EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Salient context-based semantic matching for information retrieval

Yuanyuan Qi[1], Jiayue Zhang[2*], Weiran Xu[1] and Jun Guo[1]

*Correspondence:
zhangjiayue@bjut.edu.cn
[2]Beijing University of Technology, Beijing, China
Full list of author information is available at the end of the article

Abstract

Neural networks provide new possibilities to uncover semantic relationships between words by involving contextual information, and further a way to learn the matching pattern from document-query word contextual similarity matrix, which has brought promising results in IR. However, most neural IR methods rely on the conventional word-word matching framework for finding a relevant document for a query. Its effect is limited due to the wide gap between the lengths of query and document. To address this problem, we propose a salient context-based semantic matching (SCSM) method to build a bridge between query and document. Our method locates the most relevant context in the document using a shifting window with adapted length and then calculates the relevance score within it as the representation of the document. We define the notion of contextual salience and the corresponding measures to calculate the relevance of a context to a given query, in which the interaction between the query and the context is modeled by semantic similarity. Experiments on various collections from TREC show the effectiveness of our model as compared to the state-of-the-art methods.

Keywords: Semantic matching; Contextual salience; Salient context-based semantic matching

1 Introduction

The main goal of information retrieval (IR) task is to identify relevant documents for a given query, which highly depends on understanding document meanings and the search task. In the computation of relevance between a query and a document, keyword matching has been playing a dominant role [1, 2]. Typically, as BM25 [3], keyword matching combines the relevance of the keywords in a document as the relevance of the document, and identifies the keywords as the original query terms and the terms that are related to the query with a certain relevance. Although it works efficiently and has been widely applied, as a drawback, it often returns irrelevant documents due to the keyword mismatching caused by the word ambiguity issue. Actually, as we can see shortly, the meaning of text usually is defined by its context; hence, the missing evidential contextual information of keywords is the root cause of the term-mismatching problem.

To deal with this problem, varieties of neural IR models have been proposed to incorporate context information by embedded representation [2] through deep neural networks, which are often called semantic matching. Some methods consider the whole document as a global context and embed it into one vector. The query is embedded into the same vector space, and these vectors are used to calculate the relevance between the query and the document [4–6]. Other methods consider a certain scope around the keyword as the local context. Only this local context is encoded into embedding vectors and used to compute the relevance [7]. Whereas the approaches made important progress for semantic matching, space has been left for further improvement. The global context methods could not capture the individual interactions between the query and document from term level since the whole document is encoded into one vector. The local method does not have this problem, but leaves the mismatching problem unsolved, because the context still relies on the correctness of the keywords. If a keyword is a mismatch, the context around it provides evidence in a wrong direction.

The above suggests that the context chosen by each individual keyword separately is not reliable. But the question is how to find a truly relevant context. To answer the question, the idea of salient context has been proposed to build a bridge between query and document [8]. The idea is motivated by the word semantic distribution as shown in Fig. 1. It shows that rather than the document having fewer terms with strong relevance in scattered positions, the documents having more terms with weak relevance but in adjacent positions are assessed as relevant by human judges. The sequence of text with most relevant terms is supposed to be able to best convey the relevance of the document to the query, namely the salient context. A shifting window strategy is used to find the salient content in [8].

The shifting window method breaks the boundary limit of natural language sentences and paragraphs when locating the salient context and proves to be effective. However, it is not flexible and does not work well for all queries because its length is fixed. The reasons are as follows. Analyzing from the perspective of query, the fixed-length window lacks consideration from two aspects. On the one hand, the lengths of queries are different, and longer query contains more keywords. It may need longer sequence of text for a document to cover all keywords in the query. On the other hand, the semantic distributions of keywords in queries are different. Some queries' keywords have similar semantics, and some queries' keywords have large semantic distances. For the former ones, the sequence of text that can convey the relevance of the document to the query may be shorter than the latter ones. In addition, analyzing from the perspective of document, the writing style of different documents is different, and using the same-length window for all documents ignores the diversity of documents. Therefore, shifting window with adapted length is necessary and important to find the salient context. Inspired by this idea, we propose a length-adapted window method to locate the salient context, and then define a measurement to quantify the relevance of the window as contextual salience. Our goal is to find the salient context that can best represent the relatedness between the query and document. In this way, we eliminate the risk of single-keyword mismatching while including word-word interactions, thus addressing the shortcomings of the models mentioned earlier.

The paper has threefold contribution. Firstly, we present the definition of salient context and propose a length-adapted window method to find the salient context in a document

with respect to a query. Secondly, we analyze the influence of the query length and keyword semantic distribution to the locating of salient context, and accordingly give the definition and measurement of contextual salience. Finally, we propose to use the salient context as a representation of a document, which serves as a bridge between the query and document combining BM25 [3] in relevant ranking.

The rest of this paper is organized as follows. We discuss related work in Section 2 and put our work in proper context. Section 3 first describes the proposed method in detail; then, the basis of semantic matching recaps the classic keyword matching model BM25. The data, the experimental setup, the results, and the discussions of our extensive experimental results and parameters are presented in Sections 4, 5, and 6. Section 7 concludes the paper.

2 Related work

As aforementioned, the conventional lexical matching models in IR only take keywords into consideration when searching for relevant documents, which inevitably faces the term-mismatching problem. To deal with it, researchers propose to involve contextual information to find relevant matches from semantic perspective as a complimentary strategy to the keyword matching. Modeling contextual information in queries and documents has been a hot topic in IR for a long time. In recent years, besides the conventional methods, deep learning techniques have played an important role to incorporate contextual information in web search as semantic matching. Here, we discuss the flourishing publications of semantic matching in IR from three textual levels.

Firstly, we introduce the term-level semantic matching models. To start with, we brief the long-standing representation learning problem in natural language processing (NLP). On the foundation of the distributional hypothesis (Harris 1954) that semantically

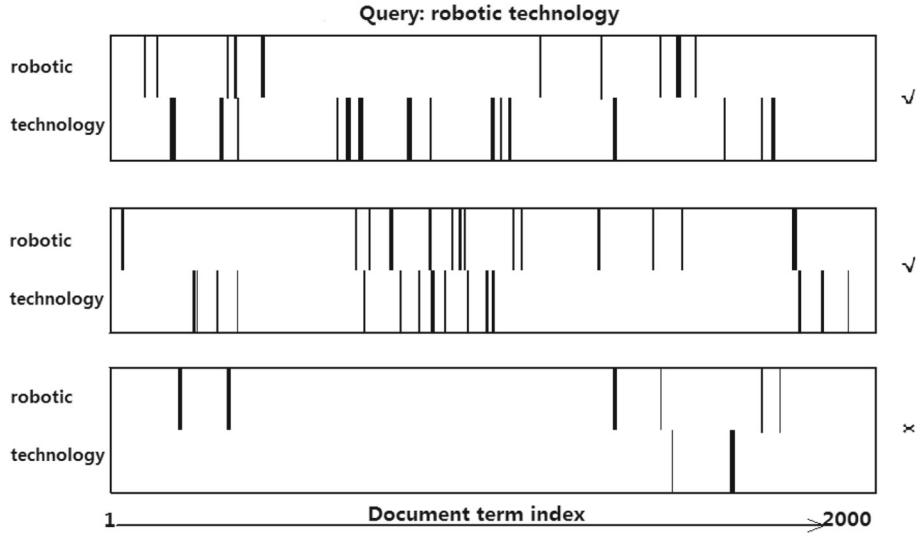

Fig. 1 Term relevance distribution. The vertical axis denotes the query term, and the horizontal axis denotes the term position index. In each box, the upper part shows the terms related to the query term "robotic," and the lower part shows that to "technology." The thickness of the line indicates the relevance score of the term, and the thicker the higher. The thickest line indicates the appearance of keyword. The first two documents are rated relevant by human judges whereas the third one is irrelevant

similar words often share similar contexts, neural network-based language models came into existence and quickly attracted extensive attention. The widely studied word embedding [9–14] leverages the information around the local context of each word to derive the distributional representation of a word. Word embeddings are typically trained from term proximity in a large corpus and aim to capture semantic similarity between terms. This ensures two words having close vector representations if and only if they are used in similar context. Thus, word embedding is directly used as a good tool for bringing semantic features into relevance matching. For example, Ganguly et al. [15] addressed the term mismatch problem by taking into account the semantic similar terms besides the exact keywords from query, while term re-weighting through word embedding is adopted in [16, 17].

Secondly, we introduce the sentence-level semantic matching models. Unlike NLP tasks, dialog generation, or question answer (QA), which have parallel size of learning objects, the comparing objects in IR are query and document, which have large gap with respect to the length. A query is short, even not a sentence, only composed of several words, while a document is long, consisting of multiple fields [18], including title, body, and anchor text. This leads to the learning complexity gap between the query and the document. To reduce the gap, most methods focus on eliminating useless content as in [4–6, 19–21] where only document titles are modeled as an extreme example. Several neural networks are adopted in these methods, such as convolutional neural network (CNN), recurrent neural network(RNN) and long short-term memory (LSTM), to map the query and the documents to semantic representations, and use the similarity between their semantic representations as the relevance between the query and the documents. However, these representation learning models perform poorly when dealing with rare terms and complex documents. Guo et al. [22] pointed out that these models performed worse when trained on a whole document than when trained on only the document title.

Thirdly, we introduce the document-level semantic matching learning (relevance matching). A long document is a mixture of different topics, and mapping all of them into one distributed semantic representation is actually anti-semantic. Rich information is condensed into one vector which makes it impossible to model the term-level interactions between query and document. Therefore, relevance matching is proposed to individually compare different parts of the query with different parts of the document, and then aggregate these partial evidences as the final relevance of the document. Guo et al. [22] designed a kind of neural networks (DRMM) to learn relevance interactions based on semantic similarity of query and document. In PACRR [23], Hui et al. proposed to use semantic similarity matrices to encode position-specific information into embeddings. McDonald et al. [24] proposed a position-aware model (PACRR + DRMM) to involve the context features into relevance matching. Co-PACRR model [7] proposed three new neural components into embedding through a cascade model, such as local context which is obtained by a local text window and global context which is the match signals in the whole document. Mitra et al. [25] incorporated the local features with the global features and jointly trained into one single vector. Ai et al. [26] put word-context information into paragraph vector model and learned the jointly embedding.

Although the local context is taken into account in these models, yet it follows the locating of a matching signal. The context is still viewed as a complimentary strategy to the keyword matching. Our proposed method emphasizes that the context should be

considered first as a basic framework, only in which the matching keywords can account for the relevance of the document. In addition, the local context explicitly reveals the semantic correlations within itself, rather than using it as a feature for embedding, and we argue that unsupervised direct calculation of the local context semantic relevance is more explainable and efficient.

3 Methods

Our goal is to find the salient context, which is a sequence that can best represent the relevance of a document to a query. Firstly, we need a metric to quantify the sequence's ability of representing the document. Secondly, we need a strategy to go through the whole document to measure all the sequences. Finally, the sequence having the highest score of the metric is the salient context that we are looking for.

In this section, we first introduce the definition of contextual salience, which is the metric mentioned above. Then, we illustrate a window-shifting method for searching the salient context through the whole document. Afterwards, we present the calculation method of contextual salience in the window-shifting framework. Finally, we show how to combine the salient context with the existing IR model.

For clarification, we outline the notations used through the paper in Table 1.

3.1 Salient context definition

According to the query-centric assumption proposed in [27], relevant information for a query only locates in the contexts around query terms in a document. This echoes our analysis of Fig. 1 that terms with high relevance to the query tend to appear around each other within a certain sequence. It implies that this sequence can best represent the relevance of the document to the query, thus a potential candidate of the salient context. To

Table 1 Notations

Notations	Description
d	Single document
q	Single query
d_j	The jth term in document
q_i	The ith term in query
$\boldsymbol{w_i}$	The ith term vector in query
$\boldsymbol{w_j}$	The jth term vector in document
dl	Document length
$avdl$	Average document length of indexed documents in data
$\mathbb{Q}$	Set of query terms
$\|\mathbb{Q}\|$	Number of query terms
$\mathbb{T}$	Set of terms in the window
$\|\mathbb{T}\|$	The window width
N	Total number of indexed documents in data
n	Number of indexed documents that contain a term
IDF	Inverse document frequency
tf	Within-document term frequency
qtf	Within-query term frequency
co	Within-document co-currency
b_1, k_1, k_3	Parameters in BM25
a,b,α, β	Parameters in our model

quantify a sequence's ability of representing the relevance of a document to a query, we propose to define a metric named contextual salience. The definition is as follows.

Definition 1 *Contextual salience: Given a pair of a query and a document, the sum of the top K strongest semantic similarities between all query terms and the terms within a width L text window of the document is called contextual salience of this window.*

When designing the concept of contextual salience, three aspects are considered. The first is motivated by the attention model. Instead of treating all terms the same way, we propose that terms having strong relationship with query terms should be paid more attention, because they make more contribution to the final relevance of the document. The second is the inevitable text noise and semantic drift in the document. Therefore, we choose to aggregate the top K highest semantic relevance as contextual salience within a text window. The third is the various document length leading to a dynamic window width. Since the contextual salience depends on the parameter K and L, we will present the mathematical representation in Section 3.3 after the illustration of searching strategy. With the definition of contextual salience, we can give the definition of salient context as follows.

Definition 2 *Salient context: Given a pair of a query and a document, a width L text window of the document having the highest contextual salience.*

3.2 Salient context searching

To find the salient context, we need to measure all sequence's contextual salience. We propose a window-shifting method to go through the whole document to make the measurement. We firstly use an example as shown in Fig. 2 to illustrate how to find the salient context from a document given a query, and then, we illustrate the process specifically.

In Fig. 2, the right box contains the given query which contains two terms, and the left side box is a list of documents as candidates. To locate the salient context, we adopt a certain-width text window moving from the beginning of the document to its end, and calculate the contextual salience of the window. The window moves forward one word per step, so there will be overlap between the windows within a certain span. For the first query term "robotic," the highly semantic related terms are labeled as orange color, such as "robot," "personalized," "quadriplegic," and "commands"; we can see these terms cluster at the second sentence in the document; the yellow color terms in the text are labeled

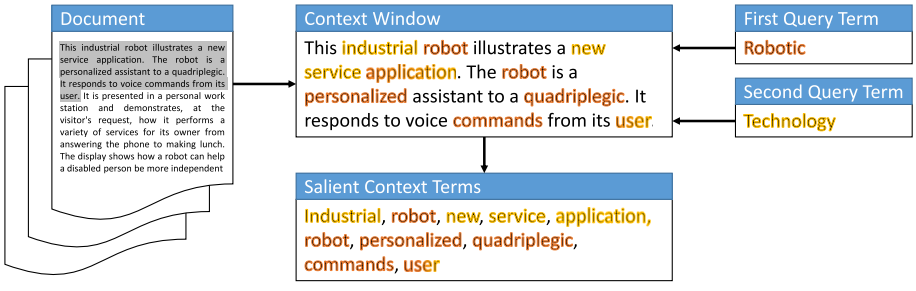

Fig. 2 Analysis of the salient context searching in a document to the query "robotic technology"

as highly semantic related terms to the second query term "technology." The top salient terms clustered at the first and the second sentence in the document. We collect these highly semantic related terms together and put them into a box called Salient Context Terms.

Based on these, we measure semantic salience of each contextual window for our model to locate the truly relevant contexts.

The window context for the query term q_i is contained in the set $\mathbb{S}_i$ given by:

$$\mathbb{S}_i = \{s_{ij} | j = \{1, \ldots, |\mathbb{T}|\}\}, \tag{1}$$

where s_{ij} is the cosine similarity between the query term q_i and the jth term T_j in the window. $|\mathbb{Q}|$ is the cardinality of $\mathbb{Q}$ and gives the number of elements in the set of queries, and $\mathbb{T}$ is the set of terms in the window.

We utilize the pre-trained word embeddings as the basis of our semantic representation on which we model the query/document matching interactions. We suppose that the query and the document are represented as a series of term vectors, respectively. We apply cosine similarity to capture the word-level semantic matching interactions, which is given by:

$$s_{ij} = \frac{(w_i)^T w_j}{\|w_i\| \cdot \|w_j\|}, \tag{2}$$

where w_i and w_j represent the pre-trained vectors of the ith query term and the jth document term. The distributional representations of text, i.e., word embeddings, encapsulate useful context information and effectively represent semantic information of a word. Models that employ pre-trained word embeddings have shown better performance compared with the relevance model which uses term co-occurrence counting between the query and the documents [7, 22–24].

In the IR problem, query plays a central role. The keyword matching methods combine the query terms' relevance in a document as the relevance of the document, and in local context methods, query terms reflect and locate the potential relevant parts of the document. In our paper, we utilize query to aggregate the highly related terms from documents to make the salience of a context stand out. We design two new functions of the window width to locate the salient context based on query length and semantic distribution.

Linear function. We compute the varying window width with a linear function based on query length as follows:

$$\text{Linear Function: } L = a \cdot |\mathbb{Q}| + b, \qquad a, b \in \mathbb{R} \tag{3}$$

where we choose the width which depends on the number of query terms, and this flexibility allows us to model fine-grained salient semantic information in the context.

Gaussian function. We compute the varying window width with a Gaussian function based on semantic distribution as follows:

$$\begin{aligned}
&\text{Gaussian Function: } L = a \cdot |\mathbb{Q}| \cdot e^{-x^2} + b,\ a, b \in \mathbb{R} \\
&x = \frac{\mu_{\mathbb{S}_q}}{\sigma_{\mathbb{S}_q}}, \\
&\mu_{\mathbb{S}_q} = \frac{1}{|\mathbb{Q}|} \sum_{s \in \mathbb{S}_q} s\,, \\
&\sigma^2_{\mathbb{S}_q} = \frac{1}{|\mathbb{Q}|} \sum_{s \in \mathbb{S}_q} (s - \mu_{\mathbb{S}_q})^2 + \delta\,, \\
&\mathbb{S}_q = \{s_{ij} |\, i, j = \{1, \ldots, |\mathbb{Q}|\}, i \neq j\}\,,
\end{aligned} \tag{4}$$

where the set $\mathbb{S}_q$ is the cosine similarity between the query terms. In Eq. 4, if the terms in the query have closer semantic similarity, the average semantic distance $\mu_{\mathbb{S}_q}$ is higher, and the greater the variance $\sigma^2_{\mathbb{S}_q}$, the greater the dispersion degree, the greater semantic difference among words in the set $\mathbb{S}_q$, the greater co-occurrence context span. δ is a minimum value to keep the variance of the query terms $\sigma^2_{\mathbb{S}_q}$ non-zero.

3.3 Contextual salience computing

With the definition of context salience and window shifting-based searching strategy, we present the mathematical description of contextual salience. We initially define $\mathbb{S}_i^{(0)} = \mathbb{S}_i$ which allows us to get the maximum values in $\mathbb{S}_i$ as:

$$\mathbb{M}_i^{(1)} = \left\{ m \in \mathbb{S}_i^{(0)} \mid m \geq s \forall s \in \mathbb{S}_i^{(0)} \right\}, \tag{5}$$

where m is a real number and $\mathbb{M}_i^{(1)}$ is a set that contains one maximum value. In our method, we define a set added by a scalar means that all members of that set is added by that scalar.

We then define $\mathbb{S}_i^{(1)} = \mathbb{S}_i^{(0)} \setminus \mathbb{M}_i^{(1)}$, and by induction, we write:

$$\mathbb{M}_i^{(n+1)} = \left\{ m \in \mathbb{S}_i^{(n)} \mid m \geq s \forall s \in \mathbb{S}_i^{(n)} \right\} \tag{6}$$

with $\mathbb{S}_i^{(n+1)} = \mathbb{S}_i^{(n)} \setminus \mathbb{M}_i^{(n)}$. Then, $\mathbb{M}_i^{(n+1)}$ is the set of the $n+1$ maximum values of the set $\mathbb{S}_i$. We then define the contextual salience, S_i^{cs}, for the query term number i as:

$$S_i^{\text{cs}} = \mathbb{M}_i^{(1)} + \alpha \frac{1}{K} \sum_{m \in \mathbb{M}_i^{(K)}} m\,, \tag{7}$$

where $K = \log(L) + 1$ is the number of maximums and is decided by the window width L. The influencing factor α balances the impact of the window context. When α is 0, only the highest relevance score represents the contextual salience. When α is 1, then the highest relevance score and the average relevance of the top K maximums equally join together to represent the contextual salience. When $0 < \alpha < 1$, the average top K relevance is discounted to form the contextual salience with a combination with the highest relevance.

Besides this consideration, it is also necessary to take the importance of terms in the query into account. In operational search, the compositional relation among the query terms is usually the simple "and" relation, but they often have different importance. Take the given query "arrested development" for example. A relevant document should refer to "arrested" and "development," where the term "arrested" is more vital than "development." There have been many previous studies on retrieval models showing the importance of term discrimination [28]. Here, we introduce an aggregation weight for each query term

as a measure of how much that term contributes to the window's final relevance score. Therefore, we define the whole contextual salience as:

$$S^{\mathrm{cs}} = \sum_{i=1}^{|\mathbb{Q}|} g_i S_i^{\mathrm{cs}}, \tag{8}$$

$$g_i = \frac{\exp\left(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_i\right)}{\sum_{m=1}^{|\mathbb{Q}|} \exp\left(\boldsymbol{w}_m^{\mathrm{T}} \boldsymbol{w}_m\right)}, \tag{9}$$

as $\boldsymbol{w}_i \in (-1, 1)^d$, with d being the dimensionality of the weight vector, the resulting scalar will be positive and equal to the square of the magnitude of $\boldsymbol{w}_i$. Eq. 9 is the normalized exponential, or softmax, function, with $0 < g_i < 1$. It returns a scalar which is proportional to the normalized magnitude of the term vector, but with an emphasis on the vectors with the largest magnitudes. Thus, it regularizes the relevance score.

3.4 Relevance aggregation

Generally, the distribution word representation highly relies on rich co-occurrence information learning mechanism, and when query terms are new or rare, salient context hardly locates highly relevant interactions. Hence, we suggest to combine with exact keyword matching rules and still give them relevance scores. In this paper, we choose to use the classic and widely applied BM25, a probabilistic formulation proposed by Robertson [3] as compensation to salient context.

BM25 considers the number of occurrences of each query term in the document and the corresponding inverse document frequency of the same terms in the full collection. BM25 focuses on studying the exact keyword matching signals over global document through learning query term frequency in the document, and applies full collection to distinguish query term influences through inverse document frequency in the full collection. The whole document terms and full collection provided as the global distribution information in the model. The classic BM25 ranking function is defined as:

$$\mathrm{BM25}(q, d) = \sum_{q_i \in q \cap d} \frac{(1 + k_1) \cdot \mathrm{TF}}{k_1 + \mathrm{TF}} \cdot \frac{(k_3 + 1) \cdot q_{tf}}{k_3 + q_{tf}} \cdot \mathrm{IDF}, \tag{10}$$

$$\mathrm{IDF} = \log \frac{N - n + 0.5}{n + 0.5}, \tag{11}$$

where $\mathrm{TF} = \frac{tf}{(1-b_1)+b_1 \cdot \frac{dl}{avdl}}$ is the pivoted document length normalization of term frequency, tf is the within-document term frequency, q_{tf} is the within-query term frequency, IDF is the inverse document frequency, dl is the document length, and $avdl$ is the average document length of collection. b_1 is a parameter used to balance the impact of document length dl.

Up to now, we have considered one single context window. The S^{cs} defined above holds the relevance score for a single context window. Next, we define the set $\mathbb{S}^{\mathrm{cs}}$ which contains all the scores for all windows in a given document. where st is the step size of our sliding window. For a document with dl terms (document length), the set thus has N elements. The number of the set is calculated as: $N = (dl - L)/st + 1$, without padding in document,

Table 2 Overview of the TREC collections used in our model

Collection name	Topics	Topic number	Docs
WT2G	401–450	50	247,491
Robust04	301–450 601–700	250	528,155
Blog06	851–950	100	3,215,171

where *st* is the step size of our sliding window. From the set of window context scores, we define the final document score as:

$$\mathcal{S} = \log(co) \cdot \max(\mathbb{S}^{cs}) + \beta \cdot \text{BM25}\,, \tag{12}$$

where *co* is the number of the co-occurrence of query terms within the document and balances the effects of salient context relevance. The parameter β balances the effects of BM25 in the relevance scoring. When β is 0, only the contextual salience contributes to the relevance scoring. When $0 < \beta < 1$, the contextual salience and BM25 both contribute to the score.

4 Datasets and evaluation

We evaluate the proposed approaches on three standard TREC collections, which are different in their sizes, contents, and topics. The TREC tasks and topic numbers associated with each collection are summarized in Table 2.

The WT2G collection is a 2-GB size of general Web documents (TREC'99 Web track), and the Robust04 contains news articles from various source, which are usually considered as high-quality text data with little noise (TREC'97–99 Ad-hoc track). The Blog06 collection includes 100,649 blog feeds collected over an 11-week period from December 2005 to February 2006. Following the official TREC settings [29], we index only the permalinks, which are the blog posts and their associated comments.

We use the TREC retrieval evaluation script[1] focusing on MAP (mean average precision), RP (recall precision) and P@5, P@20, NDCG@5, and NDCG@20 in our experiments. The MAP metric is commonly done in TREC evaluations. The MAP metric reflects the overall accuracy, and the detailed descriptions for MAP can be found in [30]. Recall values the documents that are relevant to the queries that are successfully retrieved. The P@k in the evaluation measures precision at fixed low levels of retrieved results, such as 5 or 20 documents. The NDCG is short for normalized discounted cumulative gain; it values for all queries which can be averaged to obtain a measure of the average performance of a ranking algorithm.

5 Experiment setup

For all the test collections used in our experiments[2], we apply pre-trained GloVe [13] word vectors[3] which is trained from a 6 billion token collection (Wikipedia 2014 plus Gigawords 5), because reliable term representations can be better acquired from large-scale unlabeled text collections rather than from the limited ground truth data for IR task. This setting is different from the experiment setting in the papers [22–24][4].

[1] https://trec.nist.gov/trec_eval/
[2] Our source code is available at https://github.com/YuanyuanQi/SCSM/
[3] https://nlp.stanford.edu/projects/GloVe/
[4] DRMM, PACRR, and PACRR-DRMM all apply pre-trained word vectors of GloVe in our experiments

Each query contains three fields, namely title, description, and narrative. We only use the title field that contains limited keywords related to the query. This is because the title-only queries are usually short and reveal a realistic snapshot of real user queries in practice.

In the IR task, the neural IR models rely on sufficient training data to tune model parameters; however, only query has few documents with related labels to train. Since the number of queries of some data collections is too small to tune model parameters, we only reproduce experiments on three deep learning methods (DRMM [22], PACRR [23], and DRMM-PACRR [24]) on Robust04 TREC collection. In this paper, we apply k-fold cross-validation to tune the parameters in our experiments, where k is 5. The 5-fold cross-validation is performed five times, and the mean of 5 test results is used as the final results. For the experiments of BM25, the optimized parameter values are as follows: $b_1 = 0.35$, $k_1 = 1.2$, and $k_3 = 8.0$. As for the parameters in the Eqs. 3, 7, and 12, they are also obtained by this way, and the values are listed in the corresponding positions in Section 6.

In our experiment, the baseline of BM25 model is running on an open source search engine Terrier[5]. The version we use is Terrier-3.6. Our codes run on the i7-8700K CPU @ 3.70 GHz, and the amount of RAM is 64 GB.

6 Experimental results and discussions

In this section, we present our experimental results as follows. Firstly, we show the comparisons of our model with three deep learning methods which are recently released in IR. We then show the robustness of our model by analysis on two TREC collections of different sizes. Finally, we analyze the influence of the two parameters in our model across three TREC collections.

6.1 Comparisons of deep learning methods

Table 3 shows the performance of our method SCSM (salient context-based semantic matching) on the Robust04 collection in comparison with the deep learning based methods recently proposed in DRMM [22], PACRR [23], and PACRR + DRMM [24]. SCSM_{lf} means using varying window width with linear function, and SCSM_{gf} means using varying window width with Gaussian function. CSSM_{C} means using fixed window width with Constant which was our work in [8]. In SCSM_{lf}: $a = 26$, $b = 9$; in SCSM_{gf}: $a = 17$, $b = 2$. The percentage of how much deep learning-based models and our model outperforms BM25 is also listed.

It is apparent from this table that SCSM_{lf} achieves the best performance on MAP, P@5, P@20, NDCG@5, and NDCG@20, while SCSM_{gf} achieves the best performance on RP. The second best performance on MAP, P@20, and NDCG@20 is achieved by SCSM_{gf}, while CSSM_{C} achieves the second best performance on the left two metrics: P@5 and NDCG@5.

As for the performance on MAP, the three methods that we proposed significantly outperform the four other methods; specifically, the SCSM_{lf} achieves the best performance. This can be attributed to two reasons. Firstly, compared with BM25 and DRMM, our three methods shorten the text gap between the query and the document. We use the salient context to represent the document, and the context is chosen by a shifting

[5] http://www.terrier.org/

Table 3 Comparisons of deep learning methods with MAP, RP and P@5, P@20, NDCG@5, and NDCG@20 on Robust04 collection

Corpus	Methods	MAP	RP	P@5	P@20	NDCG@5	NDCG@20
Robust04	BM25	0.239	0.283	0.481	0.354	0.497	0.425
	DRMM	0.243	0.281	0.485	0.355	0.504	0.432
		+ 1.67%	+ 0.00%	+ 0.83%	+ 0.28%	+ 1.41%	+ 1.65%
	PACRR	0.245	0.283	0.486	0.359	0.507	0.434
		+ 2.51%	+ 0.35%	+ 1.04%	+ 1.41%	+ 2.01%	+ 2.12%
	DRMM + PACRR	0.247	0.285	0.489	0.362	0.511	0.436
		+ 3.35%	+ 0.71%	+ 1.66%	+ 2.26%	+ 2.82%	+ 2.59%
	$CSSM_C$	0.262	0.304	0.496	0.376	0.508	0.445
		+ 9.94%	+ 7.72%	+ 3.18%	+ 6.19%	+ 2.13%	+ 4.69%
	$SCSM_{lf}$	0.267	0.307	0.500	0.380	0.516	0.452
		+ **11.91**%	+ 8.78%	+ **4.01**%	+ **7.43**%	+ **3.76**%	+ **6.40**%
	$SCSM_{gf}$	0.265	0.309	0.492	0.376	0.507	0.446
		+ 11.15%	+ **9.24**%	+ 2.18%	+ 6.30%	+ 1.99%	+ 5.11%

Boldface numbers indicate statistically significant improvements over BM25 by permutation test

window; then, the document is reshaped into shorter word sequence units. In this way, the document and the query share similar text length and text granularity. Thus, it decreases the information gap between the query and the document. Secondly, our model encodes keyword matching and contextual semantic matching together and aggregates them via linear scoring function while query terms are rare or new in corpus. In Robust04, over 12% queries contain low-frequency terms and around 50% query terms are not covered in related documents. Comparing with other methods in the table, the two methods with varying window width methods show the best two performances. It is positive feedback to focus on studying the query to locate and model salient contexts from whole text of documents.

As for the results on RP, we find that $SCSM_{gf}$ achieves the best performance than other six methods. In Robust04, the average length of queries is 2.73, and over half of query terms never show up in related documents. In our model, salient context clusters not only include query terms themselves but also the top ranked semantic related words of query terms. It broadens horizons of retrieval model to retrieve related documents with low frequency of exact keywords and improve the effectiveness of model retrieval. We choose P@5, P@20, NDCG@5, and NDCG@20 to analyze the ranking quality. Our method can retrieve most related documents and rank them at the top of the list, which shows better performance than other methods.

Note that the deep text matching models DRMM [23] and PACRR + DRMM [24] can lead to bad performance, because they are invented mainly for sequence around keywords from query and can hardly capture meaningful semantic interactions in article pairs. When the text is long, it is hard to get an appropriate context vector for matching. For interaction-focused neural network model DRMM [22], most of the interactions between words in query and long documents are meaningless.

In our experiment, the computation costing for BM25, DRMM, PACRR, PACRR + DRMM, and our methods are different. BM25 is the fastest one with less than 15 min, the running time of our three methods is less than 40 min, and the other neural IR methods[6] take over 6 h to tune the model's parameters. Both BM25 and our

[6]We run the codes of the neural IR models from https://github.com/nlpaueb/deep-relevance-ranking

Table 4 Comparisons of SCSM and BM25, with MAP, RP and P@5, P@20, NDCG@5, and NDCG@20 on WT2G collection

Corpus	Methods	MAP	RP	P@5	P@20	NDCG@5	NDCG@20
WT2G	BM25	0.313	0.340	0.532	0.391	0.542	0.470
	$CSSM_C$	0.368	0.378	0.600	0.428	0.616	0.521
		+ 17.82%	+ 11.09%	+ 12.78%	+ 9.46%	+ 13.63%	+ 10.88%
	$SCSM_{lf}$	0.370	0.383	0.592	0.435	0.611	0.526
		+ **18.27**%	+ **12.77**%	+ 11.28%	+ **11.25**%	+ 12.63%	+ **11.97**%
	$SCSM_{gf}$	0.370	0.381	0.616	0.426	0.628	0.521
		+ 18.01%	+ 12.03%	+ **15.79**%	+ 8.95%	+ **15.88**%	+ 11.05%

Boldface numbers indicate statistically significant improvements over BM25 by permutation test

methods save more time through tuning less model parameters than the neural matching methods. Compared with the three deep methods, our model's performance is efficient and accurate. It is minute level while the deep methods are hour level.

6.2 Different sizes of data

In this section, we discuss experiments on two different sizes, topics, and sources of TREC collections: WT2G and Blog06. Different from the Robust04 TREC collection, the query number of WT2G and Blog06 is smaller and not fit for tuning complicated deep learning-based model parameters; hence, we only compare results with the method BM25. In Table 4, in $SCSM_{lf}$: $a = 7$, $b = 7$; in $SCSM_{gf}$: $a = 6$, $b = 0$. In Table 5, in $SCSM_{lf}$: $a = 1$, $b = 2$; in $SCSM_{gf}$: $a = 1$, $b = 0$.

In Table 4, three methods perform more stable and show larger improvements on WT2G over six metrics than the other two TREC collections. This is due to the specialities of WT2G: small size as 2 GB collection, only 25% of all queries with whole query terms show up in related documents and over 75% of all queries contain high-frequency terms. General or high-frequency terms encapsulate much more semantic information than rare or low-frequency terms in distributional representations. The high-frequency terms in query can offer rich and precise context information for the salient semantic contexts in related documents to get higher score and rank more related documents in the top five lists. It also explains that the method of $SCSM_{gf}$ shows better performances than $SCSM_{lf}$ on P@5 and NDCG@5.

As for the analysis of the results on Blog06 collection in Table 5, the method of $SCSM_{lf}$ achieves the best result over all six evaluations and $SCSM_{gf}$ is the second best result. Our model shows strong robustness in our experiments of IR. The size, contents, and topics of collections have few limitations and negative effects on our model.

Table 5 Comparisons of SCSM and BM25, with MAP, RP and P@5, P@20, NDCG@5, and NDCG@20 On Blog06 collections

Corpus	Methods	MAP	RP	P@5	P@20	NDCG@5	NDCG@20
Blog06	BM25	0.318	0.371	0.634	0.605	0.625	0.611
	$CSSM_C$	0.346	0.403	0.670	0.642	0.659	0.648
		+ 8.748%	+ 8.79%	+ 5.68%	+ 6.03%	+ 5.48%	+ 6.09%
	$SCSM_{lf}$	0.349	0.408	0.694	0.657	0.684	0.665
		+ **9.75**%	+ **10.17**%	+ **9.46**%	+ **8.51**%	+ **9.58**%	+ **8.88**%
	$SCSM_{gf}$	0.347	0.405	0.682	0.646	0.672	0.654
		+ 9.06%	+'9.20%	+ 7.57%	+ 6.78%	+ 7.67%	+ 7.08%

Boldface numbers indicate statistically significant improvements over BM25 by permutation test

We bring Tables 3, 4, and 5 together to analyze and find that the method $SCSM_{lf}$ outperforms $SCSM_{gf}$ and $CSSM_{C}$. The two varying window width methods $SCSM_{lf}$ and $SCSM_{gf}$ outperform fixed constant window width method $CSSM_{C}$; it proves that it is useful to research more inner connection of query to improve the effectiveness of model retrieval in IR. Comparing the two methods with varying window width, $SCSM_{lf}$ shows better performance than $SCSM_{gf}$. Analyzing the two varying window width functions, the Gaussian method depends on the semantic distribution among query words while the linear method depends on the length of query. The semantic variance between keywords of query is very large among the three datasets; therefore, the fluctuation of the Gaussian function leads to the larger window width and brings more noise information in the window. Taking all together, these results indicate that the given query length has positive influences on analyzing the related text from documents in our relevance ranking function for the IR.

6.3 Parameter sensitivity

To illustrate the performance differences of contextual salience semantic matching and exact keyword matching in our model graphically, we pick and plot MAP results into Fig. 3.

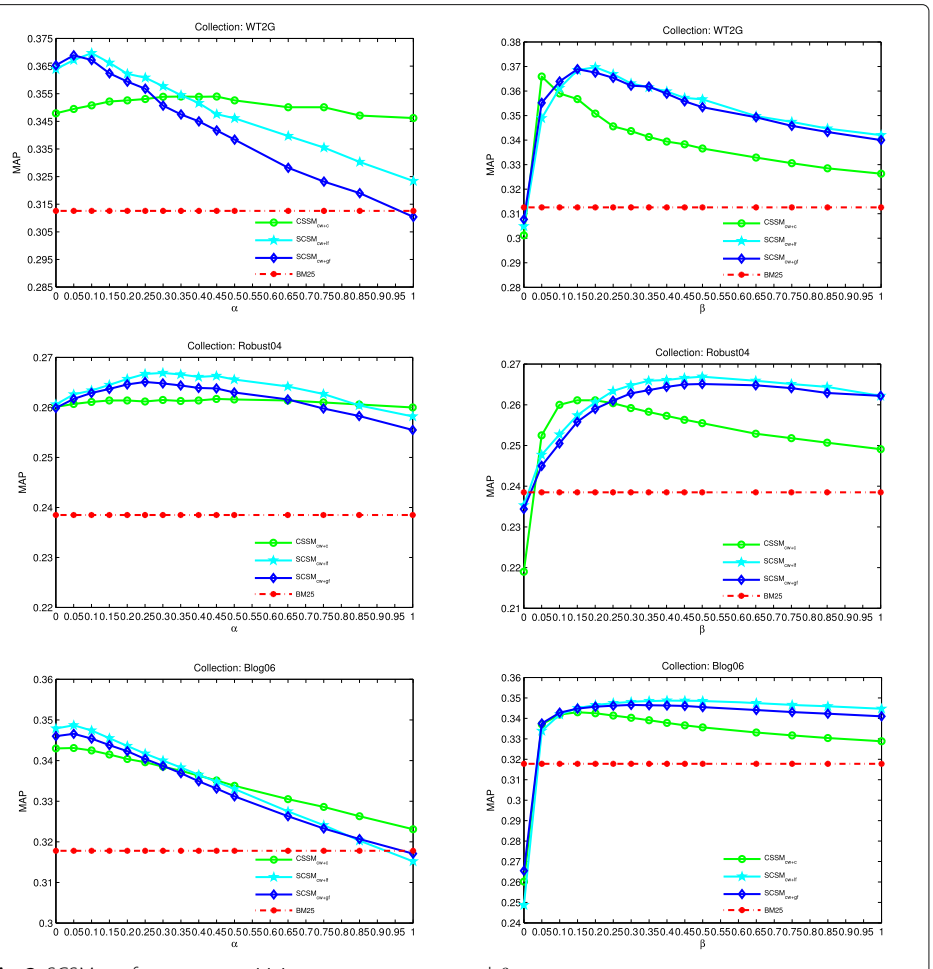

Fig. 3 SCSM: performance sensitivity over parameter α and β

On the left side of three sub-figures in Fig. 3, the curve tendency of MAP in the three TREC collections is similar: rise first and then fall with parameter α rising from 0.0 to 1.0. The middle figure of Robust04 indicates most gentle tendency in our model. When we analyze the three TREC collections, we find that the Robust04 owes the longest average length of queries. Hence, long length of queries broaden the width of salient context window and cluster rich related terms to construct contextual salience in our model. Facing complicated and rare topic of query, three methods in our model still have space to learn more potential relationship between the query and the document.

On the right side of three sub-figures in Fig. 3, the general tendency is a bit dissimilar. WT2G and Blog06 collections share similar curve tendency of the document-level exact keyword matching contribution at MAP metric: higher β, better MAP. In particular, the two varying window width methods demonstrate that traditional probability model BM25 provides positive complementary supports to contextual semantic modeling, and keyword matching shows strong robustness in studying related clues while comparing with semantic matching. Only WT2G shows adverse tendency against scoring precision while with higher influence of BM25 at document-level exact keyword matching. Fixed window width method shows highly similar tendency over the three TREC collections.

7 Conclusions

In this paper, we analyze the importance of context's semantic relevance for eliminating mismatch problem, and we define the contextual salience of a document given a query. We propose to prioritize the locating of the semantic salient context in the relevance calculation. After these, we develop an unsupervised framework to combine the query-document interactions into the contextual salience by aggregating the strongest semantic relevance interactions from term level to document level. Our method provides an efficient and explainable relevance ranking solution for IR and shows promising improvements over the strong BM25 baseline and several neural relevance matching models. The extensive comparisons between several neural relevance matching models and our approach suggest that explicitly modeling the salient query-related context in document can significantly improve the effectiveness of relevance ranking for IR.

Abbreviations

IR: Information retrieval; NLP: Natural language processing; QA: Question answer; CNN: Convolutional neural network; RNN: Recurrent neural network; LSTM: Long short-term memory; TREC: Text REtrieval Conference; DRMM: Deep relevance matching mode; PACRR: Position-aware convolutional recurrent relevance matching; Co-PACRR: Context-aware PACRR; CSSM: Contextual salient based on semantic matching; C: Fixed window width with Constant; SCSM: Salient context-based semantic matching; lf: Varying window width with linear function; gf: Varying window width with Gaussian function; MAP: Mean average precision; RP: Recall precision; NDCG: Normalized discounted cumulative gain

Acknowledgements

We thank the reviewers for their useful comments and suggestions, which helped improve the manuscript. This work was supported by the Beijing Natural Science Foundation (4174098), the National Natural Science Foundation of China (61702047), the National Natural Science Foundation of China (61703234), and the Fundamental Research Funds for the Central Universities (2017RC02).

Authors' contributions

The algorithms proposed in this paper have been conceived by Y. Qi. Y. Qi designed and performed the experiments. Y. Qi and J. Zhang analyzed the results and wrote the paper. All authors read and approved the final manuscript.

Funding

This work was supported by the Beijing Natural Science Foundation (4174098), the National Natural Science Foundation of China (61702047), the National Natural Science Foundation of China (61703234), and the Fundamental Research Funds for the Central Universities (2017RC02) and Beijing Municipal Postdoctoral Foundation and Beijing Chaoyang District Postdoctoral Foundation.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Beijing University of Posts and Telecommunications, Beijing, China. [2]Beijing University of Technology, Beijing, China.

Received: 15 October 2019 Accepted: 2 June 2020
Published online: 11 July 2020

References

1. C. Manning, P. Raghavan, H. Schütze, Introduction to information retrieval. Nat. Lang. Eng. **16**, 100–103 (2010)
2. K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, et al., Neural information retrieval: at the end of the early years. Inf. Retr. J. **21**(2-3), 111–182 (2018)
3. S. Robertson, H. Zaragoza, *et al*, The probabilistic relevance framework: Bm25 and beyond. Found Trends Inf. Retr. **3**(4), 333–389 (2009)
4. P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck, in *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management - CIKM '13*, Learning deep structured semantic models for web search using clickthrough data (ACM, 2013). https://doi.org/10.1145/2505515.2505665
5. Y. Shen, X. He, J. Gao, L. Deng, M. Grégoire, in *Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion*, Learning semantic representations using convolutional neural networks for web search (ACM, 2014). https://doi.org/10.1145/2567948.2577348
6. Y. Shen, X. He, J. Gao, L. Deng, M. Grégoire, in *A neuralprobabilistic language mode Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management: 2014; Shanghai*, A latent semantic model with convolutional-pooling structure for information retrieval (ACM, 2014), pp. 101–110. https://doi.org/10.1145/2661829.2661935
7. K. Hui, A. Yates, K. Berberich, G. de Melo, in *Proceedings of Web Search and Data Mining: 2018; Los Angeles*, Co-pacrr: a context-aware neural ir model for ad-hoc retrieval (ACM, 2018), pp. 16–27. https://doi.org/10.1145/3159652.3159689
8. Y. Qi, J. Zhang, W. Xu, J. Guo, L. Yan, Finding salient context based on semantic matching for relevance ranking. arXiv Preprint arXiv:1909.01165 (2019)
9. Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model. J. Matching Learn. Res. **3**, 1137–1155 (2013). https://doi.org/10.1162/153244303322533223
10. R. Collobert, J. Weston, in *Proceedings of the 25th International Conference on Machine Learning: 2008; Helsinki*, A unified architecture for natural language processing: deep neural networks with multitask learning (ACM, 2008), pp. 160–167
11. T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space. arXiv Preprint arXiv:1301.3781 (2013)
12. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, in *Proceedings of the 26th International Conference on Neural Information Processing Systems: 27-30 June 1996; Lake Tahoe*, Distributed representations of words and phrases and their compositionality, (2013), pp. 3111–3119. https://doi.org/10.5555/2999792.2999959
13. J. Pennington, R. Socher, C. Manning, *Glove: global vectors for word representation* (Association for Computational Linguistics, 2014). https://doi.org/10.3115/v1/d14-1162
14. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, in *arXiv Preprint arXiv:1810.04805*, Bert: pre-training of deep bidirectional transformers for language understanding, (2018)
15. D. Ganguly, D. Roy, M. Mitra, G. J. F. Jones, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15*, Word embedding based generalized language model for information retrieval (ACM, 2015). https://doi.org/10.1145/2766462.2767780
16. G. Zheng, J. Callan, in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15*, Learning to reweight terms with distributed representations (ACM, 2015). https://doi.org/10.1145/2766462.2767700
17. G. Zuccon, B. Koopman, P. Bruza, L. Azzopardi, in *Proceedings of the 20th Australasian Document Computing Symposium on ZZZ - ADCS '15*, Integrating and evaluating neural word embeddings in information retrieval (ACM, 2015). https://doi.org/10.1145/2838931.2838936
18. C. Zhai, J. Lafferty, in *ACM SIGIR Forum: 2017*, A study of smoothing methods for language models applied to ad hoc information retrieval, vol. 51 (ACM, 2017), pp. 268–276. https://doi.org/10.1145/3130348.3130377
19. H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, R. Ward, in *In arXiv Preprint arXiv:1412.6629: 2014*, Semantic modelling with long-short-term memory for information retrieval, (2014)
20. H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, R. Ward, Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. IEEE/ACM Trans. Audio Speech Lang. Process. **24**, 694–707 (2016)
21. X. Li, C. Guo, W. Chu, Y.-Y. Wang, J. Shavlik, Deep learning powered in-session contextual ranking using clickthrough data. NIPS Workshop: Personalization: Methods and Applications (2014)
22. J. Guo, Y. Fan, Q. Ai, W. B. Croft, in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, A deep relevance matching model for ad-hoc retrieval (ACM, 2016). https://doi.org/10.1145/2983323.2983769
23. K. Hui, A. Yates, K. Berberich, G. de Melo, in *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*, Position-aware representations for relevance matching in neural information retrieval (ACM Press, 2017). https://doi.org/10.1145/3041021.3054258
24. R. McDonald, G.-I. Brokos, I. Androutsopoulos, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Deep relevance ranking using enhanced document-query interactions (Association for Computational Linguistics, 2018). https://doi.org/10.18653/v1/d18-1211

25. B. Mitra, F. Diaz, N. Craswell, in *Proceedings of the 26th International Conference on World Wide Web: 2017; Perth*, Learning to match using local and distributed representations of text for web search (International World Wide Web Conferences Steering Committee, 2017). https://doi.org/10.1145/3038912.3052579
26. Q. Ai, L. Yang, J. Guo, W. B. Croft, Analysis of the paragraph vector model for information retrieval (ACM, 2016). https://doi.org/10.1145/2970398.2970409
27. W. H. Chung, R. W. P. Luk, K.-F. Wong, K. L. Kwok, A retrospective study of a hybrid document-context based retrieval model. Inf. Process. Manag. **43**(5), 1308–1331 (2007)
28. H. Fang, T. Tao, C. Zhai, Diagnostic evaluation of information retrieval models. ACM Trans. Inf. Syst. **29**(2), 7–1742 (2011)
29. I. Ounis, C. Macdonald, I. Soboroff, *Overview of the trec-2008 blog track*. (Glasgow Univ, United Kingdom, 2008)
30. S. Teufel, An overview of evaluation methods in trec ad hoc information retrieval and trec question answering, pp. 163–186. https://doi.org/10.1007/978-1-4020-5817-2_6

Publisher's Note