Enhanced Word Embedding Similarity Measures Using Fuzzy Rules for Query Expansion

Qian Liu*†, Heyan Huang*, Jie Lu†,Yang Gao*, Guangquan Zhang†
* Beijing Engineering Research Center of High Volume Language Information Processing
and Cloud Computing Applications, School of Computer,Beijing Institute of Technology, Beijing, P. R. China
† Decision Systems & E-Service Intelligence Research (DeSI) Laboratory,
The Centre for Artificial Intelligence (CAI), Faculty of Engineering and Information Technology,
University of Technology, Sydney, NSW, Australia

***Abstract*—Query expansion has been widely used to select additional words that are related to the original query words in the field of information retrieval. In this paper, we present a novel query expansion method that jointly uses fuzzy rules and a word embedding similarity calculation. The expansion words are generated using a word embedding method and selected according to their semantic similarity to the original query. Fuzzy rules are used to enhance the word similarity calculations and reweight expansion words. When measuring and ranking the relevance of a retrieved document, the original query and the expansion words with their weights are considered. We conduct experiments on the query expansion in document ranking tasks. Experimental results from the document ranking task show that the proposed method is able to significantly outperform state-of-the-art baseline methods.**

***Index Terms*—Query expansion, fuzzy rule, information retrieval, document ranking.**

I. Introduction

Information retrieval (IR) aims to provide a user with easy access to information of interested. Traditionally, term-based document retrieval methods generate queries that capture uses' interests from a collection of documents, and an IR system ranks the retrieval documents and returens relevant documents to the user. This method is computationally efficient and, therefore, widely used along with mature term weighting theories, such as TFIDF, BM25, and Rocchio [1], [2]. However, it suffers heavily from words mismatch problem, known as the lexical gap problem. Additionally, some queries can be too short or too ambiguous to express complete or accurate semantics. To address these problems, we propose a query expansion method to capture more semantic information about users' interests.

Query expansion aims to select new relevant words to a query to improve the performance of an information retrieval system. Typically, a set of candidate words is generated using external resources, for example, a lexicon such as WordNet [3] or the Paraphrase database [4], Wikipedia, query logs, initial feedback documents, etc. Several expansion words are selected from the candidate list, and each word is assigned a weighting. An extended query set that includes the original query words and the selected expansion words with their weights is then generated to assess the relevance of the retrieved document. However, two important questions in the query expansion process need to be answered: How should the expansion words be selected? and How should these words be reweighted considering their similarity to the original query?

This study enhances word embedding similarity measures using fuzzy rules for query expansion by using semantic similarity to select the expansion words. The word embedding method is introduced to capture word semantic similarity in expansion words' selection. Recently, word embedding has shown its power in natural language processing and information retrieval tasks [5], [6]. Unlike traditional word representation, word embedding overcomes data sparsity, high-dimensional data, and lexical gap problems by capturing semantics and syntactics through dense vectors. Semantic similarity is estimated by the cosine distance of the word vector to the original query in the vector space, which, in this case, is generated by Word2Vec [7], [8]. Fuzzy rules enhance the word embedding similarity calculations when reweighting the expansion words because most hold almost equal embedding similarity when calculated according to their cosine distance. A document ranking method is also presented that matches, then reweights, all queries to each retrieval document. The relevance of each document is evaluated using the new weights.

The main contributions of this study are:

- A new query expansion method is proposed to model users' interests with contextually associated words rather than synonyms from external resources.
- Word embedding is introduced into a basic query expansion method to better select expansion words.
- Incorporating fuzzy rules into reweighted expansion queries allows our document ranking method to return more relevant documents.

The outline of this paper follows. Section II presents a background on word embedding. Section III describes our method for an enhanced word embedding similarity measure for query expansion in document ranking tasks. Experimental results from document ranking tasks in a real-world application follow in Section IV. Section V surveys related work, and we conclude the paper in Section VI.

978-1-5090-6034-4/17/$31.00

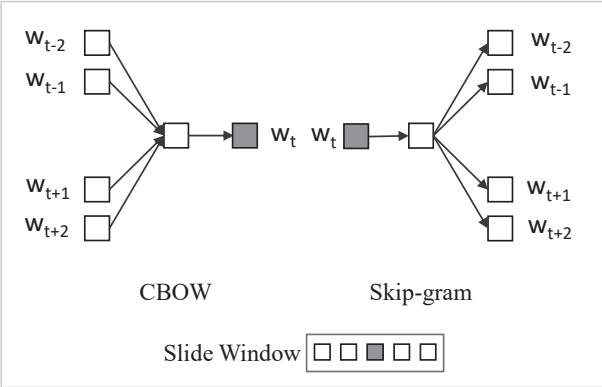


Fig. 1. The architecture of Word2Vec method [7], [8].

II. Background

Traditional query expansion methods fail to take the contextual associations of words into consideration, which means that a query words semantic and syntactic similarity is typically ignored in document ranking systems. By introducing word embedding, our method selects additional query words from a list of word associations generated in a vector space. This section explains how our word embedding method generates word representations from large-scale unstructured text data.

The basic assumption behind word embedding is the *distribution hypothesis*—words with similar context tend to have similar meanings [9]. There has been a recent surge of work focusing on neural network algorithms learning word embedding, including a series of works that apply deep learning techniques to learn high-quality word representations [10], [11], [12]. We chose the Word2Vec method for our word embedding process since it has been shown to be effective and efficient for learning high-quality word embeddings in large-scale unstructured text data.

The word embedding process generates a vector space and projects every word to a point in that space. Similar words to a given word are selected by computing its cosine distance from the original word in the vector space. The generated word embeddings can also be interpreted as relations. For example, if we select the word vectors for *king*, *queen*, *man*, and *woman*, then we obtain the following relation:

$$v(king) - v(queen) \approx v(man) - v(woman). \quad (1)$$

The Word2Vec method aims to assign a word with intensive representations based on its context. To accomplish this goal, a sliding window is used over the input text stream. The central word is the target word, and the other words are the contextual words. Figure 1 shows Word2Vecs architecture, which contains two models: a continuous bag-of-word model (CBOW) and a Skip-gram model.

The CBOW model attempts to predict the target word using the contextual words in the sliding window. Formally, given a word sequence $\mathcal{D} = \{w_{i-k}, ..., w_{i-1}, w_i, w_{i+1}, ..., w_{i+k}\}$, where w_i is the target word, the objective of CBOW is to maximize the average log probability

$$L(\mathcal{D}) = \frac{1}{T}\sum_{i=1}^{T} \log Pr(w_i \mid w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k}). \quad (2)$$

where, T is the corpus size, and k is the context size of the target word, which indicates that the window size is $2k+1$. CBOW formulates the probability $Pr(w_i \mid w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k})$ with a softmax function as

$$Pr(w_i \mid w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k}) = \frac{\exp(\mathbf{x}_i \cdot \mathbf{x}_c)}{\sum_{w \in \mathcal{W}} \exp(\mathbf{x} \cdot \mathbf{x}_c)}, \quad (3)$$

where $\mathcal{W}$ represents the vocabulary, $\mathbf{x}_i$ is the vector representation of the target word w_i, and $\mathbf{x}_c$ is the average vector of all the contextual words.

Different from CBOW, Skip-gram model aims to predicts context words given the target word. Therefore, the objective of Skip-gram is to maximize the average log probability

$$L(\mathcal{D}) = \frac{1}{T}\sum_{i=1}^{T} \sum_{-k \le c \le k, c \neq 0} \log Pr(w_{i+c} \mid w_i), \quad (4)$$

where, k is the context size of the target word, and the probability $Pr(w_{i+c} \mid w_i)$ is formulated with softmax function, which is denoted as

$$Pr(w_{i+c} \mid w_i) = \frac{\exp(\mathbf{x}_{i+c} \cdot \mathbf{x}_i)}{\sum_{w \in \mathcal{W}} \exp(\mathbf{x} \cdot \mathbf{x}_i)}, \quad (5)$$

where $\mathcal{W}$ represents the vocabulary, $\mathbf{x}_i$ is the vector representation of the target word w_i, and $\mathbf{x}_{i+c}$ is the vector of context word.

Word2Vec has proven to be useful for many applications. We introduce Word2Vec to capture a words contextual associations to enhance the accuracy of query expansion.

III. A new document ranking method

This section describes the proposed query expansion method for document ranking in detail. It is based on word embedding and fuzzy rules and can be formally described by three algorithms:

- *The Qury Expansion Algorithm*: This algorithm expands document queries using contextual associations between words generated from word embedding.
- *The Expansion Queries Reweighting Algorithm*: After query expansion, fuzzy rules are designed to assign a new weights to every expansion word based on both original query weight and its similarity to the original query.
- *The Document Ranking Algorithm*: The document ranking algorithm computes a relevance score for each retrieval document that considers all words and their weights, and then returns relevant documents to the user.

A. Query Expansion based on Word Embedding

Document queries are formulated using a BM25 term weighting method. Each words BM25 score is computed first according to a set of documents the user is interested in.

BM25 is one of the state-of-the-arts term-based document ranking approaches. The term weights are estimated by

$$W(t) = \frac{tf \cdot (k+1)}{k \cdot ((1-b) + b \cdot \frac{dl}{avgdl})} \cdot \log(\frac{N - n + 0.5}{n + 0.5}), \quad (6)$$

where N is the number of documents in the collection; n is the number of documents that contain term t; tf represents the term frequency; dl is the document length; $avdl$ is the average document length; and k and b are parameters set to 1.2 and 0.75, respectively. The top 10 terms in BM25 are selected as the document queries that represent the user's interest, denoted as Q.

The word vector space is generated based on the Word2Vec method. For each query q in collection Q, an expanded collection $Q_q^+ = \{q_1^+, ..., q_k^+\}$ is contructed by selecting the top-k most similar words with cosine similarity in word vector space. Each term q_i^+ is associated with a weight according to its cosine distance to the query q.

We then rank the additional terms in Q^+ by their cosine similarity to the original query. Query q's sorted expansion words collection is represented as

$$T_{sorted}^{(q)} = \{(t_1, sim_1), ..., (t_i, sim_i), ..., (t_k, sim_k)\},$$

where t_i is the expansion word. i is the relevance ranking (i.e. t_1 is the most similar word), and sim_i is its similarity to query q:

$$sim_i = \cos(v_{t_i}, v_q). \quad (7)$$

B. Reweighting the Expansion Words Using Fuzzy Rules

Fuzzy rules are designed on the collection $T_{sorted}^{(q)}$. As mentioned above, additional queries are only weighted according to their cosine distance to the original query. However, in information retrieval tasks, a corpus may contain millions of words, and the gap between the top k words is likely to be small. To address this issue, fuzzy rules are used to reweight the expansion queries.

The fuzzy rules are based on two variables Δ and avg. They are denoted as:

$$avg = \begin{cases} S & \text{if } \frac{\sum_{i=1}^{k} sim_i}{k} < \alpha \\ L & \text{if } \frac{\sum_{i=1}^{k} sim_i}{k} \geq \alpha \end{cases} \quad (8)$$

$$\Delta = \begin{cases} S & \text{if } (sim_i - sim_k) < \beta \\ L & \text{if } (sim_i - sim_k) \geq \beta \end{cases} \quad (9)$$

where α and β are threshold parameters. The reweighting function is defined as:

$$f(i,p) = (\lceil (k/2) \rceil - i) * p * |sim_i - sim_{avg}| + sim_{avg}, \quad (10)$$

where k is the size of additional terms, i is the ranking of the term, sim_{avg} is the average of term's similarity, p is a parameter assigned with one of three degree−LOW, MEDIUM, or HIGH. The following fuzzy rules are used to set the reweighting function:

Rule 1 IF avg = S and Δ = S, THEN p = p_{MEDIUM}.
Rule 2 IF avg = S and Δ = L, THEN p = p_{LOW}.
Rule 3 IF avg = L and Δ = S, THEN p = p_{HIGH}.
Rule 4 IF avg = L and Δ = L, THEN p = p_{LOW}. .

The reweighting function follows two constraints: (1) after reweighting, simavg is equal to the middle terms weight in the sorted list; and (2) the weight is in the range of (0,1).

In summary, the additional term reweighting using a fuzzy rules method can be calculated with the following algorithm:

Algorithm 1 Reweighting the Expansion Words.

Input: query q original weight w_q;
additional query terms collection $T_{sorted}^{(q)}$;
Output: $W_q = \{w_1, w_2, ...w_i, ..., w_k\}$

```
sim_avg = avg(sim_1, sim_2, ..., sim_k)
Δ = sim_1 − sim_k
generate p in reweight function f using 4 fuzzy rules
for each term t_i ∈ T_q do
    w_i = f(i, p) * w_q
    if w_i not in (0,1) then
        constrain the weight to range (0,1)
    end if
end for
return W_q
```

C. Document Ranking Algorithm

Given a retrieval document d, we propose to estimate the relevance of d to the user's interest based on the expanded document queries, denoted as Q^+. A general formulation of the document ranking method is

$$w(q, Q_q^+) = (1-\gamma) * w_q + \gamma * score_{Q_q^+} \quad (11)$$

where γ is the combination coefficient; q is the original query, w_q is the weight of origin query q; $Q_q^+$is a set of additional terms from query q, and $score_{Q_q^+}$ is the weight of expansion words, which is calculated by

$$score_{Q_q^+} = \sum_{q_i^+ \in Q_q^+} w(q_i^+), \quad (12)$$

where $w(q_i^+)$ is generated in Algorithm.1.

Inheriting the above method, the relevance of document d is defined as:

$$Rank(d) = \sum_{q \in Q} ((1-\gamma) * w_q + \gamma * \sum_{q_i^+ \in Q_q^+} w_{q_i^+}), \quad (13)$$

where γ is the combination coefficient, w_q is the weight of origin query term, and $w_{q_i^+}$ is the reweight of expansion words.

IV. Experiments

In the following section, we present a set of experiments to evaluate the performance of our method in document ranking tasks. The results show that our document ranking method significantly outperforms the state-of-the-arts methods.

TABLE I
STATISTIC OF RCV1 DATABASE

# Documents	Corpus	Vocabulary Size	# Sentences
806,791	70.1M	111,257	20,300

A. Dataset

To evaluate and compare the performance of our proposed method with existing baseline methods, we conducted our experiments using the Reuters Corpus Volume 1 (RCV1) dataset, which is widely used in document ranking tasks. There are a total of 806,791 documents in the RCV1 dataset covering a variety of topics with a large amount of information. The documents are divided into 100 collections, and each collection is divided into a testing set and a training set. The first 50 collections were composed by human assessors, and another 50 collections were artificially constructed from intersection collections. Only the first 50 collections were used for our experiments. Each document contains $'title'$ and $'text'$, and these parts were used by all methods. We tokenized all text in the dataset with the help of the Stanford tokenizer tool and converted every word to lower case.

To train the word embedding model using Word2Vec toolkit, we combined all the documents in the RCV1 dataset as a training corpus totaling 16 million words. The size of word vector was set to 300. Statistical information about the database is provided in Table I.

B. Measures

To evaluate performance, we use six standard evaluation metrics: average precision of the top 10 documents ($P@10$), average precision of the top 20 documents ($P@20$), the $F1$ measure, mean average precision (MAP), the break-even point (b/p) and interpolated average precision (IAP) on 11-points.

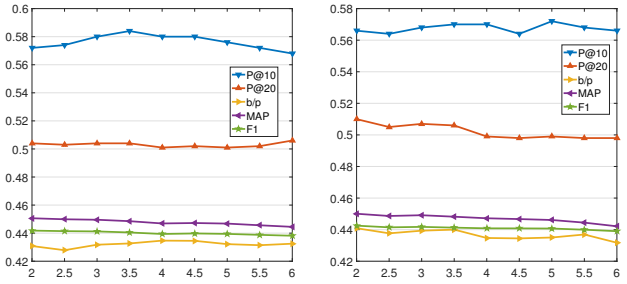
Precision was calculated as the proportion of labeled documents that were correctly identified. Recall was calculated as the proportion of labeled documents in the results that were correctly identified. The F1 measure is a criterion that assesses the effect of both precision (p) and recall (r), which is defined as $F1 = \frac{2pr}{p+r}$. The 11-points measure is the precision at 11 standard recall levels (i.e., recall = 0, 0.1,...,1). The larger the $P@10$, $P@20$, MAP, b/p, $F1$ score, the better the system performs.

C. Baseline Models

We chose BM25, WordNet, CBOW, and Skip-gram as baseline methods.

BM25 is a state-of-the-art term-based document ranking method. Our method uses BM25s approach to rank the weight of all words and select the top 10 as the original queries to capture users interest. We also regard BM25 method as one of the term-based document ranking baseline methods.

WordNet is a large lexical database of English words. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (denoted as synsets in WordNet), each expressing a distinct concept. It also provides short, general definitions and records the various semantic relationships between synsets. Synsets are interlinked by means of conceptual-semantic and lexical relations, and its structure makes it a useful tool for computational linguistics and natural language processing. We selected WordNet as a query expansion baseline method because it superficially resembles a thesaurus, in that it groups words together based on their meanings. For each query, we looked up its synsets for all possible parts of speech and selected no more than 5 words for the expansion set.



(a) CBOW (b) Skip-gram

Fig. 2. The effect of parameter p.

The Word2Vec method contains two models: the **CBOW** model and the **Skip-gram** model, and therefore we also selected these as baseline methods for query expansion. The Word2Vec method was selected to verify the effectiveness of fuzzy rules in reweighting expansion queries.

D. Settings

In the document ranking task, we generated document queries for each collection by selecting the top-10 terms based on their BM25 weight. The queries were expanded to a range of 5 using both the CBOW method and the Skip-gram method. The parameters of Word2Vec were set as follows: word vector dimensionality 300; negative samples 25; and window size 5 words. The reweighting function parameters were set to $\alpha = 0.5$, $\beta = 0.08$, $\gamma = 0.5$. The parameter $p(p_{LOW}, p_{MIDDLE}, p_{High})$is an essential parameter and it was set $(p - 0.5, p, p + 0.5)$. We tuned this parameter for different methods in terms of $P@10$, $P@20$, b/p, MAP and $F1$. As Figure 2 shows, the results of $p = 3.5$ gave the best reweighting performance based on CBOW word similarity and Skip-gram's word similarity.

Different words have a different number of similar words in the WordNet synsets. We chose no more than 5 words from each synsets for each query. WordNet is not able to provide the similarity between words, so all expansion queries were assigned with the same average weight.

E. Overall Performance

Tabel II shows the performance comparison between our method and the baseline methods for the document ranking tasks. From the table, we observe that: All query expansion methods significantly outperformed the traditional

TABLE II
OVERALL PERFORMANCE

Methods	P@10	P@20	b/p	MAP	F1
BM25	0.446	0.441	0.406	0.408	0.415
QE-WordNet	0.526	0.495	0.421	0.436	0.432
QE-CBOW	0.550	0.499	0.423	0.437	0.432
QE-Skipgram	0.538	0.498	0.424	0.436	0.430
QE-CBOW-Fuzzy rules	**0.584**	0.504	0.433	**0.449**	0.440
QE-Skipgram-Fuzzy rules	0.570	**0.506**	**0.440**	0.448	**0.441**

BM25 method. This demonstrates the effectiveness of query expansion in overcoming the lexical gap and shows improved performance for document ranking. Query expansion based on word embedding methods (QECBOW and QE-Skip-gram) outperformed the external lexical methods (i.e., query expansion based on WordNet, QE-WordNet). This demonstrates that word association information generated from semantic vector space is more effective for selecting additional query terms. Our query expansion methods (QE-CBOW-Fuzzy rules and QE-Skipgram-Fuzzy rules) achieved better results than the word embedding models without reweighting terms using fuzzy rules. This is mainly because the corpus contained millions of words and the difference in similarity between the top k words was small. As previously mentioned, our method introduces fuzzy rules to reweight the expansion words which helps to amplify these differences. The 11-point results of all methods are shown in Figure 3. The results indicate that our methods have achieved the best performance compared with all the other baseline methods.

- All query expansion methods significantly outperformed the traditional BM25 method. This demonstrates the effectiveness of query expansion in overcoming the lexical gap and shows improved performance for document ranking.
- Query expansion based on word embedding methods (*QE-CBOW* and *QE-Skip-gram*) outperformed the external lexical methods (i.e., query expansion based on WordNet, *QE-WordNet*). This demonstrates that word association information generated from semantic vector space is more effective for selecting additional query terms.
- Our query expansion methods (*QE-CBOW-Fuzzy rules* and *QE-Skipgram-Fuzzy rules*) achieved better results than the word embedding models without reweighting terms using fuzzy rules. This is mainly because the corpus contained millions of words and the difference in similarity between the top k words was small.As previously mentioned, our method introduces fuzzy rules to reweight the expansion words which helps to amplify these differences.

The 11-point results of all methods are shown in Figure 3. The results indicate that our methods have achieved the best performance compared with all the other baseline methods.

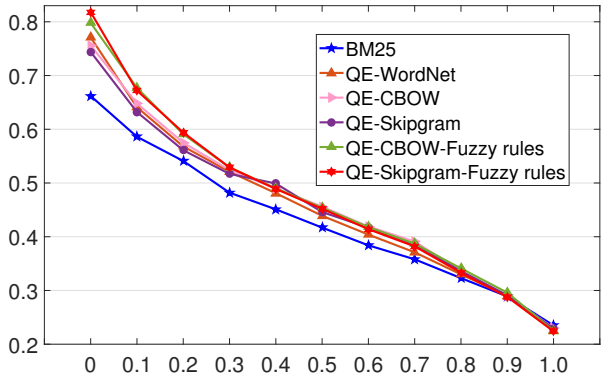


Fig. 3. The performance of IAP on 11-points

F. Case Study

A case study was conducted to further analyze why our method surpassed the other candidate selection methods. Several examples are listed in Table III. Given a query, we present the expansion words and their weights using different methods.

According to observations from the dataset, two facts mainly account for the failure of the WordNet method. The first is that the size of original querys synonyms was less than 5. As a result, it is difficult to extend the semantics of query words for document ranking tasks. Another problem is that WordNet only selects words with same semantic meaning as valuable associated words. As we can see, WordNet selected *federal bureau of investigation* because it has exactly the same meaning as *FBI*; however, other associated words were ignored, such as *CIA* and *spy* which were recognized in the word embedding methods. Therefore, query expansion methods based on word embedding can achieve much better performance than methods based on WordNet.

Fuzzy rules play an important role in reweighting expansion words. For example, in the QE-CBOW method, the weight gap between the maximum (word *dad*) and the minimum (word *thriller*) for the query *kid* was only 0.036. While in our method, introducing fuzzy rules enlarged the difference in weights based on their cosine similarity. Assigning proper weights to different terms further enhances document ranking performance. In summary, our method achieved the best performance because the expansion words were chosen carefully and then reweighted.

V. RELATED WORK

In document retrieval tasks, representing words using fixed-length vectors is an essential step for processing text. The one-hot representation method is traditionally favored for its simplicity and efficiency; however, this method does not consider semantic information. As a result, it suffers from data sparsity, the curse of dimensions, and the lexical gap problem, which makes information retrieval tasks difficult. Distributed word representation, also known as word embedding, has been introduced to solve these problems. In this method, words are represented as dense, low-dimensional, real-valued

TABLE III
CASE STUDY

kid	Expansion words and weights									
QE-WordNet	child	0.333	kyd	0.333	pull_the_leg_of	0.33				
QE-CBOW	dad	0.293	someone	0.290	yankees	0.269	woman	0.261	thriller	0.257
QE-Skipgram	supper	0.422	someone	0.413	arnelle	0.394	batman	0.374	ego	0.373
QE-CBOW-Fuzzy rules	dad	0.404	someone	0.328	yankees	0.274	woman	0.229	thriller	0.157
QE-Skipgram-Fuzzy rules	supper	0.581	someone	0.458	arnelle	0.395	batman	0.322	ego	0.242
FBI	**Expansion words and weights**									
QE-WordNet	federal_bureau_of_investigation	1.0								
QE-CBOW	cia	0.427	oss	0.378	ntsb	0.378	investigators	0.371	spy	0.345
QE-Skipgram	freeh	0.626	cia	0.578	pitts	0.569	undercover	0.516	mislock	0.515
QE-CBOW-Fuzzy rules	cia	0.661	oss	0.375	ntsb	0.380	investigators	0.353	spy	0.173
QE-Skipgram-Fuzzy rules	freeh	0.953	cia	0.613	pitts	0.561	undercover	0.426	mislock	0.285

vectors. Each dimension represents the latent semantic and syntactic features of words. More recently, there has been a surge of work focusing on neural network (NN) algorithms for learning word representations (Bengio et al.[10]; Collobert and Weston[11]; Mnih and Hinton; Mikolov et al.[8].).

Several studies have focused on query expansion using word embedding. Kuzi et al.[13] proposed a query expansion method, based on CBOW, which uses the terms to either expand the original query or for integration with an effective pseudo-feedback relevance model. Fernando Diaz et al.[14] studied the use of term-relatedness information, generated by a word embedding method, in the context of query expansion for ad hoc information retrieval.

Recently, fuzzy theory has been applied in many data analysis applications, such as recommendation systems (Zhang et al.[15]), pattern recognition (Chu et al.[16]). Several works have also used fuzzy rules to enhance the performance of query expansion. Bhatnagar et al.[17] proposed a query expansion method by hybridizing corpus information with a genetic-fuzzy approach and a semantic similarity notion. Hsi-Ching Lin et al.[18] used fuzzy rules to infer the weights of the additionally generated terms based on user relevance feedback techniques.

VI. CONCLUSION AND FURTHER STUDY

In this paper, we presented a new method for enhancing word embedding similarity measures for query expansion that selects expansion words according to their semantic similarity from a candidate list generated by word embedding. To enhance the accuracy of the document ranking, fuzzy rules are used to reweight the expansion words. The original queries and the expansion words, together with their weights, are then used to rank documents to improve the performance of information retrieval tasks. The method was evaluated in a series of document ranking tasks using the RCV1 dataset and demonstrates excellent strength in both query expansion and document ranking compared to the state-of-the-art baselines.

In future work, we hope to further improve our method by incorporating association patterns to better gauge user interests and explore reweighting the expansion words given interactive user information.

ACKNOWLEDGMENT

The work was supported by National Hi-Tech Research & Development Program (863 Program, Grant No. 2015AA015404), and Australian Research Council (ARC) under discovery grant DP170101632.

REFERENCES

[1] Stephen E. Robertson, Hugo Zaragoza, Michael J. Taylor. Simple BM25 extension to multiple weighted fields. CIKM., 2004, pp.42-49.
[2] Florian Beil, Martin Ester, Xiaowei Xu. Frequent term-based text clustering. KDD., 2002, pp.436-442.
[3] George A. Miller. WordNet: A Lexical Database for English. Communications of the ACM., Vol. 38, No.11, pp.39-41, 1995.
[4] Juri Ganitkevitch, Benjamin Van Durme, Chris Callison-Burch. PPDB: The Paraphrase Database. HLT-NAACL., 2013, pp.758-764.
[5] Guangyou Zhou, Tingting He, Jun Zhao, Po Hu. Learning Continuous Word Embedding with Metadata for Question Retrieval in Community Question Answering. ACL., 2015, pp.250-259.
[6] Meng Zhang, Yang Liu, Huan-Bo Luan, Maosong Sun, Tatsuya Izuha, Jie Hao. Building Earth Mover's Distance on Bilingual Word Embeddings for Machine Translation. AAAI.,2016, pp.2870-2876.
[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. NIPS., 2013, pp.3111-3119.
[9] Firth, John Rupert. A synopsis of linguistic theory 1930-1955. Studies in Linguistic Analysis, pp.132.
[10] Yoshua Bengio, Rjean Ducharme, Pascal Vincent, Christian Janvin. A Neural Probabilistic Language Model. JMLR., vol.3: pp.1137-1155, 2003.
[11] Ronan Collobert, Jason Weston, Lon Bottou, Michael Karlen, Koray Kavukcuoglu, Pavel P. Kuksa. Natural Language Processing (Almost) from Scratch. JMLR., vol.12: pp.2493-2537, 2011.
[12] Jeffrey Pennington, Richard Socher, Christopher D. Manning. Glove: Global Vectors for Word Representation. EMNLP., 2014, pp.1532-1543.
[13] Saar Kuzi, Anna Shtok, Oren Kurland. Query Expansion Using Word Embeddings. CIKM., 2016, pp.1929-1932.
[14] Fernando Diaz, Bhaskar Mitra, Nick Craswell. Query Expansion with Locally-Trained Word Embeddings. ACL., 2016, pp.367-377.
[15] Qian Zhang, Dianshuang Wu, Guangquan Zhang, Jie Lu. Fuzzy user-interest drift detection based recommender systems. FUZZ-IEEE., 2016, pp.1274-1281.
[16] Chun-Hsiao Chu, Kuo-Chen Hung, Peterson Julian. A complete pattern recognition approach under Atanassov's intuitionistic fuzzy sets. Knowl.-Based Syst. vol.66, pp.36-45, 2014.
[17] Pragati Bhatnagar, Narendra Pareek. Improving pseudo relevance feedback based query expansion using genetic fuzzy approach and semantic similarity notion. J. Information Science., vol.40(4), pp.523-537, 2014.
[18] Hsi-Ching Lin, Li-Hui Wang, Shyi-Ming Chen. Query expansion for document retrieval based on fuzzy rules and user relevance feedback techniques. Expert Syst. Appl., vol.31(2), pp.397-405, 2006.