Task-oriented Word Embedding for Text Classification

Qian Liu[1,2,3], **Heyan Huang**[1,2]*, **Yang Gao**[1,2], **Xiaochi Wei**[1,2], **Yuxin Tian**[1,2], **Luyang Liu**[1,2]
1. Department of Computer Science, Beijing Institute of Technology, China
2. Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, China
3. Centre for Artificial Intelligence, University of Technology Sydney, Australia

Abstract

Distributed word representation plays a pivotal role in various natural language processing tasks. In spite of its success, most existing methods only consider contextual information, which is suboptimal when used in various tasks due to a lack of task-specific features. The rational word embeddings should have the ability to capture both the semantic features and task-specific features of words. In this paper, we propose a task-oriented word embedding method and apply it to the text classification task. With the function-aware component, our method regularizes the distribution of words to enable the embedding space to have a clear classification boundary. We evaluate our method using five text classification datasets. The experiment results show that our method significantly outperforms the state-of-the-art methods.

1 Introduction

Learning word representation is a fundamental step in various natural language processing tasks. Tremendous advances have been made by distributed representations (also known as word embeddings) which learn a transformation of each word from raw text data to a dense, lower-dimensional vector space. Most existing methods leverage contextual information from the corpus (Mikolov et al., 2013; Pennington et al., 2014) and other complementary information, such as subword information (Cao and Lu, 2017), implicitly syntactic dependencies (Shen et al., 2018a; Shen et al., 2018b), and semantic relations (Bollegala et al., 2016; Liu et al., 2018).

In traditional evaluations such as word similarity and word analogy, the aforementioned context-aware word embeddings work well since semantic information plays a vital role in these tasks, and this information is naturally addressed by word contexts. However, in real-world applications, such as text classification and information retrieval, word contexts alone are insufficient to achieve success in the absence of task-specific features. Figure 1 illustrates this problem with the classification task as an example. Several sentences from different categories are given at the far left of the figure where the words in bold are salient words for the category distinction. We also illustrated expected word distribution of these salient words in the embedding space. To obtain a good classification performance, the expected word distribution should have a clear classification boundary: words within the same category are close to each other and far away from words in other categories as illustrated in Figure 1. However, the actual distribution obtained from Word2Vec at the far right of Figure 1 is normally not satisfactory because Word2Vec only focuses on context similarity. For example, although *learning* and *educational* are with similar context as recognized by Word2Vec, they are salient words to distinguish categories of *AI* and *Sociology*, so they should be far away from each other. Apparently, using word embedding directly from Word2Vec would not obtain good performance on the text classification task due to the fact that words' functional features in the real tasks are ignored in the training process.

In this paper, we propose a task-oriented word embedding method (denoted as ToWE) to solve the aforementioned problem. It learns the distributed representation of words according to the given specific

*Corresponding author

Proceedings of the 27th International Conference on Computational Linguistics, pages 2023–2032
Santa Fe, New Mexico, USA, August 20-26, 2018.

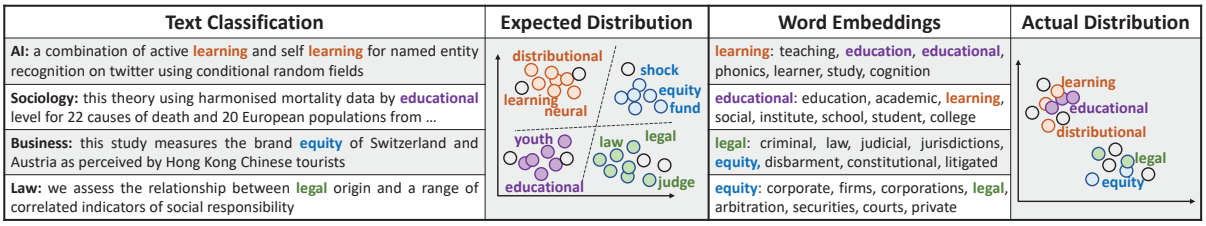


Figure 1: The example sentences from the text classification dataset. Words in bold are salient words to distinguish the sentence category. Their most similar words in the Word2Vec space are shown in the right-hand column. The word color indicates the category, and the words in black are general words for the task.

NLP task. Specifically, we focus on text classification. In our method, the words' contextual information and task information are inherently jointed to construct the word embeddings. In the joint learning framework, the contextual information is captured following the context prediction task introduced by (Mikolov et al., 2013). To model the task information, we regularize the distribution of the salient words to have a clear classification boundary, and then adjust the distribution of the other words in the embedding space correspondingly. To give an intuitive understanding on how our method works from the classification perspective, we design a 5AbstractsGroup dataset (detailed in Section 4.1) and conduct a qualitative analysis. Experiments show qualitative improvements of our method over context-based Skip-gram method on word neighbors for classification. We also perform empirical comparisons on five text classification datasets, which demonstrate the effectiveness of our method over the other state-of-the-art methods.

The contributions of this paper can be summarized as following:

- We propose a task-oriented word embedding method that is specially designed for text classification. It introduces the function-aware component and highlights word's functional attributes in the embedding space by regularizing the distribution of words to have a clear classification boundary.
- We design a 5AbstractsGroup dataset and present a qualitative analysis, giving an intuitive understanding on how our method works from the classification perspective. Experimental results on five text classification datasets also show that the proposed method is more optimal for classification on account of revealing functional attributes of words.

2 Related Work

Word embeddings that provide continuous low-dimensional vector representations of words have been widely studied by NLP communities (Yu et al., 2017; Liu et al., 2017; Li et al., 2017b; Chih et al., 2017). The last few years have seen the development of word embedding methods purely based on the co-occurrence information in a corpus (Bengio et al., 2003; Mnih and Hinton, 2008; Collobert et al., 2011; Mikolov et al., 2013; Mnih and Kavukcuoglu, 2013; Lebret and Collobert, 2014; Pennington et al., 2014; Cao and Lu, 2017; Bollegala et al., 2018). Some studies also pay attention to the semantic knowledge stored in the knowledge bases (Nie et al., 2015). For example, Faruqui et al. (2015) refine word representations using relational information from semantic lexicons, Liu et al. (2015b) represent semantic knowledge as a number of ordinal similarity inequalities of related word pairs to learn semantic word embeddings.

Recent works have thrown light on the problems associated with directly applying word embeddings into real-world applications. Diaz et al. (2016) demonstrated that the globally trained word embedding underperform corpus and query-specific embeddings for retrieval tasks. They proposed locally training word embeddings in a query-specific manner for the query expansion task. Zamani and Croft (2017) indicated that the underlying assumption in typical word embedding methods is not equal to the need of IR tasks, and they proposed relevance-based models to learn word representations based on query-document relevance information, which is the primary objective of most IR task. For the sentiment

analysis task, Yu et al.(2017) refined word embedding to avoid generating similar vector representations for sentimentally opposite words. For the contradiction detection task, Li et al. (2017a) developed contradiction-specific word embedding to recognize contradiction relations between a pair of sentences. These studies show that general trained word embeddings cannot be optimized for a specific task, thus, they are likely to be suboptimal. To meet the needs of real-world applications, rational word embeddings should have the ability to capture both the semantics of words and the task-specific features of words.

In this work, we focus on task-oriented word embedding for the text classification task. Several attempts have shown that revised word embeddings can boost the performance of classification. For example, topical information is shown to be effective in generating high quality word embeddings (Liu et al., 2015c; Liu et al., 2015a), which can enhance the performance of text classification. On the other hand, to enhance the performance of sentiment classification, Chih et al.(2017) proposed a word embedding refinement model to refine existing semantically oriented word vectors using sentiment lexicons in order to distinguish words with similar vector representations but opposite sentiment polarities. Our work departs from previous work in that it directly models task-specific features to construct the embedding space with a clear boundary for classification.

3 Method

Given the unlabeled corpus C and the labeled training set $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_g\}$ with g text categories, our method aims to train the task-oriented d-dimensional word embedding $\mathbf{w_i} \in \mathcal{R}^d$ for the i-th word w_i in vocabulary $\mathcal{V}$. Formally, the document collection belonging to the k-th category is denoted as $\mathcal{D}_k$. Our proposed joint learning framework contains two components, i.e., the context-aware component and function-aware component. The context-aware part models the co-occurrence in corpus C and captures the word semantic features. The function-aware part reveals the word's functional attributes following the task-specific features observed in $\mathcal{D}$. We next describe these two parts respectively.

3.1 Context-aware Component

Our method uses the Word2Vec method to model the context information and uses log-linear models to produce word embeddings. It applies a sliding window moving on the corpus. The word in the center of the window is the target word and the others are context words. Word2Vec has two versions, i.e., CBOW and Skip-gram. The CBOW model uses the average/sum of context words as input to predict the target, and the Skip-gram model uses the target word as input to predict each context word. To simplify, we represent the objective of each prediction as

$$\mathcal{L}_{context} = Pr(w|\mathbf{c}) = \frac{\exp(\mathbf{w} \cdot \mathbf{c})}{\sum_{w' \in \mathcal{V}} \exp(\mathbf{w'} \cdot \mathbf{c})}. \tag{1}$$

In CBOW, w is the target word, and $\mathbf{c}$ is the vector of the context words, and in Skip-gram, w is each word in the context, and $\mathbf{c}$ is the vector of the target word.

3.2 Function-aware Component

In the function-aware component, we define salient words as those words with the ability to distinguish the document category. These salient words are first extracted from the labeled training set $\mathcal{D}$ in an offline process. Then the correlations among these words are used to model the functional features in the embedding space.

Each salient word w of the k-th category is offline extracted according to the following the two principles: (1) The term frequency of the word w in this category (i.e., $\mathcal{D}_k$) is much higher than that in other categories; (2) w is common in other categories, expressed as a small variance of term frequencies in other categories. Formally, we design the following formula to measure the importance of word w to the k-th category as a salient word:

$$Score(w, k) = \frac{t_k - \frac{1}{g}\sum_{1 \le i \le g} t_i}{var(T_{-k}(w))}, \tag{2}$$

where t_i is the term frequency in the i-th category, $T_{-k}(w)$ is the collection of term frequencies except the k-th category (i.e., $T_{-k}(w) = \{t_j | 1 \leq j \leq g, j \neq k\}$), and $var(\cdot)$ is the variance.

According to this importance score, we generate a salience words set by selecting the top N words for each category, denoted as $S_k = \{w_j | 1 \leq j \leq N\}, k \in [1, g]$. Then, for the task, the words in the salient words set $S = \{S_1, S_2, \cdots, S_g\}$ have the ability to distinguish different categories.

The salient words are next utilized to capture the functional relations between words in the embedding space. In the learning framework, if the predicted word w is in S, the function-aware component will be activated. As to modeling the correlations of function-salient words, we expect to constrain w to be close to the words in the same category and far away from the words in different categories. According to this idea, we construct a set $P(w)$ with n word-pairs for each salient word w. Each word-pair contains a positive word u and a negative word v. The positive words are randomly selected from S which belong to the same category with w, and the negative words are randomly sampled from other categories. We maximize a margin-based ranking criterion over the training set S:

$$\mathcal{L}_{function} = argmax \sum_{<u,v> \in P(w)}^{n} [\gamma + s(\mathbf{w}, \mathbf{u}) - s(\mathbf{w}, \mathbf{v})], \tag{3}$$

where γ is a margin hyper parameter, n is the size of sample set $P(w)$, and $s(\cdot, \cdot)$ is similarity measure. Following the recommendations in prior work on word similarity measurement, we apply the cosine similarity of a pair of words by computing $s(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| \cdot |\mathbf{b}|}$. The objective function favors higher values of the similarity for positive word-pairs than for negative word-pairs, and is thus a natural implementation of the intended criterion.

3.3 Joint Learning

The context-aware component and the function-aware component are jointly optimized, so we then obtain the following object function:

$$\mathcal{L} = \underset{\Theta}{argmax} \quad \lambda \mathcal{L}_{context} + (1 - \lambda) \mathcal{L}_{function}, \tag{4}$$

where Θ is a set of all parameters in $\mathcal{L}_{context}$ and $\mathcal{L}_{function}$, and λ is the combination parameter which balances the contribution of each component in the training process.

The goal of the training objective is to maximize $\mathcal{L}$ with respect to the model parameters. The optimization process is conducted via Stochastic Gradient Descent (SGD). The optimization of $\mathcal{L}_{context}$ follows the negative sampling introduced in (Mikolov et al., 2013). If the predicted word w is in the salient words set S, the corresponding optimization process for $\mathcal{L}_{function}$ will be activated, and the parameters are updated as $\mathbf{w} \leftarrow \mathbf{w} + \eta \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$, $\mathbf{u} \leftarrow \mathbf{u} + \eta \frac{\partial \mathcal{L}}{\partial \mathbf{u}}$, and $\mathbf{v} \leftarrow \mathbf{v} + \eta \frac{\partial \mathcal{L}}{\partial \mathbf{v}}$, where η is the learning rate, and the gradients are calculated as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{w}} &= \lambda \sum_{<u,v> \in P(w)}^{n} \frac{\partial s(\mathbf{w}, \mathbf{u})}{\partial \mathbf{w}} - \frac{\partial s(\mathbf{w}, \mathbf{v})}{\partial \mathbf{w}}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{u}} &= \lambda \sum_{<u,v> \in P(w)}^{n} \frac{\partial s(\mathbf{w}, \mathbf{u})}{\partial \mathbf{u}}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{v}} &= \lambda \sum_{<u,v> \in P(w)}^{n} - \frac{\partial s(\mathbf{w}, \mathbf{v})}{\partial \mathbf{v}},
\end{aligned} \tag{5}$$

where w is the predicted word, u is its positive word and v is its negative word. Since we apply cosine distance to compute the similarity between two words, the optimization can be derived as follows:

$$\frac{\partial s(\mathbf{a}, \mathbf{b})}{\partial \mathbf{a}} = -\frac{S_{a,b} \cdot \mathbf{a}}{|\mathbf{a}|^2} + \frac{\mathbf{b}}{|\mathbf{a}| \cdot |\mathbf{b}|}, \tag{6}$$

Algorithm 1 Task-oriented Word Embedding Method.

Input: Corpus C, the labeled training set $\mathcal{D}$ with g categories, dimensionality d, sampling times n, and word vocabulary $\mathcal{V}$

Output: Embeddings $\mathbf{w} \in \mathcal{R}^d$ of all words in the vocabulary $\mathcal{V}$.

Initialization: randomly set $\mathbf{w} \in \mathcal{R}^d$ for all words in $\mathcal{V}$; generate the salient words set S; constructing T prediction tasks using a sliding window.

for t = 1, 2, . . . , T **do**
  optimize $\mathcal{L}_{context}$ using negative sample method introduced in (Mikolov et al., 2013)
  if w in S **then**
    for n **do**
      sampling the positive word u and the negative word v.
      optimize $\mathcal{L}_{function}$ using Eq.(5) to update $\mathbf{w}$, $\mathbf{u}$, $\mathbf{v}$.
    end for
  end if
end for
return $\mathbf{w}$ for all words in $\mathcal{V}$.

where $S_{a,b} = \frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}|\cdot|\mathbf{b}|}$. The pseudo code for our word embedding method is shown in Algorithm 1, and the source code is available on the Github[1].

4 Experiments

4.1 Datasets

To undertake an extensive evaluation, we investigate the empirical performances of our proposed method on five text classification datasets. The detailed statistics of all the datasets are listed in Table 1. Each dataset is briefly described as follows:

Datasets	Type	Train Size	Test Size	#Classes	Avg.L	Vocab Size	#Tokens
20NewsGroup	Doc.	11,314	7,532	20	315	179,092	6,555,230
5AbstractsGroup	Doc.	2,500	3,756	5	223	38,103	1,203,022
IMDB	Doc.	25,000	25,000	2	126	170,543	6,141,136
MR	Sen.	32,361	32,359	2	21	47,568	974,626
SST	Sen.	5,928	5,927	2	12	19,362	152,474

Table 1: Statistics of the five mainstream datasets for text classification.

(1) The **20NewsGroup**[2] is a popular text classification dataset which contains 18,846 documents from 20 different newsgroups. Each document contains several sentences. The dataset is separated into a training set of 11,314 documents and a test set of 7,532 documents. (2) The **5AbstractsGroup** dataset is academic papers from five different domains collected from the Web of Science namely, business, artificial intelligence, sociology, transport and law. We extracted the abstract and title fields of each paper as a document. The dataset contains 6,256 documents, and we randomly selected 500 papers in each category as the training set, and the others as the test set. The dataset is published on the Github[3]. (3) The **IMDB**[4] contains movie reviews with binary classes (i.e., positive and negative). It consists of 50,000 movie reviews (Maas et al., 2011), and each movie review has several sentences. (4) The **MR**[5] dataset consists of movie reviews from Rotten Tomato website with two classes labeled by (Pang and Lee, 2005). Each review contains only one sentence. (5) The **SST**[6] dataset contains the movie reviews

[1] https://github.com/qianliu0708/ToWE
[2] http://qwone.com/ jason/20Newsgroups/.
[3] https://github.com/qianliu0708/5AbstractsGroup
[4] https://www.cs.jhu.edu/mdredze/datasets/sentiment/unprocessed.tar.gz
[5] https://www.cs.cornell.edu/people/pabo/movie-review-data/
[6] http://nlp.stanford.edu/sentiment

Business	AI	Law	Sociology	Transport
employee	distributional	jurisdiction	educational	driver
entrepreneur	predefine	interpreted	sport	departure
stakeholder	inference	law	food	urban
trait	recognition	dispute	farmer	intersection
consumer	variant	qualified	sociology	accident
marketplace	analytic	congress	experience	incident
asset	learn	interfere	poverty	route
bond	aggregate	contract	religious	transferring
manager	object	victim	youth	passenger
markets	uncertain	permit	ethnicity	vehicle

Figure 2: Top ten salient words for each category in the 5AbstractsGroup dataset.

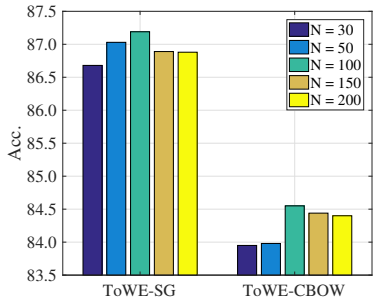
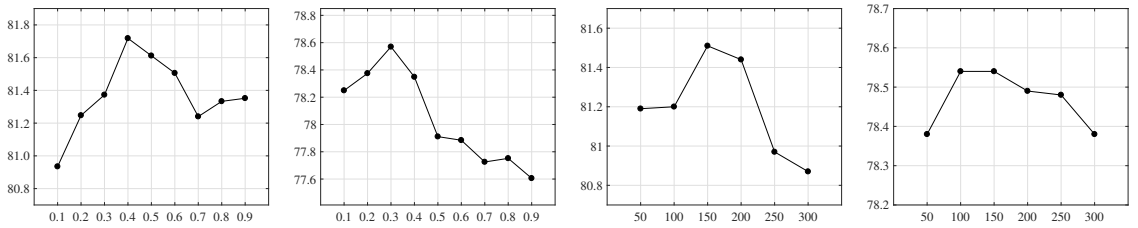


Figure 3: Performance of the ToWE method with varying N on the 5AbstractsGroup dataset.



(a) λ on the ToWE-SG (b) λ on the ToWE-CBOW (c) n on the ToWE-SG (d) n on the ToWE-CBOW

Figure 4: Performance of the ToWE method with the varying parameter λ and the size of sampling n. The Y-axis represents the accuracy (%) on the 20NewsGroup dataset.

in the Stanford Sentiment Treebank labeled by (Socher et al., 2013) comprising one sentence for each review. 50% of the MR and SST datasets are partitioned randomly into the training set and 50% into the test set.

4.2 Baseline Methods

To evaluate our method, we consider the following baselines: (1) the **BOW** method is employed as a basic baseline. It represents each document as a bag of words and the weighting scheme is TFIDF. We select the top 2,000 words according to the TFIDF scores as features; (2) the Word2Vec method is a neural network language method which learns word embeddings by maximizing the conditional probability leveraging contextual information. It comprises two models, i.e., **CBOW** which predicts the target word using context information, and the Skip-gram (denoted as **SG**) which predicts each context word using the target word; (3) the **GloVe** (Pennington et al., 2014) method is a state-of-the-art matrix factorization method. It leverages global count information aggregated from the entire corpus as word-word occurrence matrix to learn word embeddings; (4) the Topical Word Embedding method (denoted as **TWE**) (Liu et al., 2015c) learns a topic model from the training set, then generates word embeddings by jointly considering words and topics in a neural network; (5) the **Retrofit** method (Faruqui et al., 2015) is a popular method that refines pre-trained word embeddings using relational information from the knowledge base (e.g., WordNet used in our experiments).

4.3 Experimental Settings

In this paper, we use the text classification task to evaluate the performance of word embeddings. Word embeddings are used to construct the document embeddings $\mathbf{d}$ by simply averaging all word embeddings in the given document, i.e., $\mathbf{d} = \frac{1}{|d|}\sum_{w \in d} \mathbf{w}$, where w is a word in document d. We regard document embedding as a document feature and trained a linear classifier using Liblinear[7] (Fan et al., 2008), since the feature size is large, and Liblinear can quickly train a linear classifier with high dimension features. The classifier is then used to predict the class labels of documents in the test set. The multi-group

[7]https://www.csie.ntu.edu.tw/ cjlin/liblinear/.

Methods	20NewsGroup				5AbstractsGroup				IMDB	MR	SST
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.	Acc.	Acc.
BOW	73.6	73.6	72.8	73.0	77.1	76.6	77.2	76.5	85.3	59.3	73.4
GloVe	62.3	61.2	61.1	60.5	79.6	78.4	79.4	79.4	87.4	58.7	75.5
CBOW	74.5	73.6	73.5	73.4	79.4	78.6	78.8	78.8	87.1	61.8	77.9
SG	76.7	75.9	75.6	75.4	85.2	84.0	85.0	84.4	89.1	63.5	77.3
TWE	81.5	81.2	80.6	80.6	81.5	80.5	81.2	80.7	87.1	56.0	76.9
Retrofit-CBOW	75.6	75.9	73.5	72.1	78.2	77.4	77.6	77.3	86.6	61.9	78.0
Retrofit-SG	77.4	77.9	75.5	74.3	83.3	82.3	83.0	82.6	88.8	63.7	77.9
ToWE-CBOW	80.9	80.2	79.9	79.9	84.7	84.0	84.4	84.4	90.1	64.5	**78.8**
ToWE-SG	**86.0**	**85.5**	**85.0**	**85.0**	**87.2**	**86.2**	**87.1**	**87.1**	**90.8**	**65.1**	78.4

Table 2: Performance of our methods on five datasets against the baselines. Bold scores are the best overall.

classification performance was evaluated in terms of four measures: accuracy (Acc.), precision (Prec.), recall (Rec.) and F-measure (F1), and the binary classification performance was evaluated by accuracy (Acc.). All the measures are computed by averaging the metrics of each class and are weighted by the number of true instances for each class.

For each dataset, all documents are joined together as a corpus for embedding training. We tokenized the corpus with the Stanford Tokenizer[8] and converted it to lower case, then removed the stop words. For a fair comparison, all word embeddings adhere to the following settings: the dimensionality of vectors is 300, the size of the context window is 5, the number of negative samples is 25.

In our method, an offline process is used to extract salient word set S from labeled training set $\mathcal{D}$. To obtain an intuitive understanding of these salient words, we list the top ten words for each category in the 5AbstractsGroup dataset. The result is displayed in Table 2. We vary parameter N (detailed in section 3.2) in the range between 30 and 200, and show the performance in Figure 3. Our method achieves the best performance when N is set to 150 for the 5AbstractsGroup dataset. If the value of N is too large, this may hinder the performance because too much noise will be involved. The recommended N is 150 with the constraint that the total size of S is under 1200 based on practical experience.

There are two hyper-parameters in our method, i.e., the combination parameter λ in Eq.(4) and the size n of sample set $P(w)$ in Eq.(3). We carefully tune these parameters by fixing one and varying the other. The parameters corresponding to the best accuracy in 20NewsGroup are used to report the final settings. As shown in Figure 4, the optimal values for λ were tuned from 0 to 1, with a step size of 0.1. The proposed method based on Skip-gram and CBOW reaches optimal performance when $\lambda = 0.4$ and $\lambda = 0.3$, respectively. We tuned the value for n from 50 to 300, and the methods achieve the best performance when $n = 150$. We follow the optimal settings in this work, with recommended settings of $\lambda \in (0.3, 0.4)$ and $n \in (100, 150)$.

4.4 Overall Performance

We compared our proposed method with the baseline methods. Table 2 shows the evaluation results. Based on the experiment results, we make several observations:

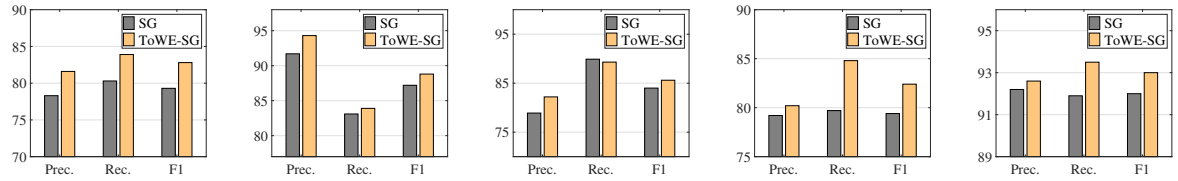
(1) Our method performs better than the other methods, and are proved to be highly reliable for the text classification task. In particular, the ToWE-SG method significantly outperforms the other baselines on the 20NewsGroup, 5AbstractsGroup, and MR. This is mainly attributed to the task-specific modeling mechanism, which enables our models to capture functional features among words, therefore, it can more accurately distinguish classes.

(2) The word embedding methods outperform the basic bag-of-words methods in most cases, indicating the superiority of distributed word representation over the one-hot representation. Moreover, the

[8]https://nlp.stanford.edu/software/tokenizer.shtml

manager (*Business*)		**layer** (*AI*)		**congress** (*Law*)		**poverty** (*Sociology*)		**accident** (*Transport*)	
ToWE-SG	SG	ToWE-SG	SG	ToWE-SG	SG	ToWE-SG	SG	ToWE-SG	SG
managerial	innovate	appearance	form	federal	chapter	deprivations	urbanization	accidents	crash
extant	pursuing	recurrent	forgetting	permit	authorized	belonging	projections	drivers	severity
executives	incentives	architecture	symbolic	administrative	secrecy	homeless	urbanization	severity	injury
stakeholder	subfield	automatic	space	enforcement	prohibiting	inequality	auto	red	rtc
investors	accord	collecting	encoding	earned	bureaucrats	affordability	commuting	mobility	crashes
bond	strategically	cognitive	involves	regulating	dockets	malnutrition	deforestation	road	fatal
moderates	helps	proposed	structures	exception	wrongful	adulthood	anthropogenic	elasticity	rollover
innovation	strategic	learning	polarization	submitted	defense	ethnicity	co-benefits	safety	single-vehicle
marketing	tailor	algorithms	activation	regulate	hear	religious	modal	estimated	taz
asymmetry	create	neural	discontinuous	defense	he	discursive	ownership	delay	crash-related

Table 3: Ten most similar words to the salient words using the ToWE-SG method and SG method. The bold words are salient words, and their category is marked in italic.



(a) Business (b) AI (c) Law (d) Sociology (e) Transport

Figure 5: Performance of SG and ToWE-SG methods for each category in the 5AbstractsGroup dataset.

methods which integrate the abundant information discovered from the datasets (i.e., TWE and ToWE) achieve better performance compared to those that only consider contextual information, such as GloVe, CBOW, and SG. This demonstrates the effectiveness of refining context-aware word embeddings with task information.

(3) The Retrofit method is the knowledge-base enhanced word embedding method. Our method achieves better performance over Retrofit method, indicating that the task-specific features could be more effective compared with general semantic relations constructed by humans in the knowledge bases.

(4) In sentence classification, such as the MR and SST datasets, it is obvious that TWE achieves a relatively lower performance. This observation shows that topical information enhanced word embedding does not accurately represent a short text. Our method outperforms the TWE method on both the document-level and sentence-level tasks, which shows the stability and reliability of modeling task-specific features in real-world applications.

4.5 Case Study

A case study was conducted to qualitatively analyze in-depth why task-oriented word embedding methods surpass typical context-aware word embedding methods. We selected several salient words from different categories in the 5AbstractsGroup dataset, and then compared the top ten similar words obtained by ToWE-SG and SG, respectively. The results are displayed in Table 3. We observe that the similar words selected by the ToWE-SG method belong to the same category, while the SG method may select words from different categories. Taking the word *manager* as an example, the most similar words selected by ToWE-SG all belong to the *Business* category, whereas the SG method selects *helps*, *create* which can hardly be regarded as being in the *Business* category. This demonstrates that our method is capable of capturing a clear boundary in the embedding space. For further investigation, we compared the classification performance of these two word embeddings in each category. As shown in Figure 5, ToWE-SG outperforms SG in all these categories. This indicates that by forcing words in the same category to have similar representations, the classifier achieves better performance.

5 Conclusion

In this paper, we proposed a novel approach for learning task-oriented word embedding, especially for the text classification task. Instead of learning embedding vectors merely based on context information, we incorporate task-specific features into the training process in order to reveal the words functional

attributes in the embedding space. The results of the experiments with different datasets show that the proposed method outperforms the existing state-of-the-art word embedding learning methods on text classification tasks. In the future, we will study how to effectively construct the task-oriented word embeddings with the help of transferable task-features across domains.

6 Acknowledgment

The research work is supported by the National Key Research and Development Program of China under Grant No.2017YFB0803302, National Natural Science Foundation of China (Key Program) under Grant No.61751201 and National Nature Science Foundation of China under Grant No.61602036.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Danushka Bollegala, Mohammed Alsuhaibani, Takanori Maehara, and Ken-ichi Kawarabayashi. 2016. Joint word representation learning using a corpus and a semantic lexicon. In *Proceedings of AAAI*, pages 2690–2696.

Danushka Bollegala, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2018. Using k-way co-occurrences for learning word embeddings. In *Proceedings of AAAI*, pages 5038–5044.

Shaosheng Cao and Wei Lu. 2017. Improving word embeddings with convolutional feature learning and subword information. In *Proceedings of AAAI*, pages 3144–3151.

Yu Liang Chih, Wang Jin, Lai K. Robert, and Xuejie Zhang. 2017. Refining word embeddings for sentiment analysis. In *Processings of EMNLP*, pages 534–539.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query expansion with locally-trained word embeddings. In *Proceedings of ACL*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard H. Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL-HLT*, pages 1606–1615.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through hellinger PCA. In *Proceedings of EACL*, pages 482–490.

Luyang Li, Bing Qin, and Ting Liu. 2017a. Contradiction detection with contradiction-specific word embedding. *Algorithms*, 10(2):59.

Zheng Li, Yu Zhang, Ying Wei, Yuxiang Wu, and Qiang Yang. 2017b. End-to-end adversarial memory network for cross-domain sentiment classification. In *Proceedings of IJCAI*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2015a. Learning context-sensitive word embeddings with neural tensor skip-gram model. In *Proceedings of IJCAI*, pages 1284–1290.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015b. Learning semantic word embeddings based on ordinal knowledge constraints. In *Proceedings of ACL*, pages 1501–1511.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015c. Topical word embeddings. In *Proceedings of AAAI*, pages 2418–2424.

Qian Liu, Heyan Huang, Jie Lu, Yang Gao, and Guangquan Zhang. 2017. Enhanced word embedding similarity measures using fuzzy rules for query expansion. In *Proceedings of FUZZ-IEEE*, pages 1–6.

Qian Liu, Heyan Huang, Guangquan Zhang, Yang Gao, Junyu Xuan, and Jie Lu. 2018. Semantic structure-based word embedding by incorporating concept convergence and word divergence. In *Proceedings of AAAI*, pages 5261–5268.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of ACL*, pages 142–150.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Andriy Mnih and Geoffrey E. Hinton. 2008. A scalable hierarchical distributed language model. In *Proceedings of NIPS*, pages 1081–1088.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Proceedings of NIPS*, pages 2265–2273.

Liqiang Nie, Yi-Liang Zhao, Mohammad Akbari, Jialie Shen, and Tat-Seng Chua. 2015. Bridging the vocabulary gap between health seekers and healthcare knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):396–409.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.

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

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018a. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of AAAI*.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. 2018b. Bi-directional block self-attention for fast and memory-efficient sequence modeling. In *Proceedings of ICLR*.

R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Liang-Chih Yu, Jin Wang, K. Robert Lai, and Xue-Jie Zhang. 2017. Refining word embeddings for sentiment analysis. In *Proceedings of EMNLP*, pages 534–539.

Hamed Zamani and W. Bruce Croft. 2017. Relevance-based word embedding. In *Proceedings of SIGIR*, pages 505–514.