5242	0.6369
BERT Q_0 vs. BERT BU	0.3569	0.9215
BM25 vs. BERT Q_0	0.0195	2.3379
BM25 vs. BERT BU	0.1700	1.3727
nDCG@1		
ConvKNRM Q_0 vs. **ConvKNRM μ**	$2.45 \cdot 10^{-12}$	7.0499
ConvKNRM Q_0 vs. **ConvKNRM w**	$2.82 \cdot 10^{-16}$	8.2578
BERT Q_0 vs. **BERT BU**	0.0043	2.8598

Table 3. Two sided t-tests between model pairs using the individual nDCG@20 or nDCG@20 scores per query. For testing significance we use a significance threshold of $\alpha = 0.01$. All bold marked models are significantly better than its comparing model. For simplicity we show the absolute value of the t-statistics.

4.6 BM25 Results vs. Neural Models.

The BM25 baseline outperforms every other model that was tested. This is an unexpected result that does not reflect results from previous work. We believe that there are different layers to this phenomenon.

First, we are working with a reduced dataset for our experiments as training and evaluating models on the full TREC Web Track 09-12 collections would be infeasible. Due to further computational constraints we reduced the $10k$ documents per topic to $1k$ documents. This leads to the fact that multiple topics were missing facets, since facets for which no relevant documents are available are not considered during evaluation. This results in a considerable shift in data distribution. Also the data became less complex, which we assume to be the reason for this dramatic performance increase. As a further sanity check we used our BM25 implementation and performed ranking on the TREC Robust 2004 dataset [19].

Here, we obtain an nDCG@20 score of 0.426 which is consistent with the results of previous work running the same baseline [10]. This underlines that our BM25 implementation and evaluation pipeline is correctly implemented.

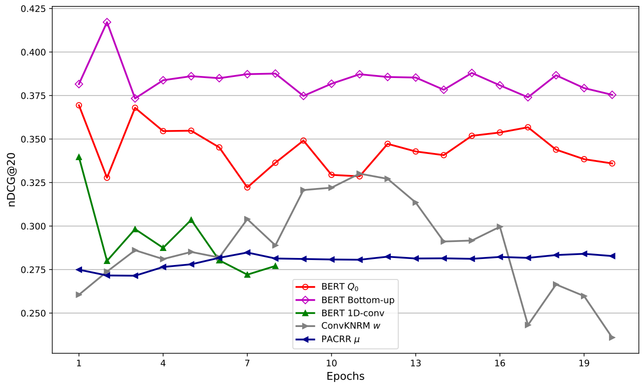

Figure 3. Validation nDCG@20 scores during training.

However, this does not explain the drop in performance of the neural ranking models, since one would assume that the neural rankings would perform even better than BM25. Is particularly applies because we perform the document re-ranking and the neural rankers receive the BM25 ranking as initialization. Nevertheless, we argue that the lower performance of the neural rankers can be explained by considerable overfitting. First, since we use only a fraction of the data, it is plausible that the expressive neural ranking models overfit the few available training examples. Further, the Qulac dataset possesses some redundancies as different permutations for Q_0, Q and A exist. This means the same query Q_0 will occur with multiple different clarifying questions Q. The same holds for clarifying questions and answers. This further reduces the effective variance in the data. A neural ranking model will see often see similar input when only one of Q_0, Q and A changes.

Further evidence for over-fitting can be found in figure 3 which shows the nDCG@20 validation scores. Here we observe that the values reach their peak in the early epochs and then decline or stagnate. This is most noticeable in the BERT-based rankers, which reach their best score after the first two epochs. These are hints for overfitting behavior. These aspects give us reason enough to be curious about how our models would behave when trained on the entire Qulac data set.

5 Limitations & Future work

The performance of the BM25 baseline compared to the neural models indicates that further exploration of the models using a larger dataset might provide additional insights with more accuracy. This would make training require more epochs to converge and is likely to generalize better.

Our results highlight the importance of finding a good aggregation method to incorporate clarifying question and answer into conventional neural models for ranking. The lack of consistent improvements over all polarity tests using different aggregation methods shows the need for more exploration into the full effects of different weight distribution for *weighted aggregation*. Aggregation methods can be used to acquire the expected input for conventional neural rankers but a more fine grained approach with some modifications to the neural model itself might be able to provide more significant improvements. One approach could be turning the weights λ_1, λ_2 and λ_3 into trainable parameters which can be implicitly optimized by the relevance gradients of the ranking loss. This could provide additional insights into the importance of Q_0, Q and A for ranking performance.

The results for BERT with bottom-up aggregation shows that BERT can improve performance overall but more analysis needs to be performed to effectively incorporate clarifying question and answer. Taking a step back and analyzing the BERT attention maps might provide insights that allow for a modified variation of BERT that utilizes specialised representations for incorporating clarifying question and answer. Doing this might provide a model that is able to deliver more notable improvements in performance.

6 Conclusion

With this work we have shown that mixed-initiative for neural rankers can be enabled in a conversational search setup without major modifications of the models themselves. Conventional neural rankers can be extended to the conversational setup by applying aggregation methods to vector representations of the Q_0, Q and A so that they match the expected input of the models. BERT based models can be extended by a simple bottom-up aggregation where Q_0, Q and A are concatenated at token level, or by applying 1D convolutions on BERTs `[CLS]` embeddings.

Clarifying questions and answers that had explicit and/or elaborate answers like the ones classified under *Yes*, *No* or *Other* polarity, showed the most amount of improvements. However, irrelevant questions and/or answers that provided either ambiguous or little additional information such as the ones classified under the *idk* polarity, led to less performance improvement. In some cases their inclusion even had a negative impact on the results.

Both conventional neural rankers had similar overall performance when the best performing variations of the models were compared. When comparing the results with the baseline, ConvKNRM had the most significant improvement in performance after incorporating Q and A when *weighted aggregation* was used. PACRR was better at handling *idk* polarity than ConvKNRM, but the results also showed that the improvements after the integration of Q and A depended strongly on the distribution of the weights and were overall not significant.

We also showed that BERT based approaches can achieve clear improvements over conventional neural rankers. The BERT approach with bottom-up aggregation yielded slight improvements for each type of response polarity. However, the improved performance was not significant enough to conclude that BERT is able to find a meaningful representation of Q_0, Q and A. Our results when performing 1D-convolution for aggregating rich [CLS] embeddings provided by BERT, was not competitive against the bottom-up aggregation approach and provided slightly worse results than the best performing variation of ConvKNRM and PACRR. This leads to the conclusion that this aggregation method fails to use the additional information from Q and A and is therefore not able to improve ranking performance.

References

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. Asking Clarifying Questions in Open-Domain Information-Seeking Conversations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 475–484. https://doi.org/10.1145/3331184.3331265

[2] Paul Bennett, Ryen White, Wei Chu, Susan Dumais, Peter Bailey, Fedor Borisyuk, and Xiaoyuan Cui. 2012. Modeling the impact of short- and long-term behavior on search personalization. *SIGIR'12 - Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval* (08 2012). https://doi.org/10.1145/2348283.2348312

[3] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-Hoc Search. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 126–134. https://doi.org/10.1145/3159652.3159659

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. 2016. A Deep Relevance Matching Model for Ad-Hoc Retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 55–64. https://doi.org/10.1145/2983323.2983769

[6] Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*. Association for Computational Linguistics, Atlanta, Georgia, USA, 44–52. https://www.aclweb.org/anthology/S13-1005

[7] Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. PACRR: A Position-Aware Neural IR Model for Relevance Matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1049–1058. https://doi.org/10.18653/v1/D17-1110

[8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[9] Antonios Minas Krasakis, Mohammad Aliannejadi, Nikos Voskarides, and Evangelos Kanoulas. 2020. Analysing the Effect of Clarifying Questions on Document Ranking in Conversational Search. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval.* 129–132.

[10] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized Embeddings for Document Ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 1101–1104. https://doi.org/10.1145/3331184.3331317

[11] Nicolaas Matthijs and Filip Radlinski. 2011. Personalizing web search using long term browsing history. *Proceedings of the 4th ACM International Conference on Web Search and Data Mining, WSDM 2011*, 25–34. https://doi.org/10.1145/1935826.1935840

[12] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[13] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[15] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). arXiv:1901.04085 http://arxiv.org/abs/1901.04085

[16] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*, Vol. 14. 1532–1543.

[17] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. https://doi.org/10.18653/v1/N18-1202

[18] Filip Radlinski and Nick Craswell. 2017. A Theoretical Framework for Conversational Search. 117–126. https://doi.org/10.1145/3020165.3020183

[19] Ellen M. Voorhees. 2005. The TREC Robust Retrieval Track. *SIGIR Forum* 39, 1 (June 2005), 11–20. https://doi.org/10.1145/1067268.1067272

[20] Marilyn Walker, Steve Whittaker, and Hewlett Laboratories. 1997. Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation. (09 1997).

[21] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-Hoc Ranking with Kernel Pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 55–64. https://doi.org/10.1145/3077136.3080809

[22] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating Clarifying Questions for Information Retrieval. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 418–428. https://doi.org/10.1145/3366423.3380126