Leveraging Pattern Associations for Word Embedding Models

Qian Liu[1,2], Heyan Huang[1], Yang Gao*, Xiaochi Wei[1], and Ruiying Geng[1]

1. School of Computer Science and Technology, Beijing Institute of Technology
2. Faculty of Engineering and Information Technology, University of Technology Sydney

Abstract. Word embedding method has been shown powerful to capture words association, and facilitated numerous applications by effectively bridging lexical gaps. Word semantic is encoded with vectors and modeled based on n-gram language models, as a result it only takes into consideration of words co-occurrences in a shallow slide windows. However, the assumption of the language modeling ignores valuable associations between words in a long distance beyond n-gram coverage. In this paper, we argue that it is beneficial to jointly modeling both surrounding context and flexible associative patterns so that the model can cover long distance and intensive association. We propose a novel approach to combine associated patterns for word embedding method via joint training objection. We apply our model for query expansion in document retrieval task. Experimental results show that the proposed method can perform significantly better than the state-of-the-arts baseline models.

1 Introduction

As an improvement of traditional one-hot word representation method, word embedding overcomes the data sparsity, high dimensional, and lexical gap problems by capturing both word semantics and syntactics with dense vectors. A great efforts have been conducted to construct prominent word embedding [4, 16, 21, 10, 25, 23, 20], and they are widely used in natural language processing tasks to compute word semantical similarity and regularity.

Most existing word embedding methods generate word vectors based on its surrounding context, encoding only shallow adjacency information. However, in real world, many valuable associative relationships between words actually exist in a longer linguistic distance instead of adjacent rules. A few examples are shown in Fig. 1(a). As can be seen from these sentences, *"programming languages"* and *"algorithm"* hold intensive relations, but they are distributed apart in long distance. And it's hard for a local slide window (i.e. the length of slide window defines as n=5) to cover with.

It is worth mentioning that association rule mining[1, 27], as a data analysis technique, is generally used for discovering frequently co-occurring data items

* Corresponding author.

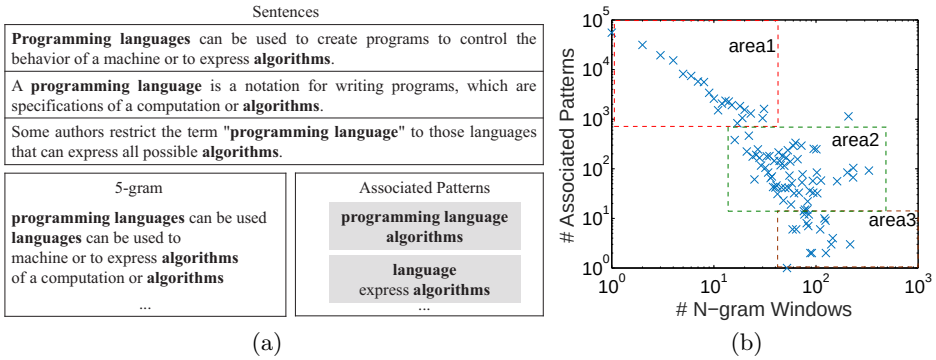


Fig. 1. Examples of associated words in long sentences. (a) shows several sentences, and associated words in the sentences are in bold. Related words are shown using n-gram (n=5) and associated patterns respectively. (b) compares the ability of mining related words in a 5-gram context window and associated patterns. Relations in area1 can be discovered by associated patterns while hard by context. N-gram method is good at capturing relations in area3. Both methods work well in discovering relations in area2.

without the limitation of n-gram coverage. We employ association rule mining in text data processing to mine frequent patterns, and utilise the discovered association rules among them as trustful relationships between individual words. Fig. 1(a) present several associated items mined from sentences using 5-gram and associated patterns respectively. As we can see, several intensive relations (i.e. *"programming languages"* with *"algorithm"*, *"language"* with *"express algorithm"*) cannot be discovered in 5-gram, while can be easily captured by pattern mining methods.

Associated patterns can discover rich words associations[3, 7, 9] but not yet well-utilised in generating word embedding. To illustrate this fact, we visually show the relationship between the number of word co-occurrences in n-gram windows and the number of same related words in associated patterns. The result is shown in Fig. 1(b). In the figure, the X-axis denotes the number of two words co-occurrence in n-gram model window (n=5) and the Y-axis denotes the number of two words co-occurrence in association patterns. We find that the statistic follows a power-law distribution, which means that most frequently appeared patterns can hardly be fetched by a window with a fixed-length. Specially, a great number of related words in associated patterns (marked as area1) cannot be found in local context word co-occurrence. This indicates that rich word associations in the corpus may be ignored if the word embedding methods only consider local contextual information, although area2 and area3 demonstrate that n-gram models can also cover reasonable number of related words.

In order to solve the urging problem of losing word associative relations of language models, in this paper, we propose an Associated Patterns enhanced Word Embedding model, APWE for short. In this model, associated patterns

are introduced and jointly used with context information to predict the target word, so that both local co-occurrence and long distance associated relations are considered in the generated word embedding. Further more, we apply APWE method for query expansion in document retrieval task, and the results demonstrate that our methods outperform state-of-the-arts methods.

The main contributions of our work are summarized as follows:

- Association patterns are wisely integrated in the process of generating semantic word embeddings. The new word embedding model can capture intensive word associations beyond n-gram limitation which is covered by a sentence level.
- To enable word relations discovery more sensitive for word embedding in a flexible and fast way, especially under circumstances of insufficient datasets.
- We conduct experiments to demonstrate the effectiveness of our method for document modeling in the real application.

The rest of this paper is organized as follows. Section 2 summarizes the background of our methods. We then proposed associated patterns enhanced word embedding model in Section 3. Section 4 reports the experimental results. Section 5 surveys the related work. Finally, we conclude the paper in Section 6.

2 Background

2.1 Word2Vec Method

Recently, neural networks relevant methods have been introduced to model languages with promising results. Especially, Mikolov et al.[16, 15] proposed Word2Vec method is an efficient method for learning high quality word embedding from large-scale unstructured text data.

The basic assumption behind Word2Vec is that the representation of co-occurrenced words have the similar representation in the semantic space. To this target, a sliding window is employed on the input text stream, where the central word is the target word and others are contexts. Word2Vec method contains two models: continuous bag-of-word model (CBOW) and Skip-gram model.

CBOW aims at predicting the target word using the context words in the sliding window. Formally, given a word sequence $\mathcal{D} = \{w_{i-k}, ..., w_{i-1}, w_i, w_{i+1}, ..., w_{i+k}\}$, the objective of CBOW is to maximize the average log probability

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

where, w_i is the target word, 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

$$Pr(w_i \mid w_{i-k}, ..., w_{i-1}, w_{i+1}, ..., w_{i+k}) = \frac{\exp(\mathbf{x}_i \cdot \mathbf{x}_c)}{\sum_{w \in \mathcal{W}} \exp(\mathbf{x} \cdot \mathbf{x}_c)}, \tag{2}$$

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 of all context word vectors.

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

$$L(\mathcal{D}) = \frac{1}{T}\sum_{i=1}^{T}\sum_{-k\leqslant c\leqslant k, c\neq 0} \log Pr(w_{i+c} \mid w_i), \tag{3}$$

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

$$Pr(w_{i+c} \mid w_i) = \frac{\exp(\mathbf{x}_{i+c} \cdot \mathbf{x}_i)}{\sum_{w\in\mathcal{W}} \exp(\mathbf{x} \cdot \mathbf{x}_i)}, \tag{4}$$

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 been shown useful in many applications. Nevertheless, most existing works learn word representations mainly based on the word co-occurrences, therefore the obtained word embedding cannot capture associated words if either of them yield very little context information. On the other hand, in small and insufficient datasets, word co-occurrences may be sparse and unreliable, which may mislead the training process. In this paper, we extend Word2Vec method leveraging associated patterns to further improve the word representation.

2.2 Associated Patterns

In this paper, we employ association rule mining in text data processing to discover associated patterns. Association rule mining, as a data analysis technique, is generally used for discovering frequently co-occurring data items, and aims to discover hidden rules among enormous pattern combinations based on their individual and conditional frequencies.

An association rule contains two patterns, i.e. an antecedent pattern X and a consequent pattern Y. These two patterns are considered to be a rule if its frequency satisfies a minimum support threshold (T_s) and the conditional probability satisfies a minimum confidence threshold (T_c). Formally, this can be expressed as

$$\begin{cases} Supp(X \cup Y) \geq T_s, \\ Conf(X \rightarrow Y) = \dfrac{(Supp(X \cup Y))}{Supp(X)} \geq T_c, \end{cases} \tag{5}$$

where $Supp(\cdot)$ is the frequency support, and $Conf(X \rightarrow Y)$ indicates the conditional probability of X's occurrence implies Y's occurrence. The support can be considered as a global measure of being interesting, and the confidence is used as a localization measure.

Based on the association rule, in this paper we regard X and Y as associated patterns.

Definition (Associated Patterns). Let $\mathcal{I} = \{w_1, w_2, ..., w_n\}$ be a set of items, pattern X and Y are associated patterns, if :(1) $X \subseteq \mathcal{I}, Y \subseteq \mathcal{I}, X \bigcap Y = \emptyset$; (2) $Supp(X) \geq T_s$, $Supp(Y) \geq T_s$; (3)$Conf(X \rightarrow Y) \geq T_c$.

In an associated patterns pair, we denote the antecedent pattern as pat_A and the consequent pattern as pat_C. For example, in Fig. 1, *"program language"* and *"algorithm"* are regard as associated patterns, where pat_A = *"program language"* and pat_C =*"algorithm"*.

After generating associated patterns, we need to align target word with its associated words for training the model. We denote $\mathcal{PAT}(w_i)$ as the pattern set associated with word w_i. According to associated patterns' definition, there are two roles of pattern, antecedent and consequent. Therefore, we also define $\mathcal{PAT}(w_i)$ contains two subsets, word w_i's antecedent set $\mathcal{PAT}^A(w_i)$ and w_i's consequence set $\mathcal{PAT}^C(w_i)$. For each associated patterns, if pat_C contains w_i, we select pat_A as one of w_i associated patterns in $\mathcal{PAT}^A(w_i)$. While if pat_A contains w_i, we select pat_C as one of w_i associated patterns in $\mathcal{PAT}^C(w_i)$. Hence, $\mathcal{PAT}(w_i)$ contains two subsets, $\mathcal{PAT}^A(w_i)$ is the pattern collection to predict word w_i; $\mathcal{PAT}^C(w_i)$ is the pattern collection which can be predicted by w_i. In our method, different subsets are used for training different models. The details are described as following section.

3 Model

In this section, we describe the details of the proposed associated patterns enhanced word embedding (APWE) method. Follow basic structures of generating word embedding of the Word2Vec model, we propose our novel APWE model in terms of CBOW model and Skip-gram model, and theoretically demonstrate the flexibility and applicability of the integrating approach. At last, we describe how to apply the proposed embedding method for query expansion.

3.1 Pattern Enhanced Word Embedding Model

In this subsection, we detail the approach of generating newly word embedding with associated patterns. The basic idea is that associated patterns encode the words co-occurrence information in sentence level, from which we can extract hidden relations in long distance. We refine the word embedding model according to both local context and additionally long distance patterns, in order to consider word similarity and relatedness at the same time.

Model 1 (Associated Patterns Enhanced CBOW Model). In this model, we use both context and associated patterns to predict the target word. The associated patterns set is generated with the aforementioned rules, and we choose $\mathcal{PAT}^A(w_i)$ subset (we defined it in Section 2.2) to predict target word w_i. We named this mode as antecedent associated pattern enhanced word embedding models, and we use A-APWE model for easy of reference.

The model architecture is shown in Figure. 2, it contains two prediction tasks. The former prediction task captures word relations in local level, since words with

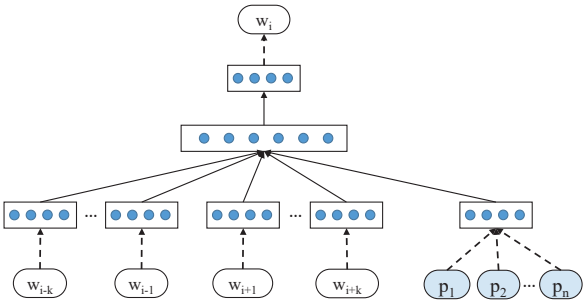


Fig. 2. Associated patterns enhanced CBOW model, we denote as A-APWE.

similar context tend to have similar representation. The latter prediction task capture words association in long distance, since words frequently co-occure in the sentence tend to have similar representation. To joint context and associated patterns in the model, we encode the associated patterns using the following objection function

$$L(\mathcal{PAT}^A(w_i)) = \sum_{pat_j \in \mathcal{PAT}^A(w_i)} \log Pr(w_i \mid pat_j). \tag{6}$$

In this function ,we present a pattern using its terms average vector, and $Pr(w_i \mid pat_j)$ is also a softmax funciton

$$Pr(w_i \mid pat_j) = \frac{\exp(\mathbf{x}_i \cdot \mathbf{x}_{pat_j})}{\sum_{w \in \mathcal{W}} \exp(\mathbf{x} \cdot \mathbf{x}_{pat_j})}. \tag{7}$$

Combining Eqn. 6 with existing CBOW model(Eqn. 1), we obtained the following objective function,

$$L = \frac{1}{T}\sum_{i=1}^{T}(\log Pr(w_i \mid w_{i-k}, ..., w_{i+k}) + \alpha \sum_{n=1}^{N} \log Pr(w_i \mid pat_n^{w_i})), \tag{8}$$

where α is the combination coefficient. We follow the similar optimization scheme as CBOW model and adopt the negative sampling technique for learning A-APWE model. The training objective function is defined as

$$\begin{aligned} \ell = &\sum_{i=1}^{T}(log\sigma(\mathbf{w_i} \cdot \mathbf{w_c}) + k \cdot \mathcal{N}(w' \sim w_i) \cdot log\sigma(\mathbf{w}' \cdot \mathbf{w_c}) \\ &+ \sum_{n=1}^{N}(log\sigma(\mathbf{w_i} \cdot \mathbf{pat_n}) + k \cdot \mathcal{N}(w' \sim w_i) \cdot log\sigma(\mathbf{w}' \cdot \mathbf{pat_n}))), \end{aligned} \tag{9}$$

where $\sigma(x) = 1/(1+exp(-x))$ and k is the number of negative samples. $\mathcal{N}(w' \sim w_i)$ denotes the sampled word collection of word w_i, and w' represents one sampled word. We use stochastic gradient descent(SGD) for optimization, and gradients are calculated using the back propagation neural networks.

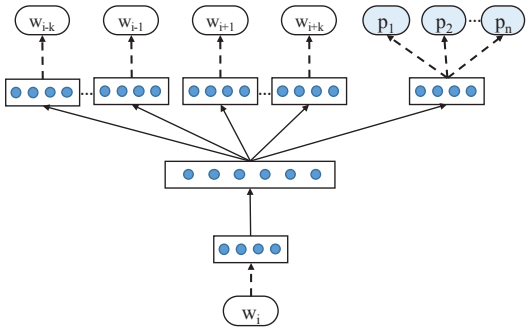


Fig. 3. Associated patterns enhanced Skip-gram model, we denote it as C-APWE

Model 2 (Associated Patterns Enhanced Skip-gram Model). We use the target word to predict context and its associated patterns in this model. Similar with model 1, we need to mine the pattern collection with the aforementioned rules firstly. However, in this model, the basic idea is using target word to predict patterns. Therefore, for target word w_i, we select $\mathcal{PAT}^C(w_i)$ (we defined it in Section 2.2) as its associated patterns set. We denote this model as consequent associated patterns enhanced word embedding model, C-APWE model for short. Fig. 3 shows the model architecture. The target word is used to predict its context, as well as its associated patterns. Formally, we encode the associated patterns using the following objection function

$$L(\mathcal{PAT}^C(w_i)) = \sum_{pat_j \in \mathcal{PAT}^C(w_i)} \sum_{w_j \in pat_j} \log Pr(w_j \mid w_i), \tag{10}$$

where $Pr(pat_j \mid w_i)$ is also a softmax function

$$Pr(w_j \mid w_i) = \frac{\exp(\mathbf{x}_j \cdot \mathbf{x}_{w_i})}{\sum_{w \in \mathcal{W}} \exp(\mathbf{x} \cdot \mathbf{x}_{w_i})}. \tag{11}$$

Combining Eqn. 10 with existing Skip-gram model(Eqn. 3), we obtained the following objective function

$$L = \frac{1}{T}\sum_{i=1}^{T}(\sum_{-k \le c \le k, c \neq 0} \log Pr(w_{i+c} \mid w_i) + \beta \sum_{n=1}^{N} \log Pr(pat_n^{w_i} \mid w_i)), \tag{12}$$

where β is the combination coefficient. When it comes to optimization, C-APWE model adopts negative sampling method and follows the similar optimization scheme as Skip-gram model. The target objection can be represented as

$$\begin{aligned} \ell = \sum_{i=1}^{T}(& \sum_{-k \le c \le k, c \neq 0} (log\sigma(\mathbf{w_{i+c}} \cdot \mathbf{w_i}) + k \cdot \mathcal{N}(w' \sim w_{i+c}) \cdot log\sigma(\mathbf{w'} \cdot \mathbf{w_i}) \\ & + \sum_{n=1}^{N} \sum_{w_p \in pat_n} (log\sigma(\mathbf{w_p} \cdot \mathbf{w_i}) + k \cdot \mathcal{N}(w' \sim w_p) \cdot log\sigma(\mathbf{w'} \cdot \mathbf{w_i}))), \end{aligned} \tag{13}$$

where k is the number of negative samples, $\mathcal{N}(w' \sim w_{i+c})$ denotes the negative samples collection of context word w_{i+c} and $\mathcal{N}(w' \sim w_p)$ denotes the negative samples collection of word w_p in pattern pat_n. We use SGD for optimization, and gradients are calculated using the back propagation neural networks.

3.2 Query Expansion for Document Retrieval

In information filtering tasks, document queries are formulated using several terms. Term-matching retrieval functions could fail at retrieving relevant documents if they cannot judge word semantic similarity, which also known as lexicon gap problem. Expanding queries based on words semantic meaning could enhance the likelihood of retrieving relevant documents. In this subsection, we detail how our proposed methods are used in query expansion.

Specifically, we obtained the word embedding with the proposed APWE models. Then, given a query q, we construct expansion set $Q^+ = \{q_1^+, ..., q_n^+\}$ by selecting top n most similar words with the cosine similarity. Each term q^+ is associated with a term weight according to its cosine distance to query q. Then the final expansion of query q is represented as

$$Q = q \bigcup Q^+. \tag{14}$$

Formally, in the information ranking task, we define the computation method on Q as

$$f(Q) = f(q) \cdot (1 + \gamma \sum_{q_i^+ \in Q^+} \cos(v_{q_i^+}, v_q)), \tag{15}$$

where γ is the combination coefficient, and $f(q)$ is the origin function on query term in the retrieval model (i.e. word frequency, vector representation).

4 Experiments

In this section, we present experiments to evaluate the performance of our method in document filtering task. We discuss the experiments and evaluation in terms of dataset, baseline models and setting, measures and results. The results show that query expansion based on our APWE method significantly outperforms the-state-of-the-arts models in terms of effectiveness.

4.1 Dataset

To evaluate the performance of the proposed method with existing different baseline approaches, we conducted our experiments using the Reuters Corpus Volume 1 (RCV1) dataset, which is widely used in document ranking task.

In RCV1 dataset, there are a total of 806,791 documents that cover a variety of topics and a large amount of information. These documents are divided into 100 collections in total, and each collection is divided into a testing set and

Table 1. Statistic of RCV1 Dataset

# Documents	Corpus	Vocabulary Size	# Sentences	# Associated Patterns
806,791	70.1M	111,257	20,300	88,564

a training set. The first 50 collections were composed by human assessors and another 50 collections were constructed artificially from intersections collections. In this paper, only the first 50 collections used for experiments. Each document contains $'title'$ and $'text'$, and these parts were used by all the models in the experiments. We then tokenized all text in the dataset with the help of Stanford tokenizer tooland we at last converted every word into lower case.

To train word embedding, we combined all documents in RCV1 dataset as the training corpus, which includes 16 million words. In this paper, we mined the association patterns in the sentence level. We hence segmented $'text'$ of positive labeled documents into sentences and combined all sentences together for association rules mining. In total, we generated 88,564 associated patterns. More statistic of the dataset is given in Table. 1.

4.2 Measures

In order to evaluate the performance, we apply four standard evaluation metrics: average precision of the top 10 documents ($P@10$), $F1$ measure, Mean Average Precision (MAP), break-even point (b/p). The precision is the proportion of labeled documents identified by the model which are correct. The recall is the proportion of labeled documents in the result which are correctly identified by the model. $F1$ measure is a criterion that assesses the effect involving both precision (p) and recall (r), which is defined as $F1 = \frac{2pr}{p+r}$. The larger the $Percision@10$, MAP, b/p, $F1$ score, the better the system performs.

4.3 Baseline Models and Settings

We choose BM25[22], Word2Vec, Topical Word Embedding[14] (TWE), and GloVe[21] as baseline methods. The underlying idea of using these methods are highlighted below.

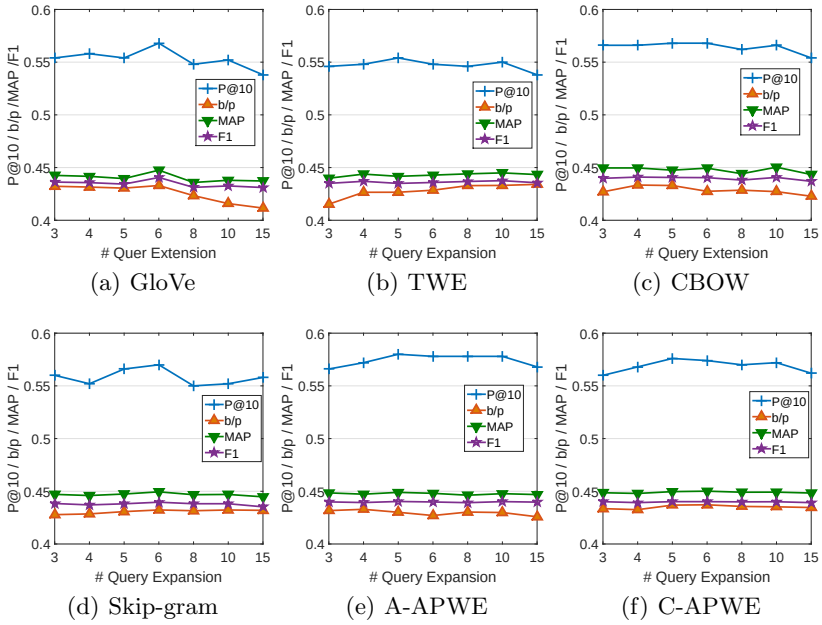
BM25 is one of the state-of-the-arts term-based document ranking approach. The term weights are estimated using the following equation:

$$W(t) = \frac{tf \cdot (k+1)}{k \cdot ((1-b) + b \cdot \frac{dl}{avgdl})} \cdot \log(\frac{N - n + 0.5}{n + 0.5}), \tag{16}$$

where N is the number of documents in the collection; n is the number of documents which contain term t; tf represents term frequency; dl is the document length and $avdl$ is the average document length.

Word2Vec, as mention above, is the-state-of-the-arts model which captures word relations based on local word co-occurrence information.

TWE method employs the widely used latent dirichlet allocation (LDA)[5]to refine Skip-gram model. TWE model can employ topic models to take advantages of all words as well as their context together to learn topical word embeddings.



(a) GloVe (b) TWE (c) CBOW

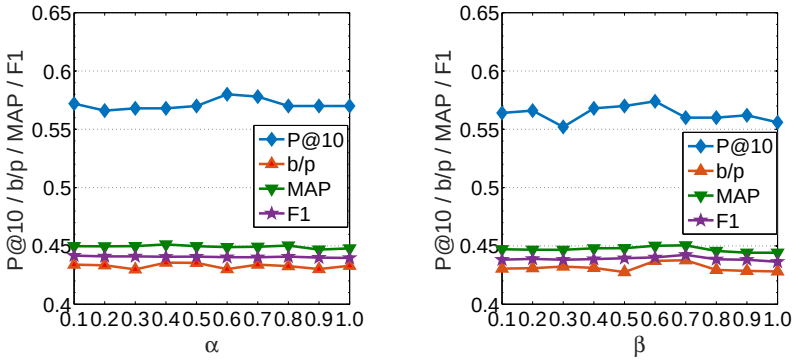
(d) Skip-gram (e) A-APWE (f) C-APWE

Fig. 4. The size of query expansion.

GloVe method is an weighted least squares regression model that performs global matrix factorization with a local context window models. GloVe model leverages statistical information by training only on the nonzero elements in a word to word co-occurrence matrix, rather than on the entire sparse matrix or on individual context windows in a large corpus.

As to the document filtering task, for each collection, we generated document queries according to term's BM25 weight. We selected top 10 terms as collection's queries and expanded them using different word embedding methods. There are two parameters in BM25 method, b and k. Following the experimental setting in [22], k is set to 1.2 and b is set to 0.75. Another essential parameter is the size of Q^+ in Eqn.14, which indicates how many associated terms were used to expand the query. We carefully tuned this parameter for different word embedding methods in terms of $P@10$, b/p, MAP and $F1$. As Fig. 4 shows, the results with 5 expanding range in our method achieves the best performance for this particular dataset. Therefore, we set number of expanding terms as 5 in APWE method. We also carefully tuned the size of Q^+ in other baseline methods, and seleted 6,5,5 and 6 for GloVe, TWE, CBOW and Skip-gram mehtod respectively. Word embedding size is 300 in all these methods.

Besides, in our APWE models, there are four parameters, T_s, T_c, α and β. The first two are the threshold of *support* and $confidence$ respectively, used in associated patterns mining. The last two are used for combination adjustment in different APWE models. We experiential set the T_s to 5 and T_c to 0.7 to take the best balance between accuracy and comprehension for extracting word



(a) Effect on α in A-APWE model (b) Effect on β in C-APWE model

Fig. 5. Combination parameters.

associations. In order to guarantee the quality of word's associated pattern set, we select no more than ten words regarding to pattern's support for $\mathcal{PAT}^A$ and $\mathcal{PAT}^C$. The parameter α used in A-APWE model and β used in C-APWE model control the contribution of pattern information in the training process. Therefore, we do an experiment on the validation dataset to determine the best α and β in terms of $P@10$, b/p, MAP and $F1$. We carefully tuned α and β from 0.1 to 1.0, with the step size of 0.1, the parameters corresponding to the best $P@10$ value are used to report the final result. As the result shown in Fig. 5, we observed that both of our models reached the optimal performance when combination weight is 0.6. We find that the performance is relatively stable around the optimal parameter.

4.4 Overall Performances

Table. 2 shows the document ranking performances comparison between our model and baselines. From the table, we can observe that: 1) Using continuous word embedding as a foundation of expanding queries is an effective way for retrieving documents. 2) The proposed A-APWE model, which is updated according to CBOW model, is outperforming CBOW based Word2Vec model, and the proposed C-APWE model, which is updated according to Skip-gram model, is better than Skip-gram based Word2Vec model, respectively. From the both results of APWE models, we can imply that combing patterns is a flexible and applicable way to capture long distance word relationships rather than n-gram assumption. It also demonstrates that the association patterns are effectively boosting to capture semantical similarities, especially for document retrieval modeling. 3) The APWE models are consistantly superior to the GloVe and the TWE model. As aforementioned, the TWE model improved Word2Vec model by integrating topic assignment for generating word vectors, while the GloVe model considered global information for local calculations. However, compared

Table 2. Overall performances

Methods	P@10	b/p	MAP	F1
BM25	0.446	0.406	0.408	0.415
GloVe+QE	0.562	0.434	0.449	0.440
TWE+QE	0.554	0.427	0.442	0.435
CBOW+QE	0.568	0.433	0.448	0.440
Skip-gram+QE	0.570	0.432	0.449	0.440
A-APWE+QE	**0.580**	0.430	0.449	**0.440**
C-APWE+QE	0.576	**0.437**	**0.450**	**0.440**

with our proposed model, none of them take into consideration of those intensive and trustful words associations that are often distributed in a long range but actually semantically related. That is why our models are outperforming those state-of-the-arts high-quality word embeddings for document modeling.

We also find that the percentage of the A-APWE model outperforming CBOW model is higher than the counterpart percentage of C-APWE to Skip-gram model. Word prediction upon patterns and surrounding context is under more trustful and sufficient condition. However, conversely utilising single word is not that convincing to predict accurate patterns. Hence, A-APWE achieved the best performance. And associated patterns' quality promotion can further enhance models' performance.

4.5 Case Study

A case study was conducted to analyze in-depth reason why the APWE models surpass conventional candidate selection methods. Several examples are listed in Table. 3. We present the most similar 5 words with cosine distance to a given target word under different word embedding approaches.

Two facts mainly accounted for the failure of traditional word embedding methods, according to the observation in our datasets. The first is that, in view of word similarity, unrelated words are selected in baseline methods, i.e. in GloVe method, *horizons* was regarded as a similar word to *computer*; in Skip-gram method, *credibity* was regarded as a similar word to *arms*, which is either unreasonable. Another problem is that, language semantic is complicated which is formed by different granularities when expressing a general or specific level of semantic meaning. But the baseline models can hardly capture the structural levels simply by similarities. For example, *javastation* is more specific than *computer* at abstraction level, yet they are in a same list of similar words in the Skip-gram model, and same examples happen in the CBOW and the GloVe and so forth. From the results in Table 3, we can find that similar words discovered by our proposed models are mostly distributed at a same abstraction level, which can keep coherent semantics.

As can be seen, the most similar words of APWE methods and baseline models are in common, i.e. the most similar word for target word *arms* is *weapons* in all methods. This demonstrates that our models also take the important

Table 3. Target words and their 5 most similar words under different models.

computer					
GloVe	assoc	navio	dell	laptop	horizons
TWE	software	computers	networking	macintosh	handheld
CBOW	motherson	audio	mediwar	disk	comint
Skip-gram	software	computers	legent	playback	javastation
A-APWE	computers	software	computing	pc	workstations
C-APWE	software	computers	internet	embed	hpcs
arms					
GloVe	weapons	cache	decommissioning	morgane	proliferation
TWE	weapons	baghdads	arsenal	cooperates	nikita
CBOW	weapons	unsafeguarded	missile	weaponry	airacraft
Skip-gram	weapons	credibity	ballastic	churns	dustruction
A-APWE	weapons	missile	iraq	weaponry	haemorrhaging
C-APWE	weapons	ballastic	bilogical	churns	iraq

context information into consideration, and the wisely joint surrounding context and associative patterns while learning word embeddings.

5 Related Work

Representing words using fixed-length vectors is an essential step in text processing tasks. In the early stage, one-hot representations have been widely used for its simplicity and efficiency. However, this traditional representation method suffers from data sparsity, the curse of dimensions and lexical gap, which make NLP and IR tasks difficult to use. Distributed word representation, also known as word embedding, is then introduced to solve these problems. In this method, words are represented as dense, low-dimensional, real-valued vectors, and each dimension represents latent semantic and syntactic features of words.

Recently, there is a surge of works focusing on Neural Network (NN) algorithms for word representations learning (Bengio et al[4];Mnih and Hinton[17]; Collobert et al[8]; Mikolov et al[16, 15]; Mnih and Kavukcuoglu[18]; Lebret and Collobert[11]; Pennington et al[21]). Most of these methods hold the assumption that words with similar context tend to have similar meanings.

Several researches focus on generating word representation beyond the context-based assumption, considering deeper relationships between linguistic items[13]. The Strudel system [2] represented a word using the clusters of lexical-syntactic patterns in which it occurs. Murphy et al[19] represented words through their co-occurrence with other words in syntactic dependency relations, and then used the Non-Negative Sparse Embedding (NNSE) method to reduce the dimension of the resulted representation. Levy and Goldberg [12] extended the Skip-gram Word2Vec model with negative sampling [**?**] by basing the word co-occurrence window on the dependency parse tree of the sentence.

There are also several works have been done to use patterns enhanced performance of word embedding. Pattern has been suggested to be useful to capture word co-occurred relations in word representation [26]. Bollegala et al[6] replaced

bag-of-words contexts with various patterns (lexical, POS and dependency). Roy Schwartz et al[24] proposed a symmetric pattern based approach to word representation which is particularly suitable for capturing word similarity. These works show the effectiveness of patterns in word representations. As mentioned above, local context also has shown its effective in capture word relatedness. To improve word representation, we consider both context and associated pattern, and leverage pattern associations for word embedding model.

6 Conclusion

This paper presents an innovative associated patterns enhanced word embedding method for covering word associations both in local and long distance. The proposed APWE method employs associated patterns to model associated words which occur in complex sentences without adjacency rules. APWE models extend the Word2Vec method by both using context information and associated patterns to capture word relatedness and semantics. Our method has been evaluated by using RCV1 for the task of document filtering. In comparison with the-state-of-the-arts models, the proposed models demonstrate excellent strength on query expansion and document ranking. As part of our future work, we will focus our efforts on dealing with those unseen words in large amount of data, since the APWE models in this stage can not well deal with the problem of the data size arises. We plan to collaborate conceptual assistance with the word associations for modeling unseen words and try to improve the robustness of the APWE models in the future.

Acknowledgments

The work was supported by National Nature Science Foundation of China (Grant No.61132009), National Basic Research Program of China (973 Program, Grant No.2013CB329303).

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB'94. pp. 487–499 (1994)
2. Baroni, M., Murphy, B., Barbu, E., Poesio, M.: Strudel: A corpus-based semantic model based on properties and types. Cognitive Science'10 34(2), 222–254 (2010)
3. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. SIGKDD'00 2(2), 66–75 (2000)
4. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. JMLR'03 3, 1137–1155 (2003)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR'03 3, 993–1022 (2003)
6. Bollegala, D., Maehara, T., Yoshida, Y., Kawarabayashi, K.: Learning word representations from relational graphs. In: AAAI'15. pp. 2146–2152 (2015)

7. Cheng, H., Yan, X., Han, J., Hsu, C.: Discriminative frequent pattern analysis for effective classification. In: ICDE'07. pp. 716–725 (2007)
8. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. JMLR'11 12, 2493–2537 (2011)
9. Gao, Y., Xu, Y., Li, Y.: Pattern-based topics for document modelling in information filtering. TKDE'15 27(6), 1629–1642 (2015)
10. Iacobacci, I., Pilehvar, M.T., Navigli, R.: Sensembed: Learning sense embeddings for word and relational similarity. In: ACL'15. pp. 95–105 (2015)
11. Lebret, R., Collobert, R.: Word embeddings through hellinger PCA. In: EACL'14. pp. 482–490 (2014)
12. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: ACL'14. pp. 302–308 (2014)
13. Li, J., Li, J., Fu, X., Masud, M.A., Huang, J.Z.: Learning distributed word representation with multi-contextual mixed embedding. KBS'16 106, 220–230 (2016)
14. Liu, Y., Liu, Z., Chua, T., Sun, M.: Topical word embeddings. In: AAAI'15. pp. 2418–2424 (2015)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS'13. pp. 3111–3119 (2013)
17. Mnih, A., Hinton, G.E.: A scalable hierarchical distributed language model. In: NIPS'08. pp. 1081–1088 (2008)
18. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: NIPS'13. pp. 2265–2273 (2013)
19. Murphy, B., Talukdar, P.P., Mitchell, T.M.: Learning effective and interpretable semantic models using non-negative sparse embedding. In: COLING'12. pp. 1933–1950 (2012)
20. Nam, J., Loza Mencía, E., Fürnkranz, J.: All-in text: Learning document, label, and word representations jointly. In: AAAI'16. pp. 1948–1954 (2016)
21. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP'14. pp. 1532–1543 (2014)
22. Robertson, S.E., Zaragoza, H., Taylor, M.J.: Simple BM25 extension to multiple weighted fields. In: CIKM'04. pp. 42–49 (2004)
23. Rothe, S., Schütze, H.: Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In: ACL'15. pp. 1793–1803 (2015)
24. Schwartz, R., Reichart, R., Rappoport, A.: Symmetric pattern based word embeddings for improved word similarity prediction. In: CoNLL'2015. pp. 258–267 (2015)
25. Sun, F., Guo, J., Lan, Y., Xu, J., Cheng, X.: Learning word representations by jointly modeling syntagmatic and paradigmatic relations. In: ACL'15. pp. 136–145 (2015)
26. Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. JAIR'10 37, 141–188 (2010)
27. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: KDD'02. pp. 639–644 (2002)