MULTITASK LEARNING FOR DOCUMENT RANKING AND QUERY SUGGESTION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a *multitask* learning framework to jointly learn document ranking and query suggestion. The proposed model leverages previous queries and click-through information from the same search session and improves performance on both tasks. Our framework consists of two major components, document ranker and query recommender. Document ranker combines current query and previous session information and compares the combined representation with the document representation to rank the documents. Query recommender tracks users' query reformulation sequence considering all previous in-session queries using a sequence to sequence approach. Both components are trained across search sessions by sharing parameters through session recurrence, which encodes session information. Comprehensive experiments including rigorous comparisons with state-of-the-art techniques are performed on the public AOL search log, and the promising results endorse the effectiveness of the joint learning framework.

1 INTRODUCTION

Understanding users' information need is the key to optimize a search engine for providing relevant search results. Search engine logs have become an important resource to mine users' search intent and click behavior (Baeza-Yates et al., 2004; Croft et al., 2010). In particular, user query logs are partitioned into search sessions, i.e. sequences of queries issued by a unique user and submitted within a short time interval. In a search session, a user submits a sequence of queries and clicks on ranked documents that he/she believes can serve his/her information need. Thus, leveraging the user search behaviors within a query session, a.k.a. *context-awareness* provides useful information about user intent and helps to narrow down ambiguity while ranking documents for the current query and predicting next query that users will submit (Jiang et al., 2014). Since, both user's click behavior and query reformulation are driven by the underlying intent, we argue that by jointly modeling both tasks can benefit each other.

In this work, we propose a joint learning framework, called *multitask neural session relevance framework* (M-NSRF) to predict users' future queries and clicks. We assume, a search session is composed of several queries and corresponding clicks that can supplement a user's underlying intent for a search task and thus helps to improve document ranking and query suggestion. Inspired by recent works (Collobert & Weston, 2008; Liu et al., 2015), we consider a deep neural network approach that allows us to learn multiple tasks by sharing model parameters, latent states, and optimizing a joint objective function. The joint learning approach improves generalization on both tasks, and allows us to learn latent vector representations of user intent carrying over the whole session.

General work flow of M-NSRF is illustrated in Figure 1. Given a sequence of user queries from the same search session (collected from AOL search log), *"how to terminate a pregnancy at home", "abortions in orlando florida", "pregnancy signs"*, M-NSRF is trained to predict the next query, *"non medical pregnancy termination"* and the clicks on a set of relevant documents. We argue that understanding the true user intent can be aided by both click and query prediction task. In addition, modeling session-level latent states which carry information about previous queries and clicks help to better understand users' search intents. Our proposed framework can be adapted to any text representation and generation model and trained end-to-end on search sessions.

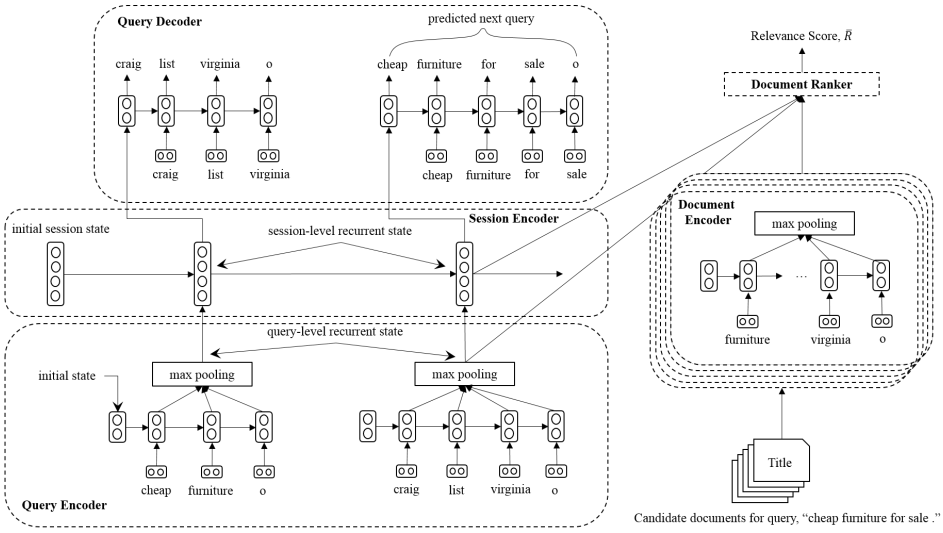

Figure 1: General work flow of the proposed intent-aware neural click model. The model is jointly trained on search sessions to predict next query and rank a set of relevant documents for the following query. The model encodes the current query in the session, *craig list virginia*, updates corresponding session-level recurrent state, maximizes the probability of the next query, *cheap furniture for sale*, encodes the candidate documents for the following query and minimizes loss for clicked documents.

We evaluate the effectiveness of the proposed framework using the publicly available AOL search log and compare with several well-known traditional retrieval models, as well as several state-of-the-art deep neural models specifically designed for ad-hoc retrieval. We also compare M-NSRF with several baseline models for query prediction task. The empirical results show that M-NSRF outperform state-of-the-art models significantly on ad-hoc retrieval task and exhibits competitive performance in query prediction task.

The key contributions of this work include:

1. We propose a novel multitask neural session relevance model that is jointly trained on document ranking and query suggestion tasks by utilizing in-session historical query and click information.
2. We conduct rigorous comparisons over traditional and neural state-of-the-art retrieval models and query prediction techniques using AOL search log and demonstrate the joint framework can improve existing models, such as Match Tensor (Jaech et al., 2017).
3. We provide detailed analysis and will release the code and data to facilitate future research.

2 Related Work and Background

A variety of approaches are proposed in literature for document ranking and query suggestion. Since our approach utilizes deep neural architectures, we only present deep neural networks based related works in this section.

Ad-hoc Retrieval. Existing neural ranking models fall into two categories: representation focused (Huang et al., 2013; Gao et al., 2014) and interaction focused (Guo et al., 2016b). The earlier focus of neural IR was mainly on representation based models (Hu et al., 2014; Shen et al., 2014), in which the query and documents are first embedded into continuous vectors, and the ranking is calculated from their embeddings' similarity. The interaction focused neural models (Hu et al., 2014; Pang et al., 2016; Guo et al., 2016a), on the other hand, learn query-document matching patterns from word-level interactions. Both the interaction and representation focused models can be combined for further improvements (Mitra et al., 2017). Similarly, Jaech et al. (2017) captures both local relevance matching and global topicality signals when computing relevance of a document to a query. In our work, we followed representation focused approach to form query and document representations and jointly model two tasks through session representations.

Query Suggestion. Deep neural models are newest addition to this literature. The closest to our work is the end-to-end hierarchical recurrent encoder-decoder architecture (HRED-qs) (Sordoni et al., 2015) to rank candidates for context-aware query suggestion. Our proposed framework mainly differs from HRED-qs as it exploits user clicks and includes it in the contextual information. Similarly, Mitra & Craswell (2015) proposed a candidate generation approach for rare prefixes using frequently observed query suffixes and suggested a neural model to generate ranking features along with n-gram features. To handle rare queries, different methods are proposed in previous works (Vahabi et al., 2013; Broccolo et al., 2012). Like other deep neural models, our proposed framework can handle rare queries as long as their words appear in the model vocabulary.

Multitask Learning. Multitask learning consider learning multiple related tasks jointly with the aim of mutual benefit. The goal of multitask learning is to improve generalization on the original task by leveraging the domain-specific information contained in the training signals of related tasks (Caruana, 1998). Multitask learning in combination with deep neural networks have been successfully used in many applications of machine learning, including natural language processing (Collobert & Weston, 2008; Liu et al., 2016), speech recognition (Deng et al., 2013; Thanda & Venkatesan, 2017) and computer vision (Girshick, 2015). Despite of wide adoption of multitask learning with deep neural networks in related areas, it has been less explored in information retrieval domain. Liu et al. (2015) proposed a multitask deep neural approach to combine query classification and document ranking and found improvements on both the tasks. Also, multitask learning is used in learning to rank in web search (Bai et al., 2009). To further investigate the utility of multitask learning, we propose a deep neural approach to jointly learn document ranking and query suggestion.

3 Neural Session Relevance Model

We propose a multitask neural session relevance framework (M-NSRF), specifically designed to jointly learn for document ranking and query suggestion tasks. We consider a session as a sequence of queries, $Q = \{Q_1, ..., Q_n\}$ submitted by a user in chronological order to satisfy a query intent. As we do not have the access to users' true intention from the query log, we assume the queries submitted within 30 minutes time frame by a user are in the same session. Every query Q_i in a session is associated with a set of relevant documents $D = \{D_1, ..., D_m\}$ which needs to be ranked by its relevance to the query Q_i. A query Q_i and a document D_j is a sequence of words $Q_i = \{w_i^1, ..., w_i^q\}$, $D_j = \{w_j^1, ..., w_j^d\}$ where $q = |Q_i|$ is the query length and $d = |D_j|$ is the document length. V is the size of vocabulary constructed over queries and relevant documents.

A detailed architecture of the multitask neural session relevance framework (M-NSRF) is provided in appendix (see A). M-NSRF is composed of two major components, document ranker and query recommender. Given all the previous information in a session and the current query submitted by a user with a set of candidate documents, the *document ranker* is trained to predict user clicks in the candidate document list. At the same time, the *query recommender* is trained in a sequence to sequence fashion (Sutskever et al., 2014) to predict the user's next query. The process is repeated for all the queries in a session. In the following, we discuss each component of the neural session relevance model in details.

3.1 Document Ranker

The document ranker in M-NSRF is made up of a query encoder, a document encoder, a session encoder and a ranker. The technical details of each constituent element is given as follows.

Query Encoder. Following Conneau et al. (2017), we use bidirectional LSTM network with max pooling technique to form query representation. Considering query as a sequence of words $Q_i = \{w_i^1, ..., w_i^q\}$, the encoder composed of forward and backward LSTM read the sequence in two opposite directions.

$$\overrightarrow{h_t} = LSTM_t(\overrightarrow{h_{t-1}}, w_i^t), \quad \overleftarrow{h_t} = LSTM_t(\overleftarrow{h_{t+1}}, w_i^t), \quad h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$$

where $h_t \in R^{2d}$ is the query-level recurrent state, d is the dimensionality of the LSTM hidden unit which is initialized to zero vector. To form a fixed-size vector representation of variable length query, maximum value is selected over each dimension of the hidden units,

$$Q_{i,k} = \max_q \ h_{k,q}, \ k = 1, ..., d,$$

where $Q_{i,k}$ is the k-th element of the latent vector Q_i.

Document Encoder. Document encoder is identical to query encoder. The only difference is the dimensionality of the LSTM hidden units. In general, documents (body or title) are lengthier than query, so we use dense vector of bigger size to represent documents.

Session Encoder. Unlike the query and document encoder, we use a vanilla LSTM for session encoding. The session encoder takes the sequence of query representations $Q_1, ..., Q_n$ as input and computes the sequence of session-level recurrent states.

$$S_t = LSTM_t(S_{t-1}, Q_t),$$

where $S_t \in R^d$ is the session-level recurrent state, d is the dimensionality of the LSTM hidden unit which is initialized to zero vector. The number of session-level recurrent states S_t is n, the number of queries in the session. The session-level recurrent state S_t summarizes the queries that have been processed up to position t.

Ranker. To rank a set of relevant documents, $D = \{D_1, ..., D_M\}$, we concatenate the current query representation Q_t with previous session-level recurrent state S_{t-1} and apply a non-linear transformation. Finally, to compute relevance score (click probability) between a query and a document, we use sigmoid function.

$$P(D_i|Q_t) = \sigma(D_i^T \tanh(W_r[Q_t, S_{t-1}] + b_r)),\ i = 1, ..., M,$$

where $W_r \in R^{(d_q+d_s)\ \times\ d_d}$ and $b_r \in R^{d_d}$ where d_q, d_s and d_d are the dimensionality of the query encoder, session encoder and document encoder hidden units.

3.2 Query Recommender

Following Sutskever et al. (2014) and Bahdanau et al. (2014), query recommender in M-NSRM predicts users' next query using a sequence to sequence approach. Basically the query recommender module estimates the probability of the next query Q_n, given all the previous queries, $Q_{1:n-1}$ up to position $n-1$ in a session as follows,

$$P(Q_n|Q_{1:n-1}) = \prod_{i=1}^{q} P(w_i|w_{1:i-1}, Q_{1:n-1})$$

In principal, a simple LSTM is the fundamental building block of the query recommender. Information about all the previous queries represented through a session vector S_t are passed to the query recommender. To this end, the recurrent state of the query recommender is initialized with a non-linear transformation of S_t,

$$h_0 = \tanh(W_q S_t + b_q),$$

where $h_0 \in R^d$ is the initial recurrent state, d is the dimensionality of the LSTM hidden unit. Then the query recommender's recurrence is computed as follows,

$$h_t = LSTM_t(h_{t-1}, w_{t-1}),$$

where h_{t-1} is the previous hidden state, w_{t-1} is the previous query term. Finally, each recurrent state is mapped to a probability distribution over the vocabulary, V using a combination of linear transformation and softmax function. Word with the highest probability is chosen as the next word in sequence.

$$P(w_t|w_{1:t-1}, Q_{1:n-1}) = g(W_p h_t + b) \tag{1}$$

where $w_t = \arg\max_w P(w|w_{1:t-1}, Q_{1:n-1})$ and $g(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}},\ j = 1, ..., K$. This formulation has shown to be beneficial for language modelling tasks (Mikolov et al., 2010).

Query Suggestion. In our framework, we consider the query suggestion task as an inference problem (Sordoni et al., 2015). Given a sequence of queries up to position $n-1$, a suggested query Q_n is:

$$Q^* = \arg\max_{Q' \in Q} P(Q'|Q_{1:n-1})$$

where Q is the space of all possible queries. To generate query suggestions of variable lengths, we use standard word-level decoding techniques (Cho et al., 2014). We iteratively consider the best prefix $w_{1:m}$ up to length m and extend it by sampling the most probable word given the distribution in Eq. (1). The process ends when we obtain a well-formed query containing the special end-of-query token.

3.3 LEARNING END-TO-END

Neural session relevance framework ranks documents and predicts next query given a query and a set of candidate documents. Therefore, while training the M-NSRF, we need to take both outputs into account. The training of M-NSRF composed of forward and backward passes. In the forward pass, M-NSRF computes the query and corresponding document encodings, updates session-level recurrent states, click probability for each candidate document and the log-likelihood of each query in the session given the previous ones. In the backward pass, the gradients are computed and the parameters are updated. All the model parameters are learned end-to-end by minimizing the sum of two loss functions, binary cross entropy loss and negative log-likelihood loss. Binary cross entropy loss is computed over click probabilities of the candidate relevant documents.

$$\mathcal{L}_1 = -\frac{1}{n}\sum\nolimits_i t_i \times log\ o_i + (1 - t_i) \times log(1 - o_i)$$

where t_i and o_i represents binary click label and click probability for i-th document. In addition, we compute negative log-likelihood of a session S:

$$\mathcal{L}_2 = -\sum\nolimits_i^{|q|} \log P(w_i | w_{1:i-1}, Q_{1:n-1}) \tag{2}$$

where current query $Q_n = \{w_1, ..., w_q\}$ is of size $|q|$ and $Q_{1:n-1}$ is all the previous queries in the session S. We observed that the probability distribution in Eq. (1) became highly peaky in initial experiments. We address this issue by regularizing the negative entropy of the probability distribution. We add the following regularization term to the loss function in Eq. (2).

$$\mathcal{L}_R = -\lambda \sum\nolimits_{w \in V} P(w | w_{1:i-1}, Q_{1:t-1})\ \log P(w | w_{1:i-1}, Q_{1:t-1}) \tag{3}$$

3.4 MULTITASK LEARNING FRAMEWORK WITH NEURAL IR MODELS

The proposed multitask learning framework is general and can be applied to combine different types of document rankers and query recommenders. As mentioned previously, our proposed framework is very close to the query suggestion model proposed in (Sordoni et al., 2015) with a multitask learning framework embedded in. So, a query suggestion model working in sequence to sequence fashion can be readily extended to our proposed multitask learning architecture. On the other hand, most of the neural IR models are built on the notion of comparing query and document representation in latent space to compute the matching degree. Extending such models to our proposed multitask learning framework is straightforward but due to distinctive nature of different neural IR models, we need to decide how to incorporate context-awareness in the final architecture.

In the experiment, we take the Match Tensor model (Jaech et al., 2017), a recently proposed neural relevance model as an example and extend it to a Multitask-Match Tensor (M-Match Tensor) model by incorporating the query recommender module to follow our proposed multitask learning framework. We take the query and document recurrence state produced by bi-LSTM following a linear projection in original Match Tensor model and feed them to the session-level encoder. In this way, we follow the same relevance computation process as in original Match Tensor model but jointly train M-Match Tensor model on document ranking and query suggestion.

4 EXPERIMENTS

4.1 DATA SETS AND EVALUATION METHODOLOGY

We conduct our experiments on the well-known publicly available AOL search log (Pass et al., 2006). The queries in this dataset were sampled between 1 March, 2006 and 31 May, 2006. In total there are 16,946,938 queries submitted by 657,426 unique users. We remove all nonalphanumeric characters from the queries, apply word segmentation and lowercasing. We follow (Jansen Bernard et al., 2007) to define the end of a session by a 30 minute window of idle time. We filtered sessions based on session length (minimum 2, maximum 10). We only keep the most frequent $|V| = 100k$ words and map all other words to an *<unk>* token while constructing the vocabulary. In total, we get 1,032,459 sessions for training, 129,053 sessions for development and 91,108 sessions for

testing. In total train, development and test sessions consists of 2,987,486; 287,138 and 259,117 queries respectively.

We consider two tasks in multitask learning framework – document ranking and query suggestion. In the document ranking task, our goal is to rank candidate documents' titles based on their relevance with the query. Since, AOL search log only contains clicks (positive examples), we selected negative examples through random sampling from the top 100 documents retrieved by BM25. Each query in the test set consists of 50 candidates including the clicked documents. However, to reduce the training time and memory use, each query in training and development set only contains 5 candidates. In the query suggestion task, we aim at predicting the next query in the same session. All evaluation and empirical analysis for document ranking used mean average precision and normalized discounted cumulative gain (NDCG) metric computed at positions one, three and ten. For query suggestion task, we computed the BLEU scores (Papineni et al., 2002).

4.2 Baselines

Document Ranking Baselines. We compare M-NSRF's performance with word-based baselines and neural ranking baselines. Word-based baselines include query likelihood model based on Dirichlet smoothing (QL) (Ponte & Croft, 1998) and BM25 (Robertson et al., 2009). In addition, following (Mitra et al., 2016), we investigated the ranking performance of a simple word embedding-based model using GloVe word embeddings (Pennington et al., 2014). To compare with neural models, we consider baselines broadly categorized in representation-focused, interaction-focused and a combination of both. Representation-focused neural baselines include: DSSM (Huang et al., 2013), CLSM (Shen et al., 2014), ARC-I (Hu et al., 2014) and interaction-focused baselines include: ARC-II (Hu et al., 2014) and DRMM (Guo et al., 2016a). A combination of representation and interaction focused models include: DUET (Mitra et al., 2017) and Match Tensor (Jaech et al., 2017). Details of these models are provided in appendix (see B). We implemented all the baseline models in PyTorch.

Query Suggestion Baselines. To evaluate performance on query suggestion task, we consider three baseline methods including Seq2seq model proposed by Bahdanau et al. (2014), Seq2seq with global attention mechanism (Luong et al., 2015) and HRED-qs (Sordoni et al., 2015). Details of these models are provided in appendix (see C). We implemented all three baselines in PyTorch and optimized using negative log-likelihood loss as in Eq. (2).

4.3 Implementation Details

The multitask neural session relevance model was trained end-to-end and we used mini-batch SGD with Adam (Kingma & Ba, 2014) for optimization with the two momentum parameters set to 0.9 and 0.999 respectively. We use 300-dimentional word vectors trained with GloVe (Pennington et al., 2014) on 840 billion of tokens to initialize the word embeddings. Out-of-vocabulary words were randomly initialized by sampling values from a zero-mean unit-variance normal distribution. All training used a mini-batch size of 32 to fit in single GPU memory. Learning rate was fixed to 0.001. We used dropout (0.10) (Srivastava et al., 2014) and early stopping with a patience of 5 epochs were used for regularization. M-NSRF is implemented in PyTorch and it runs on a single GPU (TITAN X) with roughly a runtime of 90 minutes per epoch. In general, M-NSRF runs up to 20 epochs and we select the model that achieves the minimum loss on the development set.

4.4 Evaluation Results

Document Ranking Accuracy. Table 1 shows the ranking accuracy of M-NSRF and the baseline models. M-NSRF significantly outperforms all the baselines except Match Tensor model. We suspect that because of long and complex architecture to capture immediate and larger contexts by Match Tensor model makes it more effective than M-NSRF. We developed M-Match Tensor model described in 3.4 which slightly improves its' performance. To avoid complexity in M-Match Tensor model, we did not consider session recurrence in document ranking, rather investigated the utility added by the multitask learning principle. To study the advantage of multitask learning on our proposed approach, we trained M-NSRF only on document ranking task (noted as NSRF in Table 1) and observed significant performance drop which endorses the benefits of multitask learning.

Table 1: Comparison of document ranking models over the AOL search log.

Model Type	Model Name	MAP	NDCG@1	NDCG@3	NDCG@10
Traditional IR-models	BM25	0.164	0.121	0.136	0.156
	QL	0.139	0.088	0.108	0.133
Embedding-based	ESM	0.214	0.118	0.127	0.139
Representation Focused	DSSM	0.263	0.152	0.206	0.248
	CLSM	0.465	0.369	0.441	0.482
	ARC-I	0.413	0.268	0.373	0.424
Interaction Focused	DRMM	0.277	0.221	0.242	0.267
	ARC-II	0.433	0.294	0.386	0.442
Representation and Interaction Focused	DUET	0.272	0.152	0.212	0.263
	Match Tensor	0.613	0.568	0.572	0.618
Neural Session Model (this paper)	NSRF	0.553	0.481	0.526	0.574
Multitask Model (this paper)	M-NSRF	0.581	0.523	0.568	0.614
	M-Match Tensor	0.621	0.572	0.578	0.632

Table 2: Comparison of different query suggestion models. § indicate NSRF is trained without entropy regularization in Eq. (3).

Model Name	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Seq2seq	24.5	9.7	4.5	1.9
Seq2seq with global attention	28.1	15.7	10.4	8.5
HRED-qs	26.4	13.6	7.9	5.8
M-NSRF§	26.8	14.1	8.4	6.1
M-NSRF	28.6	16.7	10.2	8.3

Previous state-of-the-art techniques DRMM and DUET architecture performed poorly in our experimental settings. We believe because of simple architecture of DRMM with few hundreds parameters, the model fall behind to show competitive performance on the evaluation dataset. On the other hand, we suspect that the use of smaller number of top character n-graphs (in our case, 5000) by DUET architecture limits its' effectiveness in modeling representation and interaction focused features to compute matching degree between query and document. In addition, use of document title while computing relevance between query and document in our experimental setting may limit the performance of DRMM and DUET architecture. It is important to note that, some negative results have been reported for document title-based ad hoc retrieval tasks (Guo et al., 2016a), so in our future work, we want to investigate M-NSRF and all baseline model's performance by leveraging document body.

Query Suggestion Accuracy Given the all the previous queries in a session, whether M-NSRF or baseline models can predict the user's next query is evaluated and results are presented in table 2. While M-NSRF and HRED-qs considers previous query information from same session, the other two baselines only consider the current query and predicts next one. Table 2 shows that M-NSRF outperformed all baselines though the performance is still far from good. From our observation, we found that Seq2seq with global attention mechanism performed relatively well but the entropy regularization technique helped M-NSRF to achieve little improvement over the model's performance. Though we found multitask learning helped M-NSRF to achieve better performance in document ranking, we could not see much gain on the query suggestion task. Adding attention over session information and personalization to improve query suggestions are two future directions, we are interested to work on. Few example of predicted queries by M-NSRF given previous queries from same session is presented in table 3.

4.5 Ablation Study on M-NSRF

We conducted experiments to better understand the effectiveness of different components in the M-NSRF. We also analyzed the impact of word embeddings and hidden units dimension in M-NSRF's performance. Our findings are presented in table 4.

Table 3: Examples of next query suggested by M-NSRF given all previous queries in a session.

Previous session queries	types of weapons of mass destruction, weapons of mass destruction, nuclear weapons
Next user query	biological weapons
Suggested next query	destructive nuclear weapons
Previous session queries	resume template, resume template free, resume word perfect template free, wordperfect com
Next user query	wordperfect resume templates free
Suggested next query	free microsoft word templates

Table 4: Ablation study for performance analysis of M-NSRF. Statistical significances are compared with NSRF's full model and presented in bold-faced. † indicates M-NSRF is trained to learn word embeddings, no pre-trained embeddings were used.

NSRF Variant	MAP	NDCG@1	NDCG@3	NDCG@10
Full model	**0.581**	**0.523**	**0.568**	**0.614**
Fixed embeddings	0.252 (-0.329)	0.182 (-0.341)	0.216 (-0.352)	0.289 (-0.325)
Learned embeddings†	0.302 (-0.279)	0.222 (-0.301)	0.261 (-0.307)	0.297 (-0.317)
Mean-pool	0.576 (-0.005)	0.515 (-0.008)	0.561 (-0.007)	0.608 (-0.006)
BiLSTM-last	0.563 (-0.018)	0.505 (-0.018)	0.541 (-0.027)	0.594 (-0.020)
M-NRM	0.553 (-0.028)	0.494 (-0.029)	0.544 (-0.024)	0.582 (-0.032)
GloVe 6B 50d	0.247 (-0.324)	0.196 (-0.329)	0.232 (-0.336)	0.296 (-0.348)
GloVe 6B 100d	0.312 (-0.269)	0.241 (-0.282)	0.273 (-0.295)	0.306 (-0.308)
GloVe 6B 200d	0.378 (-0.203)	0.356 (-0.167)	0.447 (-0.121)	0.498 (-0.116)
$Q_{128}D_{256}S_{512}$	0.562 (-0.019)	0.507 (-0.016)	0.544 (-0.024)	0.582 (-0.032)
$Q_{512}D_{1024}S_{2048}$	0.586 ($+0.005$)	0.528 ($+0.005$)	0.571 ($+0.003$)	0.617 ($+0.003$)

Impact of Different Model Components. To study the effect of different model components, we compare the full M-NSRF with several simpler versions of the model. At first, we turned off training for word embeddings and found a large drop in performance. In our training dataset, we have roughly $|OOV| = 26k$ out-of-vocabulary words and so, training the word embeddings turned out to be very important for our experimental setting. Also, we investigated the role of pre-trained word embeddings (ex., GloVe embeddings) and found significant performance drop (27.9% drop in MAP) if we train M-NSRF without any pre-trained word embeddings. Hence, we can conclude that use of pre-trained word embeddings and training them further is important to achieve better performance in M-NSRF. Secondly, we investigated the advantages of using max-pooling over mean-pooling and considering last hidden recurrent state for query and document representation. We observed that max-pooling and mean-pooling provides almost same performance while biLSTM-last approach lags slightly. To further analyze the features identified by the query and document encoders using max-pooling technique, we follow the idea of visualization proposed in (Conneau et al., 2017). We provide an example in figure 2 where *document 1* is clicked by the user (positive example) and *document 2, 3* is retrieved by BM25 and considered as unclicked document (negative examples). We observed that the query and document encoders identify distinguishing features (ex., the word *priceline* in the first document's title is most important) which help to differentiate between clicked and unclicked documents.

In another variant of M-NSRF, we did not use session recurrent state while computing relevance score for the candidate documents to examine the influence of previous information from same session on the ranking performance. We call this variant of M-NSRF as neural relevance model (M-NRM). From table 4 we can see, without session information the performance drops by 2.8% in terms of MAP. We further investigated and found session information helps particularly for longer sessions (session length > 5). In our evaluation dataset, we had roughly 5000 sessions of length greater than 5, so the performance difference is not reasonably high as we expected. We will investigate whether directly incorporating click information in the session recurrence helps in improving the performance significantly.

Impact of Dimensionality. We further study the impact of dimensionality of the word embeddings, query, document and session latent vectors. In M-NSRF, we set the dimension of query, document

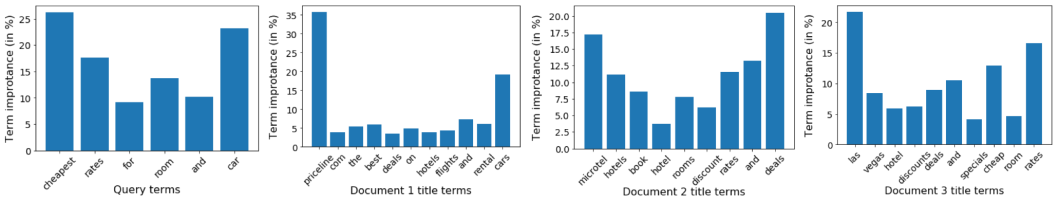

Figure 2: Example showing query and document term importance identified by M-NSRF while ranking candidate documents for the given query.

and session latent vectors as 256, 512 and 1024 respectively. As shown in table 4, decreasing the dimensions of latent vectors, drops the performance while increasing the dimensions further, does not affect the performance significantly. We also experimented with different dimensions of pre-trained word embeddings (50d, 100d and 200d GloVe (Pennington et al., 2014) embeddings). Word embeddings of different dimensionality provide different granularity of semantic similarity; with lower dimensionality, the similarity between word embeddings might be coarse and thus hard to capture matching between two text sequences. In our experiment, we found 300 dimension for embeddings works significantly better than other dimensionality.

5 Conclusions

Existing deep neural models for ad-hoc retrieval often omit session information and are only trained on query-document pairs. In this work, we propose a context-aware multitask neural session relevance framework which works in a sequence to sequence fashion and show that sharing session-level latent recurrent states across document ranking and query suggestion task benefits each other. Our experiments and analysis not only demonstrated the effectiveness of the proposed framework, but also provides useful intuitions about the advantages of multitask learning involving deep neural networks for IR tasks.

For future work, we would like to leverage document body to train M-NSRF so that we can further explore the potential of the proposed framework on ad-hoc retrieval. In addition, a broad research direction would be to find ways to summarize individual users' search log to model long-term search goals to enhance personalized search results and query suggestions.

References

Ricardo A Baeza-Yates, Carlos A Hurtado, Marcelo Mendoza, et al. Query recommendation using query logs in search engines. In *EDBT workshops*, volume 3268, pp. 588–596. Springer, 2004.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Jing Bai, Ke Zhou, Guirong Xue, Hongyuan Zha, Gordon Sun, Belle Tseng, Zhaohui Zheng, and Yi Chang. Multi-task learning for learning to rank in web search. In *Proceedings of the 18th ACM CIKM*, pp. 1549–1552. ACM, 2009.

Daniele Broccolo, Lorenzo Marcon, Franco Maria Nardini, Raffaele Perego, and Fabrizio Silvestri. Generating suggestions for queries in the long tail with an inverted index. *Information Processing & Management*, 48 (2):326–339, 2012.

Rich Caruana. Multitask learning. In *Learning to learn*, pp. 95–133. Springer, 1998.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th ICML*, pp. 160–167. ACM, 2008.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*, volume 283. Addison-Wesley Reading, 2010.

Li Deng, Geoffrey Hinton, and Brian Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 8599–8603. IEEE, 2013.

Jianfeng Gao, Li Deng, Michael Gamon, Xiaodong He, and Patrick Pantel. Modeling interestingness with deep neural networks, June 13 2014. US Patent App. 14/304,863.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE ICCV*, pp. 1440–1448, 2015.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM CIKM*, pp. 55–64. ACM, 2016a.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. Semantic matching by non-linear word transportation for information retrieval. In *Proceedings of the 25th ACM CIKM*, pp. 701–710. ACM, 2016b.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *In Proc. of NIPS*, pp. 2042–2050, 2014.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM CIKM*, pp. 2333–2338. ACM, 2013.

Aaron Jaech, Hetunandan Kamisetty, Eric Ringger, and Charlie Clarke. Match-tensor: a deep relevance model for search. *arXiv preprint arXiv:1701.07795*, 2017.

J Jansen Bernard, Amanda Spink, Chris Blakely, and Sherry Koshman. Defining a session on web search engines: Research articles. *Journal of the American Society for Information Science and Technology*, 58(6): 862–871, 2007.

Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien, and Pu-Jen Cheng. Learning user reformulation behavior for query auto-completion. In *Proceedings of the 37th ACM SIGIR*, pp. 445–454. ACM, 2014.

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

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*, 2016.

Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *HLT-NAACL*, pp. 912–921, 2015.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pp. 3, 2010.

Bhaskar Mitra and Nick Craswell. Query auto-completion for rare prefixes. In *Proceedings of the 24th ACM CIKM*, pp. 1755–1758. ACM, 2015.

Bhaskar Mitra, Eric Nalisnick, Nick Craswell, and Rich Caruana. A dual embedding space model for document ranking. *arXiv preprint arXiv:1602.01137*, 2016.

Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th WWW*, pp. 1291–1299. International World Wide Web Conferences Steering Committee, 2017.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. Text matching as image recognition. In *AAAI*, pp. 2793–2799, 2016.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th ACL*, pp. 311–318. Association for Computational Linguistics, 2002.

Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *InfoScale*, volume 152, pp. 1, 2006.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pp. 1532–1543, 2014.

Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM SIGIR*, pp. 275–281. ACM, 1998.

Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. A latent semantic model with convolutional-pooling structure for information retrieval. In *Proceedings of the 23rd ACM CIKM*, pp. 101–110. ACM, 2014.

Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *Proceedings of the 24th ACM CIKM*, pp. 553–562. ACM, 2015.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *In Proc. of NIPS*, pp. 3104–3112, 2014.

Abhinav Thanda and Shankar M Venkatesan. Multi-task learning of deep neural networks for audio visual automatic speech recognition. *arXiv preprint arXiv:1701.02477*, 2017.

Hossein Vahabi, Margareta Ackerman, David Loker, Ricardo Baeza-Yates, and Alejandro Lopez-Ortiz. Orthogonal query recommendation. In *Proceedings of the 7th ACM conference on Recommender systems*, pp. 33–40. ACM, 2013.

A MULTITASK NEURAL SESSION RELEVANCE FRAMEWORK

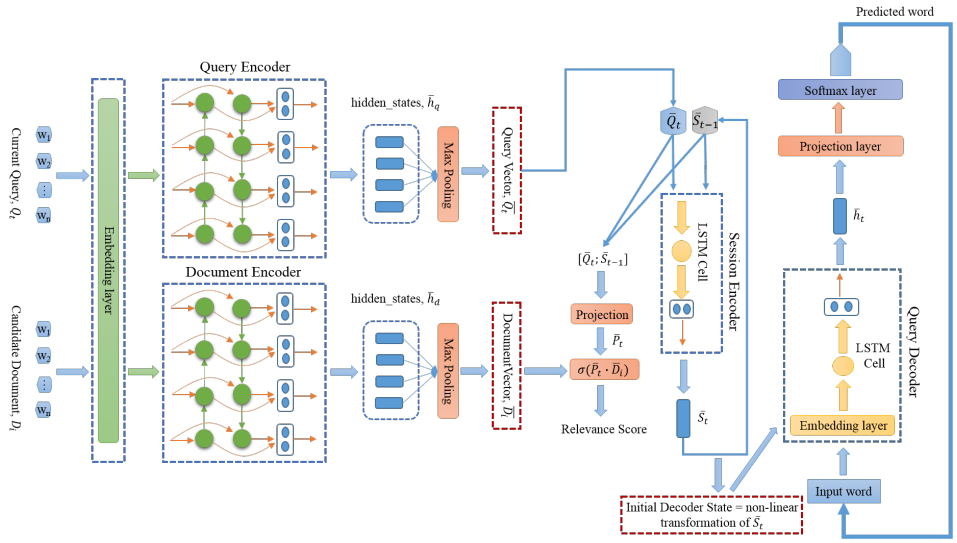

Figure 3: Architecture of the Multitask Neural Session Relevance Model (M-NSRM). M-NSRM uses bi-LSTM with max pooling to form query and document representations and use LSTM to gather session-level information. These recurrent states (current query representation and session-level recurrent state, which summarizes all previous queries) are used by query decoder and document ranker for predicting next query and computing relevance scores.

B DOCUMENT RANKING BASELINES

Deep semantic similarity model, **DSSM** (Huang et al., 2013) maps words to letter tri-grams using a word-hashing technique and uses a feed-forward neural network to build representations for both query and document. Similarity, convolutional latent semantic model, **CLSM** (Shen et al., 2014) uses word-hashing technique and uses convolutional neural networks (CNN) to build query and document representations. To compute relevance between query and document, both DSSM and CLSM uses cosine similarity. **ARC-I** (Hu et al., 2014) uses CNN to form query and document representations and employs a multi-layer perceptron to compute relevance score. In our implementation, we used 128 convolution filters of size 1, 2 and 256 filters of size 3.

ARC-II (Hu et al., 2014) was proposed by focusing on learning hierarchical matching patterns from local interactions using a CNN. To keep the ARC-II model simple, we use two layers of 2d convolution and max-pooling each and two-layer feed forward neural network to compute relevance score. **DRMM** (Guo et al., 2016a) aims to perform term matching over histogram-based features ignoring the actual position of matches. In DRMM, histogram-based features are computed using exact term matching and pretrained word embeddings based cosine similarities. In principal, the histogram counts the number of word pairs at different similarity levels. The counts are combined by a feed forward network to produce final ranking scores.

The **DUET** (Mitra et al., 2017) model composed of a local and distributed model where the distributed model projects the query and the document text into an embedding space before matching, while the local model operates over an interaction matrix comparing every query term to every document term. Similarly, **Match Tensor** (Jaech et al., 2017) model incorporates both immediate and larger contexts in a given document when comparing document to a query.

C QUERY SUGGESTION BASELINES

Seq2seq model proposed by Bahdanau et al. (2014) is a general neural network architecture that can be applied to the task where both input and output consist of a sequence of tokens. This method have been shown successful in machine translation and sequential tagging. Because different input tokens may contribute to each output token differently, attention mechanism which learns a weight between each input-output token pair can further improve the Seq2seq model. In this paper, we consider a **seq2seq with global attention** method proposed by Luong et al. (2015), which is suitable for short text such as web queries. **HRED-qs** suggested by Sordoni et al. (2015) is very close to our work which proposed to use a hierarhical recurrent encoder-decoder approach by considering session information for context-aware query suggestion.

D MORE EXAMPLES OF QUERY SUGGESTION BY M-NSRF

Previous session queries	discount pet supplies, homes for rent smyrna georgia
Next user query	homes for rent atlanta georgia
Suggested next query	pet friendly rentals in georgia
Previous session queries	language aptitude test, foreign language aptitude test
Next user query	american idol
Suggested next query	american language association
Previous session queries	saturday night fever, saturday night fever nj band
Next user query	new jersey cover band
Suggested next query	saturday night live