Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Semantic composition of distributed representations for query subtopic mining*

Wei SONG, Ying LIU, Li-zhen LIU†‡, Han-shi WANG

Information and Engineering College, Capital Normal University, Beijing 100048, China

†E-mail: liz_liu7480@cnu.edu.cn

Received Aug. 15, 2016; Revision accepted July 12, 2017; Crosschecked Nov. 12, 2018

Abstract: Inferring query intent is significant in information retrieval tasks. Query subtopic mining aims to find possible subtopics for a given query to represent potential intents. Subtopic mining is challenging due to the nature of short queries. Learning distributed representations or sequences of words has been developed recently and quickly, making great impacts on many fields. It is still not clear whether distributed representations are effective in alleviating the challenges of query subtopic mining. In this paper, we exploit and compare the main semantic composition of distributed representations for query subtopic mining. Specifically, we focus on two types of distributed representations: paragraph vector which represents word sequences with an arbitrary length directly, and word vector composition. We thoroughly investigate the impacts of semantic composition strategies and the types of data for learning distributed representations. Experiments were conducted on a public dataset offered by the National Institute of Informatics Testbeds and Community for Information Access Research. The empirical results show that distributed semantic representations can achieve outstanding performance for query subtopic mining, compared with traditional semantic representations. More insights are reported as well.

Key words: Subtopic mining; Query intent; Distributed representation; Semantic composition
https://doi.org/10.1631/FITEE.1601476 **CLC number:** TP391.3

1 Introduction

Inferring query intent is the key task for information retrieval. It is common that users face difficulties in felicitously expressing their real intention using several key words, so that web queries are usually ambiguous and multifaceted. Therefore, search results are often unable to satisfy users' real information needs as specifically as they expect. To solve this problem, a query subtopic mining task is proposed, which aims to find possible subtopics for a given query and return a ranked list of them in terms of the relevance to the query, popularity, and diversity of subtopics (Kim and Lee, 2013).

A lot of work has been done to represent query subtopics by providing a flat or hierarchical structure using different information resources. The commonly used pipeline includes the following procedures: (1) extract subtopic candidates from various resources, including query suggestions (Baeza-Yates et al., 2005; Santos et al., 2010), top-ranked documents (Xu and Croft, 1996; Beeferman and Berger, 2000; Zeng et al., 2004), query logs or online encyclopedias (Hu et al., 2009; Jiang et al., 2011), anchor texts, and uniform resource locators (URLs); (2) organize the subtopic candidates into clusters (clustering), and view each cluster as a subtopic; (3) rank the mined subtopics and select a label to represent each subtopic. During this process, the representation of subtopic candidates is

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 61876113 and 61402304), the Beijing Educational Committee Science and Technology Development Plan of China (No. KM201610028015), and the Beijing Advanced Innovation Center for Imaging Technology of China

ORCID: Ying LIU, http://orcid.org/0000-0002-9125-4326

the key point since it directly affects the quality of clustering subtopic candidates. Many existing subtopic mining methods represent subtopic candidates based on a bag of words and co-occurrence statistics in search results. Such methods face word mismatch problems and require a way to close the semantic gap.

Recently, deep learning techniques have become very popular. In such frameworks, words and pieces of texts are represented as distributed continuous vectors, which can project words and texts into a semantic space. The vector representation of words can be learned using neural networks on large-scale datasets (Bengio et al., 2003; Mikolov et al., 2013b). Such semantic representation can capture important syntactic and semantic regularities (Mikolov et al., 2013b); this makes it easy to measure semantic relatedness and alleviates the term mismatch problem. Motivated by the success of applying distributed representations to several natural language processing tasks, we wonder whether such representations work effectively for query subtopic mining. To the best of our knowledge, there is little work that systematically studied the effectiveness of using distributed representations for this task.

We exploit distributed representation based semantic composition for query subtopic mining. In particular, we focus on two semantic composition strategies for representing multiword subtopic candidates. The first one is the paragraph vector (PV), which jointly learns the vector representations of words and the arbitrary length of a text (Le and Mikolov, 2014).

The second one is “word vector composition”. This approach first learns the distributed representations of words and then uses certain semantic composition methods to obtain the distributed representation of each subtopic candidate.

We learn distributed representations on two types of data: query log data and document corpus. We are interested to see whether the data type affects the quality of learned distributed vectors. To compare the performance of different strategies, we conduct experiments on the National Institute of Informatics Testbeds and Community (NTCIR) INTENT Chinese subtopic mining task collections. Our findings include:

1. Distributed representations based query subtopic mining approaches outperform baselines based on traditional semantic representations.

2. The word vector composition approach achieves a better performance compared with the paragraph vector approach.

3. The semantic composition method based on multiplication and co-occurrence works best.

4. In most cases, distributed vectors learned from the query log data perform better; however, the performance is not consistent and is dependent on the semantic composition methods.

2 Related work

In this section, we introduce the related work on query subtopic mining and two main semantic representation methods. The first is the semantic composition method for representing phrases given word embeddings. The other is distributed representations of sentences and documents, i.e., the PV approach.

2.1 Query subtopic mining

Web queries are often ambiguous and multifaceted, and users cannot obtain specific search results as expected. Recognizing the intent behind a query is an important challenge, and various problems have been tackled related to this challenge. Subtopic mining has been proposed under these circumstances. The topic is more coarse-grained and related to other queries, and in contrast, the subtopic is fine-grained and is only about the query in question (Clarke et al., 2009). This is the major difference between topic and subtopic. Query suggestion and search result diversification, which aim to satisfy diverse user intent with a single search result page, are both applications of query subtopic mining (Rafiei et al., 2010; Zheng and Fang, 2011). Query subtopics can be used to support query-oriented summarization as well (Song et al., 2012).

Existing studies relevant to the problem of subtopic mining can be divided into different categories, including the query log mining method, the search result mining method (Jones and Klinkner, 2008; Li et al., 2008; Strohmaier et al., 2009; Hu et al., 2012), the method based on integrated data from multiple resources (Damien et al., 2013; Wang et al., 2013; Zheng et al., 2014), and many other methods. Strohmaier et al. (2009) first obtained similar queries from search sessions, filtered out noisy queries using click-through data, and then grouped the remaining queries based on random walk similarity. Radlinski et al. (2010) mined query suggestions using both session logs and web pages. Luo et al. (2014) analyzed query log data to establish bipartite graphs and then generated large-scale candidate sets by a random-walk method. Zeng et al. (2004) first extracted salient phrases from search result snippets and then grouped the search results by ranking the salient phrases. Hu et al. (2012) proposed a clustering algorithm which can effectively leverage “one subtopic per search” and “subtopic clarification by keyword” to automatically mine the major subtopics of queries, where each subtopic is represented by a cluster containing a number of URLs and keywords. Wang et al. (2013) proposed a diversified retrieval model to rank documents with respect to the mined subtopics for balancing relevance and diversity. External resources were used to recognize the subtopics for a given query in the category-based method proposed by Wang et al. (2013).

Query subtopic mining is also helpful for many other tasks, such as query suggestion. Anagnostopoulos et al. (2015) introduced an algorithmic approach capable of creating a two-fold dynamic semantic query suggestion set based on the most common Twitter entities (TEs), i.e., hashtags (#) and mentions (@), as well as the links that appear in tweets. Karunasekera et al. (2014) tracked microblog discussions on a given topic. They proposed a new term-scoring expression to score the returned microblogs to identify whether they are on topic or not.

Since 2011, NTCIR has organized the IMine task, comprising the subtopic-mining and document-ranking sub-tasks (Song et al., 2011; Sakai et al., 2013; Joho and Kishida, 2014; Liu et al., 2014). NTCIR provides topics sampled from the median-frequency queries collected from both Sogou and Bing search logs. Song et al. (2009) proposed a query intent classification taxonomy that groups queries into three types—ambiguous, broad, and clear. The number of queries within each of the three types of topics provided by NTCIR is almost equal. NTCIR’s collection is publicly available and is becoming one of the standard evaluation datasets for subtopic mining. In this study, we conduct experiments on the NTCIR INTENT Chinese subtopic mining task collections.

2.2 Distributed semantic representations

Word embedding has attracted considerable attention from many researchers. In recent years, several models, which have achieved state-of-the-art success in many natural language processing tasks, have been proposed. Bengio et al. (2003) first proposed a neural network language model (NNLM) that can learn a word embedding and a language model synchronously. Similar to this model, Mnih and Hinton (2007) proposed the log-bilinear language model, replacing the nonlinear activation function tanh of NNLM with a log-bilinear energy function. A model that trains only word embedding was proposed by Collobert and Weston (2008). All these three models have a hidden layer. Word2Vec has already become one of the most popular word embedding generation tools in the space of only a year and is a useful tool to a great extent. It has two models, the continuous bag-of-words (CBOW) and skip-gram models (Mikolov et al., 2013a). CBOW (Mikolov et al., 2013a) thoroughly removes the hidden layer of the neural network. Skip-gram (Mikolov et al., 2013a) uses the simplest strategy, choosing one word in the window of the target words as the representative of the context. They attempt to minimize computational complexity and increase computing speed. Mikolov et al. (2013a) proved that the CBOW model can perform better with a larger training corpora, even though it simplifies the neural network. Baroni et al. (2014) also claimed that the CBOW model achieves a significantly better performance than the count models (models based on a word-context co-occurrence matrix) do in almost all semantic tasks. The training corpora in our experiment are large enough. Thus, to achieve a higher efficiency, we use the Word2Vec toolkit as our word embedding generation tool.

Compositional semantic models aim to build a distributional representation of a phrase from its component word representation. More and more new compositional semantic models are being put forward. Most existing compositional distributional semantic models can be divided into traditional additive and multiplicative models. The traditional approach is to combine single word representation with compositional operators either pre-defined (Mitchell and Lapata, 2010) or learnd from data (Le and Mikolov, 2014). Some researchers expanded traditional approaches to consider the inner structure of phrases and their context (Yu and Dredze, 2015), applying a matrix transformation (Zanzotto et al., 2010; Grefenstette et al., 2013; Zhao et al., 2015) or recursive neural networks (Socher et al., 2011b). As our experiment applies the semantic representation of a phrase to mine subtopics, we employ traditional semantic composition methods to represent a phrase.

Socher et al. (2011a) used a matrix–vector operation to combine the word vectors in an order given by the parse tree of a sentence; however, it works for only those sentences based upon a parse tree. Le and Mikolov (2014) proposed an unsupervised framework named "PV", which can learn continuous distributed vector representations for pieces of text. The text can have variable lengths ranging from a phrase to a sentence to a whole document. This algorithm can learn the fixed-length feature vectors of texts with different lengths and represent each piece of text by a dense vector. The vector represents the information that is missing from the current context. PV is a way to take into consideration the variable lengths of queries and subtopics in our subtopic mining processing.

3 Subtopic mining framework

3.1 Overview

The common framework of subtopic mining can be divided into three parts. First, the subtopic candidate extraction module extracts subtopic candidates for a given query from available resources such as a query log. Next, the subtopic candidates should be represented by a certain semantic representation. Finally, the subtopic candidates are clustered into groups according to the semantic similarities between their representations, and each cluster group forms a query subtopic.

In this study, we follow this pipeline. Without loss of generality, given a query q, we extract contexts $\{a\}$ of this query from some resources, and call a phrase $s=q+a$ a "subtopic candidate". We would group these subtopic candidates into clusters to form the final query subtopics. Since a candidate usually consists of multiple words, we have to work out a way to represent it properly.

Since we focus on comparing the effectiveness of various distributed semantic representations for subtopic mining, the main difference among different approaches is the semantic representations used. Next, we introduce the three parts one by one.

3.2 Subtopic candidate extraction

User search behavior data is available in the form of a query log, as a result of user interactions with search engines. A query log records user interaction information such as query string, submission time, and user clicks, and each of these records indicates a particular user's information need at that time. Considering that millions of users use search engines every day, query logs contain a wealth of information about the users and the world. Analyzing a query log is a promising way to understand user interactions with search engines.

We extract subtopic candidates from a query log. For a given query q, we extract all queries that contain q as a substring as the subtopic candidates. For example, if q equals "Microsoft", query "Microsoft products" would be extracted as a subtopic candidate. This is reasonable. Since a query might be ambiguous, users tend to use extra words to specify their search goals so that such subtopic candidates do reflect the user intent. In implementation, we constrain that a subtopic candidate must appear in a query log at least three times as a whole query, and discard those whose lengths are too large to reduce the noise.

3.3 Subtopic candidate representation

Before clustering subtopic candidates into groups, we must project these subtopic candidates into a certain kind of representation. We compare two categories of representations: the BOW based representations and the distributed representations. The BOW based representations represent each subtopic candidate as a point in the word space. In contrast, distributed representations represent a string with a continuous distributed vector, so that they can be used to measure the semantic relatedness between texts that vary in lexical surface strings, such as "movie" and "film", which can alleviate the term-mismatch problem.

The subtopic candidates are multi-word expressions. We consider two strategies for learning their representations. The first one is based on the PV method, which was proposed by Le and Mikolov

(2014) to directly learn distributed representations for word sequences with an arbitrary length.

The other option is to obtain the representations of multi-word expressions based on the embeddings of individual words through semantic composition. In this study, we compare several composition strategies and expect to find the most effective one. We name this strategy "word vector composition".

3.4 Subtopic candidate clustering

The extracted subtopic candidates cannot reflect search intent well, since there is a great semantic redundancy even when they have different words. It is necessary to group subtopic candidates into clusters according to semantics, and each cluster corresponds to a distinct intent. This way would largely reduce the cognitive burden of the users.

Actually, all clustering algorithms can be applied. The main idea of this study is to compare different semantic representations for subtopic mining. Therefore, we apply the *k*-means clustering algorithm for its simplicity and effectiveness.

The key to clustering is to precisely measure the similarity among samples. This depends on both the similarity function and semantic representations of samples. We fix the similarity function as the cosine similarity and focus mainly on evaluating the effectiveness of various representations. The cosine similarity between vectors is defined by

$$\mathrm{sem}(c_i, c_j) = \frac{\boldsymbol{v}_{c_i} \cdot \boldsymbol{v}_{c_j}}{|\boldsymbol{v}_{c_i}| \cdot |\boldsymbol{v}_{c_j}|}, \tag{1}$$

where c_i is the i^{th} subtopic candidate and $\boldsymbol{v}_{c_i}$ denotes the semantic vector of this subtopic candidate.

4 Distributed representations of subtopic candidates

In this section, we introduce the distributed representations used to represent query subtopic candidates. Our target is to map each query subtopic candidate to a distributed dense vector.

4.1 Paragraph vector

PV provides the idea of learning the vector representation of a sentence or a document directly (Le and Mikolov, 2014). In the PV framework, there are two variations for obtaining a document vector—the distributed memory version of PV (PV-DM) and the distributed BOW version of PV (PV-DBOW). According to the report proposed by Le and Mikolov (2014), PV-DM works better than PV-DBOW; thus, we used the PV-DM model in this study.

The basic idea of the PV model is illustrated in Fig. 1. For the PV model, each paragraph has a label, and the PV model learns a vector representation for each word and label in a dataset. Given the label and the contextual words, which are represented by vectors, a certain strategy (such as average or concatenation) was used to represent the label and these words in a single vector. This vector was then used to predict the next word in the context. According to its internal structure, we could consider it as a special semantic composition of word vectors. The optimal target of the model is to make the differences between predictions and the actual words as close as possible.

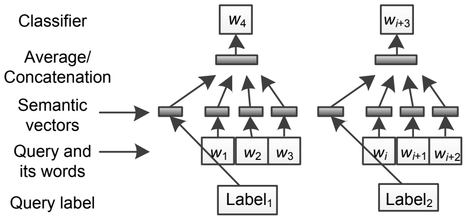

Fig. 1 Framework of the paragraph vector model

4.2 Word vector composition

4.2.1 Word vector

Word embedding is a unique distributed continuous vector, which maps a word into a high-dimensional semantic space. We used the toolkit Word2Vec (Mikolov et al., 2013a) to learn word embeddings. Word2Vec provides two models: the skip-gram and CBOW models. Its CBOW model has been proved more efficient with larger training corpora even though it simplifies the neural network. In this study, we used the CBOW model.

4.2.2 Semantic composition

We aggregated word representations to obtain the representation of a multi-word subtopic candidate by semantic composition. To achieve this, we employed the most widely used composition strategies: additive and multiplicative compositions.

1. Additive composition

In the additive function, the general form can be expressed as

$$\boldsymbol{V}_{\text{phrase}}(w_1, w_2, \cdots, w_n) = \sum_{i=0}^{n} \alpha_i \cdot \boldsymbol{V}(w_i), \tag{2}$$

where $\boldsymbol{V}(w_i)$ represents the vector of word w_i, $\boldsymbol{V}_{\text{phrase}}(w_1, w_2, ..., w_n)$ represents the vector of a phrase containing n words, and α_i is a weight parameter for the i^{th} word in the phrase subject to $\sum_{i=0}^{n} \alpha_i = 1$. In this way, a phrase is represented in a vector form, which has the same dimensionality as a word vector.

In general, words in a phrase are not equally important in indicating its meaning. Therefore, we aimed to assign larger weights to words with more importance for understanding the phrase to keep the composite vector close to its real position in the vector space. We used different metrics to measure the importance of a word in the phrase to test the effectiveness of different constructing methods on the subtopic mining results. They are as follows:

Average (AVE): The weight of each word in the phrase is equal; i.e., $\alpha_i=1/n$, where n is the number of words constructing the phrase. This weight setting is used as the baseline of other weight settings.

Co-occurrence (COO): Co-occurrences are used to measure the closeness between words. We assumed that the more times a word w co-occurs with query q, the more important it is. Weight α_i for word w_i is defined as

$$\alpha_i = \frac{\text{co}(w_i, q)}{\sum_{j=0}^{n} \text{co}(w_j, q)}, \tag{3}$$

where co(·, ·) is a function representing the co-occurrence times of two strings in a query log.

Term frequency–inverse document frequency (TF-IDF): TF-IDF is a statistical method used to evaluate the importance of a word for one of the documents in a set of files or a corpus. The importance of a word is directly proportional to the number of its occurrences in the file, but at the same time is inversely proportional to its frequency in the corpus. It is a common kind of weighted technology. Therefore, we can consider the importance of a word in the phrase using TF-IDF. Here, we expanded co-occurrence weight with TF-IDF, and weight α_i for word w_i is defined as

$$\alpha_i = \frac{\text{co}(w_i, q)}{\sum_{j=0}^{n} \text{co}(w_j, q)} \cdot \lg\left(\frac{\text{DF}(w_i)}{N} + 1\right), \tag{4}$$

where DF(w_i) is the number of documents in which word w_i appears, and N is the total number of all documents. We continued to use co-occurrence of word and query to compute TF instead of the frequency of the word alone in the whole corpus. This can ensure the affinity of the word and the original query.

2. Multiplicative composition (MUL)

The multiplicative function is another option. As a particular instance of the more general class of multiplicative functions, we define the method as follows:

$$\boldsymbol{V}_{\text{phrase}} = \boldsymbol{V}(w_1) \bullet \boldsymbol{V}(w_2) \bullet \cdots \bullet (w_n), \tag{5}$$

where "●" represents multiplication of the corresponding components:

$$\boldsymbol{V}_{\text{phrase}_i} = \boldsymbol{V}(w_1)_i \cdot \boldsymbol{V}(w_2)_i \cdot \cdots \cdot \boldsymbol{V}(w_n)_i. \tag{6}$$

5 Experimental setup

5.1 Datasets for learning distributed representations

The ability to train precise PVs and word embeddings depends greatly on the training corpus. The intuitive mainstream idea is that the more semantic information the corpus covers, the better the distributed vectors can be learned. Except for the size, different sources of training corpora can influence the quality of the learned representations. We used two types of data: a query log dataset and the encyclopedia dataset. The two datasets are both large enough. In our experiment, the size of the training dataset was not considered during comparison.

The query log dataset is the SogouQ query log corpus which was collected in August 2006, March 2007, June 2008, and November 2011 from http://www.sogou.com/. It is publicly available. Each record in the dataset includes six items: date/time of the user click, user identification (ID), user's query, URL clicked by the user, rank of the clicked URL in the result list, and the order of the URL click within the session. We used the queries to train PVs and word embeddings.

The encyclopedia dataset is a collection of web pages covering various topics from http://baike.baidu.com/, the largest Chinese knowledge base. It covers all areas of the network information, and contains more semantic information. Baidu Baike dataset is suitable for training semantic representation models of Chinese.

For learning word and query representations based on the PV approach, we treated each query in SogouQ as a single sentence and all query strings from SogouQ as the data. Because the lengths of the subtopic candidates and documents in Baidu Baike differ significantly, we did not use the Baidu Baike dataset for training PVs.

For learning the individual word embeddings, we used both the SogouQ dataset and encyclopedia dataset. Thus, we could compare how distributed vectors learned from different data types can affect the subtopic mining results.

The dimensions of the distributed vectors also have a significant impact on the experimental results. Through repeated experimentation using different values, we chose 50 as the dimensionality of PV, and 300 as that of the word embeddings.

5.2 Test datasets and evaluation metrics for subtopic mining

We conducted our experiment on the public Chinese data collection offered by NTCIR-9 (Song et al., 2011) intent task, including query suggestions from major search engines, query dimensions from search result pages, and related queries extracted from the SogouQ query log. The numbers of ambiguous queries, broad queries, and clear queries are approximately equal in this collection. To complete the candidates for each query in this data collection, the SogouQ dataset, as an additional resource, was used to extract related subtopic candidates and compute the weights of words in subtopic candidates. We compared the performance of different semantic representations with the results of the subtopic mining task with the following evaluation metrics: I-rec@n, D-nDCG@n, and D#-nDCG@n (Sakai and Song, 2011). I-rec, short for intent recall, is a measure of subtopic diversity. I-rec@n describes the coverage of query intent in the top documents of a sorted document list. D-nDCG is a measure of the overall relevance between the query intent and the original query across subtopic sets. D#-nDCG is a linear combination of previous metrics which consider both relevant performance and subtopic coverage. The document cutoff in our experiment was set to 10, which reflects the quality and reliability of the algorithm (Sakai and Song, 2011).

5.3 Baselines

The BOW model is the most common text representation method used for natural language progressing and information retrieval. In this model, a text (such as a sentence or a document) is represented by an unordered collection of words, disregarding grammar and even word order. Each term in collection is independent and has no dependencies with other terms. The feature of a text is a vector, whose dimensionality is the number of words in collection. Each dimension of the feature vector is the frequency of the corresponding term in the text.

Clustering reformulated queries (CRQs) (Dang et al., 2011) is a famous method for inferring query intent. It obtains reformulations for queries from an anchor text and Microsoft web N-gram services, and then clusters reformulations into groups, each of which is considered a coarse representation of a query aspect. It has achieved remarkable results.

We also compared the performance of the proposed methods with the best performance of the participants of NTCIR-9.

5.4 Analysis of experimental results

In this subsection, we describe and analyze the results of the experiment, and then show the results of different semantic compositions of distributed representations for subtopic mining. Each model was tested 10 times and the average score was taken as the final result. According to the report proposed by Song et al. (2016), most queries in a query log have no more than 10 topics and about 50% queries have at least 5 topics; thus, in this study, we set the number of expected subtopics to 5 and 10 and report the results respectively.

5.4.1 Results of distributed representation based methods

Table 1 shows the performance of the distributed representation based methods, including the PV method, three weighting schemes (AVE, COO, and TF-IDF) for additive composition, and MUL. The distributed representations were trained on the query log dataset.

Table 1 Performance of distributed representation based methods

Number of topics	Method	I-rec	D-nDCG	D#-nDCG
5	PV	0.5382	0.7236	0.6304
	AVE	0.5197	0.7141	0.6156
	COO	0.5582	0.7328	0.6301
	TF-IDF	0.5542	0.7307	0.6261
	MUL	0.5605	0.7362	0.6316
10	PV	0.5262	0.7175	0.6173
	AVE	0.5180	0.7139	0.6147
	COO	0.5452	0.7256	0.6199
	TF-IDF	0.5392	0.7219	0.6139
	MUL	0.5550	0.7334	0.6219

PV: paragraph vector; AVE: average; COO: co-occurrence; TF-IDF: term frequency–inverse document frequency; MUL: multiplicative composition

We can see that the word vector composition with MUL achieved the best performance. The PV method is comparable to the word vector composition with additive composition approaches. The reason for the superior performance of MUL may be that if vector definitions are optimized by word embedding training, then it is the directions of the vectors that are optimized, not their magnitudes. In multiplicative functions, the magnitudes of the terms of a phrase can affect only the magnitude of the phrase, not its direction. By contrast, in additive models, the relative magnitudes of terms can have a considerable effect on both the magnitude and direction of the phrase. This can lead to difficulties when working with similarity measure, which is insensitive to the magnitudes of vectors. This then leads to differences between the additive and multiplicative models when clustering using distance measures.

In terms of the weighting schemes for additive composition, AVE achieved the worst results. The co-occurrence-based and TF-IDF-based models both had improvements over AVE. The obvious reason is that AVE does not consider the differing importance of terms in subtopic candidate phrases. In contrast, co-occurrence and TF-IDF weightings both consider the situation in which each word of subtopic candidate phrase plays a different role in understanding the query. In all cases, the best of the additive composition models is the co-occurrence-based model. This indicates that the importance of words should be dynamic and query-dependent.

5.4.2 Effects of data types

In this part, we analyze the performance of word vector composition on different training datasets. We chose COO, the best variant of additive composition, and MUL, the best performer, for demonstration. Table 2 shows the experimental results from two types of data. The number of clusters is five. Actually, we observed the same trend for other numbers of clusters.

Table 2 Performance of different training datasets on the evaluation metrics (number of topics *K*=5)

Model	Dataset	I-rec	D-nDCG	D#-nDCG
COO	Baike	0.5131	0.7107	0.6072
	SogouQ	0.5582	0.7328	0.6301
MUL	Baike	0.5698	0.7404	0.6374
	SogouQ	0.5605	0.7362	0.6316

COO: co-occurrence; MUL: multiplicative composition

Surprisingly, we observed that the results are opposite when we used different composition models. COO performed better when using query log data for training, while MUL achieved slightly better results when using Baidu Baike data for training. Generally, query log data are less sensitive to the choice of the composition models. The reason may be that the query log data are more related to our task. The learned distributed representation can better capture the special characteristics of queries. From another perspective, it showed that MUL is more robust and less sensitive to the choice of training data. This may indicate that composition models play a more important role compared with the choice of data.

In summary, we can conclude that: (1) Word vector composition methods outperform the PV method; (2) The MUL strategy is better and more robust compared with the additive composition strategy, and co-occurrence-based term weighting is the most effective for additive composition; (3) Learning proper word embeddings for this task depends on both the choice of data types and composition models, while the query log data are more robust.

5.4.3 Comparison with baselines

We have presented the results of different semantic composition strategies for a subtopic mining task using distributed representations. In this subsection, we will compare three proposed models (PV, COO, and MUL) with two baselines (BOW and CRQ) described in Section 5.3 and the top systems in the NTCIR-9 subtopic mining task. The distributed representations are learned from the query log data.

Table 3 shows the results. Here, ICTIR-S-C-1 and THU-S-C-2 are the best performers on the NTCIR-9 subtopic mining task (Song et al., 2011). THU-S-C-2 is ranked top in terms of D-nDCG and D#-nDCG, and ICTIR-S-C-1 is the best one in terms of I-rec.

Table 3 Systematic comparison of different semantic composition strategies

Model	I-rec	D-nDCG	D#-nDCG
MUL	0.5605	0.7362	0.6316
PV	0.5382	0.7236	0.6304
COO	0.5582	0.7328	0.6301
CRQ	0.5146	0.6835	0.5975
BOW	0.4127	0.6881	0.5868
ICTIR-S-C-1	0.5161	0.6434	0.5797
THU-S-C-2	0.4801	0.7186	0.5993

MUL: multiplicative composition; PV: paragraph vector; COO: co-occurrence; CRQ: clustering reformulated queries; BOW: bag-of-words

We can see that among all these performers, the models based on distributed semantic composition rank on the top in terms of all the three metrics. This verifies that distributed semantic representation can largely improve the quality of subtopic mining. In contrast, traditional BOW-based methods suffer from the semantic gap problem like term mismatch and data sparseness. Although considering the co-occurrence information can be helpful, the nature of the BOW assumption still limits the performance.

6 Conclusions and future work

In this paper, we have proposed the exploitation of the semantic composition of distributed representations to represent query subtopic candidates for query subtopic mining. We have explored two approaches to gain the distributed representations of multi-word query subtopic candidates: PV composition, which learns the representations of sequences directly, and word vector composition, which is based on semantic composition of distributed representations of individual words.

Experimental results illustrated that: (1) Both methods based on semantic composition of distributed representation outperformed the traditional BOW- and co-occurrence-based baselines; (2) Word vector composition using the MUL strategy achieved the best performance, outperforming the PV approach. This fully proved that the semantic composition of distributed representation based methods can greatly improve the performance of query subtopic mining. In the future, we plan to incorporate more contextual information, such as user clicks and related behaviors, as signals to learn better query representations to further improve the performance of query subtopic mining. Learning good representations of long word sequences still requires much effort.

References

Anagnostopoulos I, Razis G, Mylonas P, et al., 2015. Semantic query suggestion using Twitter entities. *Neurocomputing*, 163:137-150. https://doi.org/10.1016/j.neucom.2014.12.090

Baeza-Yates R, Hurtado C, Mendoza M, 2005. Query recommendation using query logs in search engines. *LNCS*, 3268:588-596. https://doi.org/10.1007/978-3-540-30192-9_58

Baroni M, Dinu G, Kruszewski G, 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. Proc 52nd Annual Meeting of the Association for Computational Linguistics, p.238-247. https://doi.org/10.3115/v1/P14-1023

Beeferman D, Berger A, 2000. Agglomerative clustering of a search engine query log. Proc 6th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.407-416. https://doi.org/10.1145/347090.347176

Bengio Y, Ducharme R, Vincent P, et al., 2003. A neural probabilistic language model. *J Mach Learn Res*, 3: 1137-1155.

Clarke CLA, Craswell N, Soboroff I, 2009. Overview of the TREC 2009 web track. 18th Text Retrieval Conf, p.1-9.

Collobert R, Weston J, 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. Proc 25th Int Conf on Machine Learning, p.160-167. https://doi.org/10.1145/1390156.1390177

Damien A, Zhang M, Liu Y, et al., 2013. Improve web search diversification with intent subtopic mining. *CCIS*, 400: 322-333. https://doi.org/10.1007/978-3-642-41644-6_30

Dang V, Xue X, Croft WB, 2011. Inferring query aspects from reformulations using clustering. Proc 20th ACM Int Conf on Information and Knowledge Management, p.2117-2120. https://doi.org/10.1145/2063576.2063904

Grefenstette E, Dinu G, Zhang YZ, et al., 2013. Multi-step regression learning for compositional distributional semantics. https://arxiv.org/abs/1301.6939

Hu J, Wang G, Lochovsky F, et al., 2009. Understanding user's query intent with Wikipedia. Proc 18th Int Conf on World Wide Web, p.471-480. https://doi.org/10.1145/1526709.1526773

Hu Y, Qian Y, Li H, et al., 2012. Mining query subtopics from search log data. Proc 35th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.305-314. https://doi.org/10.1145/2348283.2348327

Jiang X, Han X, Sun L, 2011. ISCAS at subtopic mining task in NTCIR9. Proc NTCIR-9 Workshop Meeting, p.168-171.

Joho H, Kishida K, 2014. Overview of NTCIR-11. Proc 11th NTCIR Conf, p.1-7.

Jones R, Klinkner KL, 2008. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. Proc 17th ACM Conf on Information and Knowledge Management, p.699-708. https://doi.org/10.1145/1458082.1458176

Karunasekera S, Harwood A, Samarawickrama S, et al., 2014. Topic-specific post identification in microblog streams. IEEE Int Conf on Big Data, p.7-13. https://doi.org/10.1109/BigData.2014.7004416

Kim SJ, Lee JH, 2013. Subtopic mining based on head-modifier relation and co-occurrence of intents using web documents. *LNCS*, 8138:179-191. https://doi.org/10.1007/978-3-642-40802-1_22

Le Q, Mikolov T, 2014. Distributed representations of sentences and documents. Proc 31st Int Conf on Machine Learning, p.1188-1196.

Li X, Wang YY, Acero A, 2008. Learning query intent from regularized click graphs. Proc 31st Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.339-346. https://doi.org/10.1145/1390334.1390393

Liu Y, Song R, Zhang M, et al., 2014. Overview of the NTCIR-11 IMine task. Proc 11th NTCIR Conf, p.8-23.

Luo C, Liu Y, Zhang M, et al., 2014. Query recommendation based on user intent recognition. *J Chin Inform Process*, 28(1):64-72 (in Chinese). https://doi.org/10.3969/j.issn.1003-0077.2014.01.009

Mikolov T, Chen K, Corrado G, et al., 2013a. Efficient estimation of word representations in vector space. https://arxiv.org/abs/1301.3781

Mikolov T, Yih WT, Zweig G, 2013b. Linguistic regularities in continuous space word representations. Proc NAACL-HLT, p.746-751.

Mitchell J, Lapata M, 2010. Composition in distributional models of semantics. *Cogn Sci*, 34(8):1388-1429. https://doi.org/10.1111/j.1551-6709.2010.01106.x

Mnih A, Hinton G, 2007. Three new graphical models for statistical language modelling. Proc 24th Int Conf on Machine Learning, p.641-648. https://doi.org/10.1145/1273496.1273577

Radlinski F, Szummer M, Craswell N, 2010. Inferring query intent from reformulations and clicks. Proc 19th Int Conf on World Wide Web, p.1171-1172. https://doi.org/10.1145/1772690.1772859

Rafiei D, Bharat K, Shukla A, 2010. Diversifying web search results. Proc 19th Int Conf on World Wide Web, p.781-790. https://doi.org/10.1145/1772690.1772770

Sakai T, Song R, 2011. Evaluating diversified search results using per-intent graded relevance. Proc 34th Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.1043-1052. https://doi.org/10.1145/2009916.2010055

Sakai T, Dou Z, Yamamoto T, et al., 2013. Overview of the NTCIR-10 INTENT-2 task. Proc 10th NTCIR Conf, p.94-123.

Santos RLT, Macdonald C, Ounis I, 2010. Exploiting query reformulations for web search result diversification. Proc 19th Int Conf on World Wide Web, p.881-890. https://doi.org/10.1145/1772690.1772780

Socher R, Lin CC, Ng AY, et al., 2011a. Parsing natural scenes and natural language with recursive neural networks. Proc 28th Int Conf on Machine Learning, p.129-136.

Socher R, Pennington J, Huang EH, et al., 2011b. Semi-supervised recursive autoencoders for predicting sentiment distributions. Proc Conf on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, p.151-161.

Song R, Luo Z, Nie JY, et al., 2009. Identification of ambiguous queries in web search. *Inform Process Manag*, 45(2): 216-229. https://doi.org/10.1016/j.ipm.2008.09.005

Song R, Zhang M, Sakai T, et al., 2011. Overview of the NTCIR-9 INTENT task. Proc NTCIR-9 Workshop Meeting, p.82-105.

Song W, Yu Q, Xu ZH, et al., 2012. Multi-aspect query summarization by composite query. Proc 35th Int ACM SIGIR Conf on Research and development in Information Retrieval, p.325-334. https://doi.org/10.1145/2348283.2348329

Song W, Liu Y, Liu L, et al., 2016. Examining personalization heuristics by topical analysis of query log. *Int J Innov Comput Inform Contr*, 12(5):1745-1760.

Strohmaier M, Kröll M, Körner C, 2009. Intentional query suggestion: making user goals more explicit during search. Proc Workshop on Web Search Click Data, p.68-74. https://doi.org/10.1145/1507509.1507520

Wang CJ, Lin YW, Tsai MF, et al., 2013. Mining subtopics from different aspects for diversifying search results. *Inform Retriev*, 16(4):452-483. https://doi.org/10.1007/s10791-012-9215-y

Xu J, Croft WB, 1996. Query expansion using local and global

document analysis. Proc 19th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.4-11.
https://doi.org/10.1145/243199.243202

Yu M, Dredze M, 2015. Learning composition models for phrase embeddings. *Trans Assoc Comput Ling*, 3:227-242.

Zanzotto FM, Korkontzelos I, Fallucchi F, et al., 2010. Estimating linear models for compositional distributional semantics. Proc 23rd Int Conf on Computational Linguistics, p.1263-1271.

Zeng HJ, He QC, Chen Z, et al., 2004. Learning to cluster web search results. Proc 27th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.210-217.
https://doi.org/10.1145/1008992.1009030

Zhao Y, Liu Z, Sun M, 2015. Phrase type sensitive tensor indexing model for semantic composition. Proc 29th AAAI Conf on Artificial Intelligence, p.2195-2201.

Zheng W, Fang H, 2011. A comparative study of search result diversification methods. 1st Int Workshop on Diversity in Document Retrieval, p.55-62.

Zheng W, Fang H, Yao C, et al., 2014. Leveraging integrated information to extract query subtopics for search result diversification. *Inform Retriev*, 17(1):52-73.
https://doi.org/10.1007/s10791-013-9228-1