Neural Computing and Applications
https://doi.org/10.1007/s00521-019-04071-6

A distant supervision method based on paradigmatic relations for learning word embeddings

Jianquan Li[1] · Renfen Hu[2] · Xiaokang Liu[1] · Prayag Tiwari[4] · Hari Mohan Pandey[3] · Wei Chen[1] · Benyou Wang[4] · Yaohong Jin[1] · Kaicheng Yang[1]

Received: 15 October 2018 / Accepted: 30 January 2019

Abstract

AQ1 Word embeddings learned on external resources have succeeded in improving many NLP tasks. However, existing embedding models still face challenges in situations where fine-gained semantic information is required, e.g., distinguishing antonyms from synonyms. In this paper, a distant supervision method is proposed to guide the training process by introducing semantic knowledge in a thesaurus. Specifically, the proposed model shortens the distance between target word and its synonyms by controlling the movements of them in both unidirectional and bidirectional, yielding three different models, namely *Unidirectional Movement of Target Model* (UMT), *Unidirectional Movement of Synonyms Model* (UMS) and *Bidirectional Movement of Target and Synonyms Model* (BMTS). Extensive computational experiments have been conducted, and results are collected for analysis purpose. The results show that the proposed models not only efficiently capture semantic information of antonyms but also achieve significant improvements in both intrinsic and extrinsic evaluation tasks. To validate the performance of the proposed models (UMT, UMS and BMTS), results are compared against well-known models, namely *Skip-gram*, *JointRCM*, *WE-TD* and *dict2vec*. The performances of the proposed models are evaluated on four tasks (benchmarks): *word analogy* (intrinsic), *synonym-antonym detection* (intrinsic), *sentence matching* (extrinsic) and *text classification* (extrinsic). A case study is provided to illustrate the working of the proposed models in an effective manner. Overall, a distant supervision method based on paradigmatic relations is proposed for learning word embeddings and it outperformed when compared against other existing models. AQ2

Keywords Neural network · Word embedding · Text classification · Sentence matching

1 Introduction

Natural language processing (NLP) is one of the key concerns of artificial intelligence (AI) and machine learning (ML) techniques. NLP can be used in real-life applications such as search engine, personal assistant and online shopping and other. These applications are related to the basic NLP tasks, e.g., word-level understanding, text classification, text matching. In case of text classification, task targets classify a sentence into a specific pre-defined label while the matching task targets distinguish the relation between two sentences. These days most of the works depends on a distributed word-level presentation known as word embedding [20, 21, 26] as their input features and it achieves a great success in several typical NLP tasks. However, it meets its bottleneck in performance since these word presentations only make use of word-level co-occurrence information from external corpus but with little common sense from linguistics. It is argued in this paper that the data-driven word presentation should also incorporate some common linguistic knowledge, e.g., linguistic relations between words. Culler [6] introduces two fundamental types of relations between words: syntagmatic relation and paradigmatic relation [14]. Syntagmatic relation describes the linear relation of words in a sequence and focuses on the co-occurrence information. The typical

✉ Benyou Wang
wang@dei.unipd.it

1 Beijing Ultrapower Software Co., Ltd, Beijing, China

2 Beijing Normal University, Beijing, China

3 Department of Computer Science, Edge Hill University, Ormskirk, UK

4 Department of Information Engineering, University of Padova, Padua, Italy

Target: Little boy with bright blue eyes smiling.
Wrong: The boy with brown eyes is unhappy.
Correct: The boy eyes are a bright blue color and he is happy.

Fig. 1 Sentence matching task on the sentence "little boy with bright blue eyes smiling."

examples are word pair's such as *beef–eat*, *snow–cold* or *doctor–hospital*. Paradigmatic relation exists between words which can be substituted by one another such as synonyms *beautiful–pretty*, antonyms *up–down* and AQ3 hypernyms *fruit–apple*.

Recently, syntagmatic associations have been successfully applied to word embedding models, e.g., *word2vec* [19], which exploits the *Context Words to Predict Target Words* (CBOW) or the target words to predict context words (Skip-gram). By assuming words occurring in similar contexts tend to have similar meanings [12], word2vec attempts to capture paradigmatic relations between words with the help of syntagmatic relations. This method achieves great performance in word representations, and the pre-trained embeddings have been widely used as inputs for downstream tasks, e.g., text classification and machine translation.

Challenges However, as synonymous and antonymous words can both hold paradigmatic relations, i.e., they can be replaced by each other without affecting the grammaticality or acceptability of a sentence. As a result, antonyms become very close in the vector space as well as synonyms. It would be a serious problem for tasks that rely on word similarity information. Figure 1 shows a sentence matching example in which most embedding-based methods choose the wrong sentence as the close stone since the models cannot efficiently distinguish between antonyms, e.g., *happy* and *unhappy*.

Existing solution (s) To solve this problem, several approaches have been proposed to construct word embeddings that can capture antonyms [4, 23, 25]. However, as these methods are built specifically for detecting antonyms and they have ignored the fact that two antonymous words are still relevant and belong to the same category, e.g., *up* and *down* are both describing directions. By minimizing similarities between antonyms, these methods [4, 23, 25] are potential to destroy the global semantic distribution. Even though they have achieved surprisingly good results in antonym detection, their performance in other evaluation criteria's such as word analogy and semantic matching is much less than desirable.

Our contributions Based on the above observation, this paper proposes a novel yet effective method to learn improved word embeddings with distant supervision. We have made the following key contributions:

- A thesaurus *Para-Phrase Database* (PPDB) [11] is introduced to enrich semantic information of word representations based on paradigmatic relations. Unlike previous works that simply integrate synonyms as contexts [32], which inappropriately equate the syntagmatic relation and paradigmatic relation, our method shortens the distance between target word and its synonyms by controlling their movements in both unidirectional and bidirectional ways yielding three different models: *Unidirectional Movement of Target Model* (UMT), *Unidirectional Movement of Synonyms Model* (UMS) and *Bidirectional Movement of Target and Synonyms Model* (BMTS).
- We have presented a fresh discussion of related work on learning word embedding with the aim to identify research gaps. We highlighted the deficiencies of typical learning models in an organized manner for quick review (see Table 1).
- To develop a deeper understanding, first, we discuss the existing model and then presenting our proposed models for learning word embeddings.
- Extensive computational experiments are conducted to validate the proposed system. The experimental results demonstrate that the proposed learning method not only effectively distinguish between synonyms and antonyms but also optimize the global word vector space.
- We highlighted that all three different models (UMT, UMS and BMTS) achieve considerable improvements in both intrinsic and extrinsic evaluation tasks especially in semantic matching task that emphasizes the global semantic representations.

The rest of the paper is organized as follows: Sect. 2 presents the related work on word embedding. Section 3 discusses the proposed learning word embedding model in detail. The experimental benchmarks, implementation details, evaluation metrics and baseline methods are discussed in Sect. 4. It also presents the experimental results and analysis with a case study to develop a deeper

Table 1 Deficiencies of partial word vector models

Deficiencies	Typical models
Insensitivity to antonyms	[10, 20, 21, 26]
Insensitivity between syntagmatic and paradigmatic relations	[2, 3, 9, 29, 32]
Overemphasis of antonymous	[1, 17, 22, 25]

Journal : Large 521 Dispatch : 4-2-2019 Pages : 11
Article No. : 4071 ☐ LE ☐ TYPESET
MS Code : NCAA-D-18-02469 ☑ CP ☑ DISK

understanding. Lastly, conclusions of this paper are drawn in Sect. 5.

2 Related works on learning word embeddings

Distributional semantic models (DSMs) represent word meanings as vectors. They have a long history that could date back to the 1990s [5, 8, 13]. After [19] proposes the *word2vec* model, a great number of extensions are built based on this influential method [10, 20, 21]. In these works, large unlabeled corpus was used to train the distributed word representations. Pennington et al. [26] presented the *GloVe* model which was based on word co-occurrence statistics. This method [26] combines the advantages of the *global matrix factorization* and *local contexts*. Word embedding also developed into different types; some of them are: *Gaussian Embedding* [30], *Hyperbolic Embedding* [24, 28], *Complex-Valued Embedding* [18] and *Pre-Trained Language Model for Dynamic Embedding,* etc. In particular, [7, 27] boost largely many language models where some sort of pre-trained language models adaptively generates real-time word vector. However, these basic word vector models have utilized the word-level co-occurrence information either implicitly or explicitly; but they did not take some fine-gained between-word relation. For example, they are limited to distinguish between antonyms, which in most of the situation assumed to be very sensitive in some NLP tasks like sentiment analysis. For example, the words "*good*" and "*bad*" have closed vector in general word embedding technology (like Word2vec and Glove) due to that they might appear in a similar context and thus are embed with closed vectors. This could damage more the performance of sentiment analysis, since it is more sensitive to the word polarity.

To improve the word representations, a prominent approach is to introduce external resources into models. Lexical databases like *WordNet* or *FrameNet* [2] can be used during learning or in a post-processing step to specialize word embeddings [9]. Yu and Dredze [32] demonstrated that the *Relation Constrained Model* (RCM) improved the performance of three semantic tasks, namely *Language Modeling*, *Measuring Semantic Similarity* and *Predicting Human Judgements* by incorporating *PPDB* and *WordNet.* Tissier et al. [29] build pairs from dictionary which provides an additional context so that semantically related words can move closer. Bian et al. [3] explored three types of knowledge: *morphological*, *syntactic*, and *semantic* to train high-quality word embeddings. Most of these methods introduce synonyms or definition words from dictionary into the context to enrich semantic representations. However, considering syntagmatic relation and paradigmatic relation are two different types of relations. Context words represent the syntagmatic relations, while synonyms, antonyms and hypernyms represent paradigmatic relations. It might not be suitable to equate the paradigmatic words with the context words.

In order to capture better semantic information of antonyms, Adel and Schutze [1] suggested co-reference chains extracted from large corpora into the Skip-gram model to train word embeddings that could distinguish detect antonyms. Ono et al. [25] proposed two models: *WE-T* and *WE-TD*. The objective functions of these models were, respectively, based on maximizing the similarity between synonyms and minimize the similarity between antonyms. Lazaridou et al. [17] introduced the *multi-task Lexical Contrast Model* (mLCM), which regards the whole semantic space as a polar space to find a max-margin plane. Nguyen et al. [22] integrated the lexical contrast information with the objective of Skip-gram model and improved the quality of weighted features to distinguish antonyms and synonyms. All these efforts had achieved surprisingly good results in specifically the detection of antonyms without considering the general tasks. These methods [1, 17, 22, 25] had ignored the fact that two antonymous words still belong to the same category and are highly relevant. Minimizing similarities of antonyms might result in uncontrollable vector movement and, thus, negatively affect the global semantic distribution.

The aforementioned discussion reveals many deficiencies that are still present in the existing models which are depicted in Table 1. In this paper, we propose a novel approach to improve Skip-gram models with distant supervision. The proposed models utilize distant supervision approach that helps in shortening the distance between target word and its synonyms by controlling their movements in both unidirectional and bidirectional ways. Specifically, the synonym dictionary it built using PPDB with TF-IDF weighting methods. It is claimed in this paper that our word vector method uses both the synonym and antonymous words in a proper way for a general purpose. Here, the term "general purpose" signifies that the proposed models have abilities of not only recognizing synonyms and antonyms but also it has ability to perform general purpose tasks such as downstream tasks (e.g., text classification, text matching, etc.)

We set two optimizations goals during the implementation of the proposed model as depicted below:

(a) Learn semantic and syntactic information from contexts;
(b) Enrich the semantic information by controlling the movements of synonyms.

By achieving these two goals, the proposed models have demonstrated the ability to effectively distinguish

Journal : Large 521 | Dispatch : 4-2-2019 | Pages : 11
Article No. : 4071 | ☐ LE | ☐ TYPESET
MS Code : NCAA-D-18-02469 | ☑ CP | ☑ DISK

antonyms from synonyms and achieved significant improvements in both intrinsic and extrinsic evaluation tasks.

3 The models for learning word embeddings

In this section, first, we shade light on two popular learning word embedding models, namely word2vec and Semantic Lexicons from PPDB. Second, we discuss the proposed learning word embedding model in a comprehensive manner. Finally, we highlighted the role of distant supervision method in our proposed learning word embedding model.

3.1 Word2vec

The Word2vec is the most frequently used method for training word embeddings. Two different types of Word2vec implementation have been suggested, namely *CBOW* and *Skip-gram*. In particular, the Skip-gram model uses a sliding window to select context information. Equation (1) represents the optimization function used for the Skip-gram model.

$$\sum_{t=1}^{C}\sum_{k=0}^{n}\log p(w_{t+k}|w_t) \tag{1}$$

where n, C, w and $p(w_{t+k}|w_t)$, respectively, represents size of window, corpus, the word from corpus and probability of context w_{t+k}. Equation (2) is used to determine the probability $p(w_{t+k}|w_t)$.

$$\begin{aligned}\Pr(w_{i-k},\ldots,w_{i+k}|w_i) &= \prod_{w_c\in C(w_i)}\Pr(w_c|w_i)\\ &= \prod\frac{\exp(w_c^T\cdot w_i)}{\sum_{w_c'\in W}\exp(w_c'^T\cdot w_i)}\end{aligned} \tag{2}$$

In Eq. (2), w_c and w_i, respectively, represents embedding of context word and target word with $w_c \in C(w_i)$. The skip-gram model offers a good balance between efficiency and effectiveness for distributed language model. Therefore, we utilized Skip-gram model framework for the proposed learning word embeddings model.

3.2 Semantic lexicons from PPDB

PPDB is a semantic lexicon database built from bilingual parallel corpora. It includes over 100 million sentence pairs and over 2 billion English words. For the proposed distant supervision-based learning word embeddings model, we utilized synonyms from PPDB to construct the knowledge base. The following observations have been made: "*with the size of lexical paraphrase dataset—increases from S (small) to XXXL (extra-large), the confidence of the lexical dataset shows a continuously decreasing trend*". We have not used antonym in our proposed model mainly because our proposed model considers the phenomenon as: "*the antonyms should not be unconditionally far away from target words*".

3.3 Proposed models

3.3.1 Intuition

Paradigmatic relation exists between words which can be substituted by one another, such as synonyms, antonyms and hypernyms. The proposed distant supervision method introduces paradigmatic relation into Skip-gram model and shortens the distance between target word and its synonyms by controlling their movements in both unidirectional and bidirectional. The synonym data are only used in the model because relations between antonyms are very subtle: *on one side, they belong to the same category and are highly relevant; on the other side, they are describing the opposite meaning*. Thus, the movement of antonyms is not controllable. The concept of intuition used in this paper is very simple to understand as: "*by enabling the synonyms to move closer to each other, the distance between antonyms will also become more noticeable*". PPDB, a thesaurus is used to offer distant supervision to the Skip-gram model.

3.3.2 The global objective function

We have utilized the cosine distance function as a global objective function to measure the similarity of word vectors. Equation (3) is used as the global objective function.

$$\begin{aligned}J(w_t,w_i) &= \cos(w_i,w_t)\\ &= \frac{w_t\cdot w_i}{||w_t||\cdot||w_i||}\end{aligned} \tag{3}$$

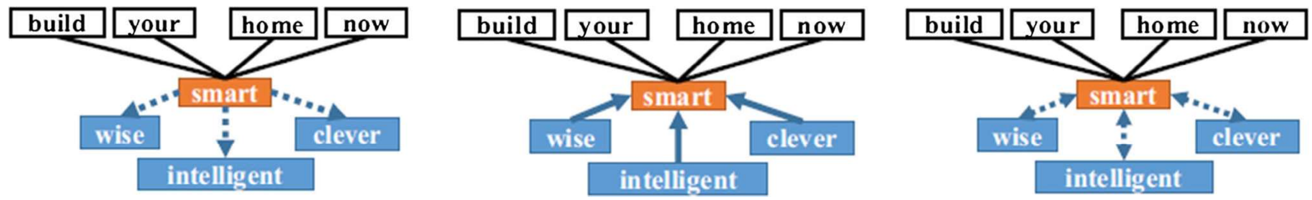
where w_t and w_i, respectively, represents target word and synonym word.

The loss function of our proposed model is determined by using Eq. (4) by summing of the cosine distance (Eq. 3) and the objective function of Skip-gram (Eq. 2). For a word sequence $(w_1, w_2, \ldots, w_n)$ and target word w_t, the model intends to maximize.

$$L(H) = \Pr(w_1,\ldots,w_n|w_t) + \alpha.J(w_t,w_{syn}) \tag{4}$$

where w_{syn} is the synonym for target word w_t and $\Pr(w_1, w_2, \ldots, w_n|w_t)$ represents the predictive probability of context words conditioned on the target word w_t. α is the weight of the external resources ranging from 0.1 to 0.2, determining how strongly the degree of movement should impact of optimization process. If the value of α becomes

Journal : Large 521 | Dispatch : 4-2-2019 | Pages : 11
Article No. : 4071 | LE | TYPESET
MS Code : NCAA-D-18-02469 | CP | DISK



(a) UMT Model **(b)** UMS Model **(c)** BMTS Model

Fig. 2 Unidirectional Movement of Target Model. **a** Unidirectional Movement of Synonyms Model. **b** Bidirectional Movement of Target and Synonyms Model. **c** Yellow rectangle represents the target word; blue rectangle depicts synonyms and the white rectangle shows context. The dashed line in blue represents the random selecting and updating of a synonym, and the solid line in blue represents the update of all synonyms

higher, then the distributional representations will rely more on distant supervision.

3.3.3 Distant supervision models

Distance between synonyms and a target word can be reduced by updating either a target word or the synonyms. Based on this consideration, three distant supervision models are introduced: *Unidirectional Movement of Target (UMT) model*, *Unidirectional Movement of Synonyms (UMS) model* and *Bidirectional Movement of Target and Synonyms (BMTS) model*. Figure 2 illustrates the moving direction of synonyms and target words in all three proposed models.

UMT model It randomly selects a synonym of the target word and moves the target word toward the synonym. This phenomenon of movement is called as *Unidirectional Movement of Target* (UMT). To make a movement, UMT model first determines the Cosine similarity between the target words and synonyms using Eq. (3) and, then, updates the target word vectors by the Cosine loss function utilizing Eq. (4). This whole process is helpful in improving the Cosine similarity between synonyms and target words.

UMS model In this model, all the corresponding synonyms of a target word are moved together toward the target. All the synonym word vectors are updated in each training step. Moreover, a *Unidirectional Movement of Synonyms with Negative Sampling* (NUMS) model is also proposed which is used to update representations in negative sampling. The updated frequency of samples in a NUMS model is much higher than original UMS model.

BMTS Model It randomly chooses one synonym to calculate the loss function using Eq. (4). BMTS model tries to move the target word vector as well as the synonym word vector as illustrated in Fig. 2c. As we can see, the movement is in both directions; therefore, it is referred as *Bidirectional Movement of Target and Synonyms* (BMTS).

4 Computational simulation

Extensive computation simulations have been conducted to evaluate the performance of the proposed models. In this section, first, we discussed about parameters setting, test data and factors used for quality measure. Second, we have presented experimental results and analysis. Third, a case study is presented to develop a deeper understanding about the working of the proposed models.

4.1 Parameters setting

In order to balance the quantity and accuracy, word pairs are selected from the XL size data of PPDB. As the number of synonyms is not balanced, ranging from one to hundreds, the top five synonyms are used by ranking them with TF-IDF value. In total, the obtained synonym vocabulary is with more than 50 k words. The proposed model is trained with the 2010 English dump from the Wikipedia. Data preprocessing includes removing the numbers, special symbols, and non-English words from corpus and converting all English letters to lowercase. In this paper, we trained three models discussed above and evaluate them on different tasks. Since our models are all based on Skip-gram model, it is reasonable to use Skip-gram as baseline.[1] In addition, our proposed models are compared with three similar models: *dict2vec*, *JointRCM* and *WE-TD* which also introduce external resources into training. For these three models, their experiments are reproduced with the hyper-parameters as described in [25, 29, 32]. For all model parameter settings, it is used with 5 negatives samples, 4 epochs, 5 window sizes, and the embedding dimension is 200. The learning rate of our model is 0.025.

[1] https://code.google.com/archive/p/word2vec/. 1FL01

Journal : Large 521 Dispatch : 4-2-2019 Pages : 11
Article No. : 4071 LE TYPESET
MS Code : NCAA-D-18-02469 CP DISK

4.2 Performance analysis

The performances of the proposed models are evaluated on four tasks: *word analogy* (intrinsic), *synonym-antonym detection* (intrinsic), *sentence matching* (extrinsic) and *text classification* (extrinsic).

Word analogy is a widely used method for evaluating embedding. This test set is designed to verify whether the trained word vectors can express syntactic and semantic relationships. Google analogy dataset is used with 19,544 questions (8869 semantic and 10,675 syntactic questions) and 14 types of relations.

A test set is constructed for synonym and antonyms.[2] Each line in this test set has three words: *target word*, *antonym*, and *synonym*. The target-antonym pairs are obtained from WordNet, and target-synonym pairs are obtained from PPDB. This dataset contains 3387 triples. The cosine distance is calculated between target-antonym and *target-synonym,* and correctness is judged by whether *target-synonym* is closer.

The Stanford Natural Language Inference (SNLI) is used, which contains 367,373 sentences pairs and 29,899 words. Each sentence pair consists of three parts: *target sentence*, *comparison sentence* and *labels*. Labels with 0 and 1 represent if these two sentences can match in semantics. Each target sentence has more than 2 comparison sentences and their labels are not the same. Word movers distance (WMD) and word centroid distance (WCD) [16] are used to calculate the similarity of sentences with normalized vectors. The correctness of certain target sentence is judged by calculating the similarity of all target's comparison sentences and choose the most similar one; if the label of this sentence is 1, then it is correct to this sentence.

Text classification is a typical example of whether word embedding can contribute to a specific NLP task. In this task, the AG's news dataset is used for training and testing. In this dataset, the size of training set is 120,000 and the test set is 7600. The news is classified into four types. Each type has 30,000 training samples and 1900 testing samples. Two methods are used to evaluate classification tasks: *Logistic Regression* (LR) and *Convolution Neural Network* (CNN). For logistic regression, average sum of word vectors is adopted as a sentence vector with L2 regularization, while, for CNN, the CNN text classification model [15][3] is used. Since the evaluation focuses on the embedding performance, this paper follows the settings of [29] to fix the embeddings; thus, they will not be updated during training.

[2] It is planned to release all the datasets and code used in this study after the paper is published.

[3] https://github.com/wabyking/TextClassificationBenchmark [31].

Table 2 Results on word analogy task. Accuracy is the percentage of correct positive samples of analogy test result. Mean rank is the average rank of correct positive samples

	Analogy	
	Accuracy (in %)	mean rank
Skip-gram	64.66	714
JointRCM	47.53	2300
WE-TD	49.86	2258
dict2vec	44.01	3612
UMT	**66.78**	632
UMS	65.28	617
BMTS	65.72	**556**

4.3 Results and analysis

4.3.1 Analogy

Table 2 shows that the proposed models perform higher than baseline, while JointRCM, WE-TD and dict2vec perform poorly in capturing semantic and syntactic relationships. Their performances are, respectively, 17.13, 14.8 and 20.65% lower than the baseline. All our model variations generally perform better than Skip-gram model. Specifically, the UMT, UMS and BMTS models improve the performance by 2.12%, 0.62%, 1.06%, respectively.

The mean ranks of the JointRCM, WE-TD and dict2vec model are 1544 higher than baseline model on average. UMS model has the lowest mean rank value compared to other models. Besides, the average mean rank values of our models are 112 lower than the baseline. The overall response of our three models to this task is very positive. And it is observed that the UMT model is more suitable in analogical reasoning of linguistic regularities.

4.3.2 Recognition of synonyms and antonyms

As shown in Table 3, all our models have achieved considerable improvements as compared with the baseline. It should be noted that the WE-TD model gets the best performance in this task since its objective function is specially designed for this task by maximizing the similarity between synonyms and minimizing the similarity between antonyms. In addition to WE-TD, NUMS (UMS with negative sampling) model achieves a result 20.78% higher than Skip-gram, 4.54% higher than JointRCM and 20.43% higher than dict2vec.

It is concluded that the NUMS model is particularly suitable for recognition of synonyms and antonyms, which means a higher update frequency has a positive effect on this task. By enabling the synonyms to move closer to each other, the distance between antonyms also become more

Journal : Large 521 Dispatch : 4-2-2019 Pages : 11
Article No. : 4071 ☐ LE ☐ TYPESET
MS Code : NCAA-D-18-02469 ☑ CP ☑ DISK

Table 3 Results of recognition of synonyms and antonyms (RSA) and sentence matching task. Means are the mean sentence similarity on the correct positive example

	RSA (%)	WCD (%)	Mean (%)	WMD (%)	Mean (%)
Skip-gram	29.85	63.37	0.310	69.83	0.691
JointRCM	46.09	60.58	0.284	67.52	0.624
WE-TD	**77.42**	62.77	0.343	69.53	0.724
dict2vec	30.20	62.68	0.231	69.02	0.520
UMT	29.97	63.90	0.305	70.20	0.669
UMS	32.54	63.70	0.296	70.08	0.648
BMTS	32.06	63.39	0.296	70.23	0.644
NUMS	50.63	**64.64**	**0.176**	**71.32**	**0.353**

noticeable. Unlike the WE-TD model which minimizes the similarity between antonyms, our unidirectional and bidirectional movements do not affect the relevance between antonyms.

4.3.3 Sentence matching

In sentence matching task from Table 3, the NUMS model achieves a state-of-the-art result, and the newly proposed models all have gained significant improvements. However, the JointRCM, WE-TD and dict2vec methods do not perform well.

Accuracy NUMS improves the performance by 1.49% in WMD and 1.27% in WCD. The UMT, UMS and BMTS models also achieve better results than the baseline. However, the performances of JointRCM, WE-TD and dict2vec are all lower than the baseline.

Mean value The proposed models make obvious progress on the mean value of WCD and WMD. Notice that the mean values of JointRCM and dict2vec model are also smaller than the baseline because the distances between synonyms are shortened. However, embeddings trained by these two models tend to confuse similar words with relevant words; thus, their performances in sentence matching task are not satisfactory.

WMD is highly interpretable because the distance between two documents can be broken down and explained as the sparse distances between several few individual words and it naturally incorporates the knowledge encoded in the word2vec space. The closer distance between synonyms results in smaller mean distance value. The results confirm that our UMS model is a good choice for sentence matching task.

4.3.4 Text classification

Table 4 shows that our models outperform the baseline and other models on both CNN and LR implementations. While embeddings trained by JointRCM, WE-TD and dict2vec scored lower than the baseline. Our UMT model with CNN improves the accuracy from 90.90 to 91.26%. Results of JointRCM, WE-TD and dict2vec are lower than the baseline. The results indicate that UMS model achieves a 0.25% improvement over the baseline and a 1.14% over the dict2vec.

Table 4 Results on text classification tasks

	Classification	
	CNN (%)	LR (%)
Skip-gram	90.90	88.21
JointRCM	90.77	87.83
WE-TD	90.64	87.57
dict2vec	90.33	87.32
UMT	**91.26**	**88.45**
UMS	91.18	**88.46**
BMTS	91.13	88.37

The LR linearly learns the relationship between the basic word vector and the final labels, while the CNN adopts a high-level feature extraction from the word vector. As shown in Table 4, the proposed models outperform all the baselines with both LR and CNN cases. This result evident that our models not only capture the word-level task as shown in the word analogy task, but also can benefit some upstream tasks in which the text representation in text classification is the most typical one.

4.3.5 Evaluation of α

To select an appropriate α and evaluate the impact of different α values on model performance, this paper trains models with different α values. Our test is based on the UMS model and α is selected within 0.08, 0.1, 0.12, 0.15, 0.2 and 0.25, respectively. Figure 3 shows the performance AQ4 with different α on each task. The results indicate that the value of α has a great influence on different tasks. Figure 3 derives the following points:

(a) The result of analogy test will decrease with the increase of α.

Journal : Large 521 | Dispatch : 4-2-2019 | Pages : 11
Article No. : 4071 | □ LE | □ TYPESET
MS Code : NCAA-D-18-02469 | ✔ CP | ✔ DISK

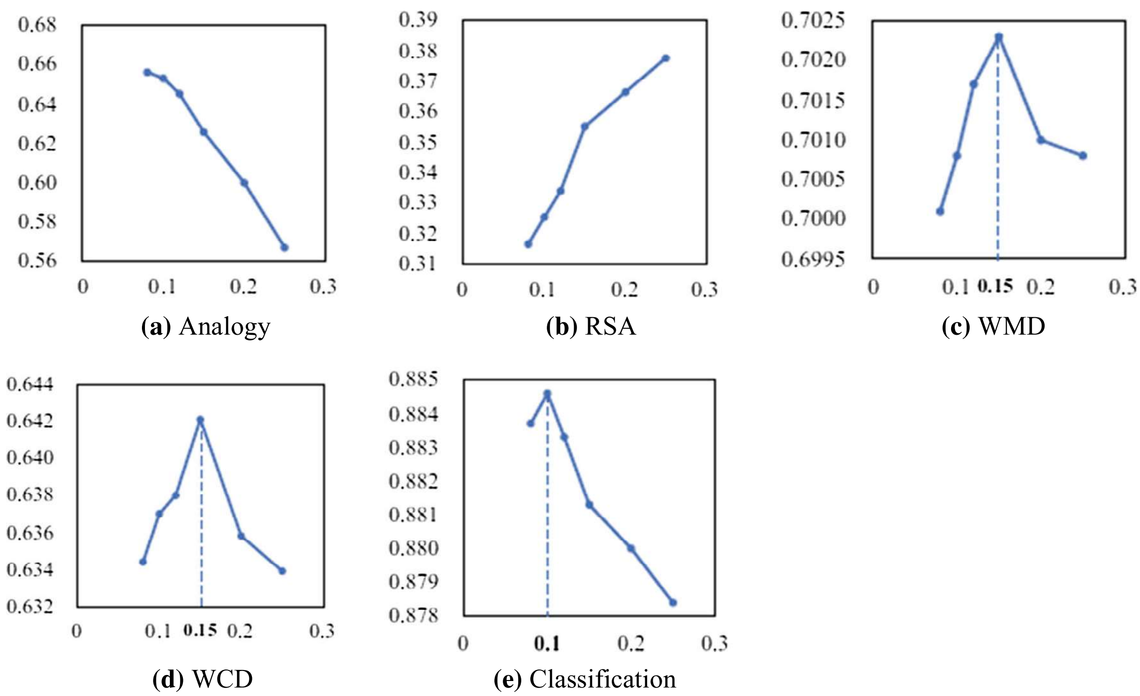


Fig. 3 Evaluation of different α based on UMS model

(b) The result of antonym test will increase with the increase of α.

(c) The result of sentence similarity comparison test and the classification test will firstly increase and then decrease with the increase of α.

Classification task will get a best result with α value equal to 0.1, the optimal value is 0.15 on sentence matching task. Different values of α for different tasks can be chosen.

4.4 Case study

The above experiments verify the effectiveness of the proposed models in all tasks. The proposed models have achieved a state-of-the-art result in sentence matching task where precise semantic information is required. To further elaborate the mechanism and effect of the proposed model, we presented a case study that shows several examples of sentences and words.

4.4.1 Improvements on word similarity

Examples of word similarity are shown in Table 5. Words are sorted from top to bottom in descent order of word similarity, in term of cosine distance. The chosen target words are continuous, precise and red. Top six similar words are chosen.

For the first two words, the antonyms are marked in red. It can be observed that UMS model performs the best as there are no antonyms in the top six similar words. It is noticed that the antonyms are the second most similar word in Skip-gram model which cannot distinguish between antonyms in the same contexts. For Joint RCM model, this problem also appears with the two words. For WE-TD model, it performs well on precise and does not differ from other models on continuous. For dict2vec, the most similar words are words with the same roots. For the proposed model, the result of UMT is similar to Skip-gram since it introduces the weakest supervision of external resources among these three models.

For the word red it has no synonyms or antonyms. The word is marked in blue if it is not color. The most similar words in our model and Skip-gram are all colors while irrelevant words appear in JointRCM, WE-TD and dict2vec.

From these cases, it is demonstrated that proposed models not only have better results in distinguish synonyms and antonyms (in the continuous and precise cases) but also capture effective global semantic information of words (in red cases). In particular, the UMS model has the best performance in these three cases without any antonyms.

4.4.2 Improvements on sentence similarity

Two cases from sentence similarity are chosen. Each model calculates the given sentence (in the first row) with all the candidate sentences and the sentence with the highest similarity score is shown in Table 6.

Journal : Large 521 | Dispatch : 4-2-2019 | Pages : 11
Article No. : 4071 | ☐ LE | ☐ TYPESET
MS Code : NCAA-D-18-02469 | ☑ CP | ☑ DISK

Table 5 Case Study on word similar task. Word in red are antonyms, word in blue are irrelevant word

Targets	Skip-gram	JointRCM	WE-TD	dict2vec	UMT	UMS	BMTS
continuous	Uninterrupted	Ceaseless	Uninterrupted	Continuously	Continual	Continual	Continual
	Noncontinuous	Uninterrupted	Continual	Semicontinuous	Uninterrupted	Uninterrupted	Constant
	Continual	Uninterruptible	Constant	Discontinuous	Discontinuous	Constant	Uninterrupted
	Discontinuous	Discontinuous	Piecewise	Uninterrupted	Noncontinuous	Linear	Continuously
	Continuously	Continual	Persisting	Intervals	Singlevalued	Continuously	Linear
	Semiinfinite	Constant	Dogging	Sinusoidal	Piecewise	Minimal	Discontinuous
precise	Accurate	Accurate	Exact	Accurate	Accurate	Accurate	Accurate
	Exact	Imprecise	Accurate	Imprecise	Exact	Exact	Exact
	Imprecise	Unambiguous	Repeatable	Accurately	Imprecise	Correct	Precisely
	Unambiguous	Correct	Meticulous	Inexact	Repeatable	Precisely	Correct
	Accurately	Repeatable	Punctual	Semidefinite	Precisely	Accurately	Accurately
	Repeatable	Meticulous	Scrupulous	Accuracy	Unambiguous	Consistent	Timing
Red	Blue	Blue	Blue	Redder	Blue	Blue	Blue
	Yellow	Bluefin	Elvises	Reds	Yellow	Yellow	Yellow
	White	Yellow	Redyellow	Yellow	White	Purple	Purple
	Lightblue	Puce	Blue	Blu	Black	Pink	White
	Purple	Sox	Orange	Bureaucratic	Purple	Green	Green
	Skyblue	Yellow	Yellower	Bleu	Pink	Black	Pink

Table 6 Case Study on sentence similarity task. Word in red are antonyms, word in blue are categories word

	Two men waiting outside the door on a snowy night.	A dog chases a dog toy on the grass.
WE-TD	Two people are indoors on a snowy night	A dog slips on the wet grass
Skip-gram	Two people are indoors on a snowy night	A dog slips on the wet grass
JointRCM	Two people are indoors on a snowy night	A dog chases a cat onto the sofa
dict2vec	Two men are sitting outside of a store on a sunny day	A dog slips on the wet grass
NUMS	Some men are standing outside in the snow	A dog is running on the grass

The main components of the candidate sentences are similar, while antonyms are marked in red and words of the same category in blue.

For the first case in the second column, WE-TD, Skip-gram and JointRCM consider indoors as the similar word with *outside*. However, the meanings of *indoors* and *outside* are opposite, which is the typical case that these models are usually confused with the synonym and antonyms. Toward dict2vec model, it considers *sunny* as the similar to *snowy*. The proposed model has its advantage to correctly distinguish the synonym pair between *snowy* and *snow*; as well the antonyms pair between *outside* and *indoors*. In the second case in the 3rd column, the proposed models also show it effectiveness to process these word pairs like *chases* and *running*.

In conclusion, it is shown from Table 6 that the proposed models have ability to effectively distinguish between antonyms and do not confuse the synonyms with words of the same category. Due to this, the proposed models effectively incorporate the synonyms and antonyms resources in both syntagmatic and paradigmatic relations.

5 Conclusions

Incorporation of the linguistic knowledge is one of the key concerns in current paradigm of the NLP. This paper proposed a distant supervision method to learn improved word representations, in order to extra incorporate the synonyms resources in paradigmatic relations. Our three variants of the proposed methods have been demonstrated with its effectiveness in four typical benchmarks: *analogy*, *recognition of synonyms and antonyms*, *sentence matching* and *text classification*.

The word embedding is one of the key input features for most typical tasks in the NLP. Although there are more and more network architectures in current NLP community, more attention should be paid in the inputting side (namely

Journal : Large 521 | Dispatch : 4-2-2019 | Pages : 11
Article No. : 4071 | ☐ LE | ☐ TYPESET
MS Code : NCAA-D-18-02469 | ☑ CP | ☑ DISK

word embedding), instead of only intermediate architectures. From empirical point of view, external resources, like linguistic knowledge and general common sense are also essential for NLP. In order to extend the proposed models in a more general purpose, a larger-scale corpus and benchmarks should be used in the future. Meanwhile, it is expected to directly model naturally both the synonyms and antonyms information in the phase part of complex-AQ5 valued word embedding [18].

Acknowledgements This work is supported by the Quantum Access and Retrieval Theory (QUARTZ) project, which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie SkCodowska-Curie grant agreement No. 721321.

Compliance with ethical standards

Conflict of interest The authors declare no conflict of interest.

References

1. Adel H, Schütze H (2004) Using mined coreference chains as a resource for a semantictask. In: Proceedings of the 2014 conference on empirical methods in natural language processing(EMNLP), pp 1447–1452
2. Baker CF, Fillmore CJ, Lowe JB (1998) The berkeley framenet project. In: Proceedings of the 17th international conference on Computational linguistics, vol 1. Associationfor Computational Linguistics, pp 86–90
3. Bian J, Gao B, Liu T-Y (2014) Knowledge-powered deep learning for word embedding. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 132–148
4. Chen Z, Lin W, Chen Q, Chen X, Wei S, Jiang H, Zhu X (2015) Revisiting word embedding for contrasting meaning. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers), vol 1, pp 106–115
5. Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th international conference onMachine learning. ACM, pp 160–167
6. Culler JD (1986) Ferdinand de Saussure. Cornell University Press, Ithaca
7. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deepbidirectional transformers for language understanding (2018). arXiv preprint arXiv:1810.04805
8. Huang EH, Socher R, Manning D, Ng AY (2012) Improving wordrepresentations via global context and multiple word prototypes. In: Proceedings of the 50th annual meeting of the association for computational linguistics: long papers-volume 1.Association for Computational Linguistics, pp 873–882
9. Faruqui M, Dodge J, Jauhar SK, Dyer CD, Hovy E, Smith NA (2014) Retrofitting word vectors to semantic lexicons. arXiv preprint arXiv:1411.4166
10. Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Mikolov T et al (2013) Devise: a deep visual-semantic embedding model. In: Advances in neural information processing systems, pp 2121–2129
11. Ganitkevitch J, Van Durme B, Callison-Burch C (2013) Ppdb: the paraphrase database. In: Proceedings of the 2013 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 758–764
12. Harris ZS (1954) Distributional structure. Word 10(2–3):146–162
13. Hinton GE (1986) Learning distributed representations of concepts. In: Proceedings of the eighth annual conference of the cognitive science society, vol 1. Amherst, MA, pp 12
14. Laura EB (2017) Key and Brittany Pheiffer Noble. Course in general linguistics. Macat Library
15. Kim Y (2014) Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882
16. Kusner M, Sun Y, Kolkin N, Weinberger K (2015) From word embeddings to documentdistances. In: International conference on machine learning, pp 957–966
17. Lazaridou A, Baroni M et al (2015) A multitask objective to inject lexical contrast into distributional semantics. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: short papers), vol 2, pp 21–26
18. Li Q, Uprety S, Wang B, Song D (2018) Quantum-inspired complex word embedding. arXiv preprint arXiv:1805.11351
19. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representationsin vector space. arXiv:1301.3781
20. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representationsof words and phrases and their compositionality. In: Advances in neural information processingsystems, pp 3111–3119
21. Mikolov T, Yih W-T, Zweig G (2013) Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Associationfor Computational Linguistics: Human Language Technologies, pp 746–751, 2013
22. Nguyen KA, Walde SS, Vu NT (2016) Integrating distributional lexicalcontrast into word embeddings for antonym-synonym distinction. arXiv preprint arXiv:1605.07766
23. Nguyen KA, Walde SS, Vu NT (2017) Distinguishing antonyms and synonyms in a pattern-based neural network. arXiv preprint arXiv:1701.02962
24. Nickel M, Kiela D (2017) Poincaré embeddings for learning hierarchical representations. In Advances in neural information processing systems, pages 6338–6347, 2017
25. Ono M, Miwa M, Sasaki Y (2015) Word embedding-based antonym detection using thesauri and distributional information. In: Proceedings of the 2015 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 984–989
26. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Empirical methods in natural language processing (EMNLP), pp 1532–1543
27. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. arXiv preprint arXiv:1802.05365
28. Sala F, De Sa C, Gu A, Ré C (2018) Representation tradeoffs for hyperbolic embeddings. In: International conference on machine learning, pp 4457–4466
29. Tissier J, Gravier C, Habrard A (2017) Dict2vec: learning word embeddings using lexical dictionaries. In: Conference on empirical methods in natural language processing (EMNLP2017), pp 254–263
30. Vilnis L, McCallum A (2014) Word representations via gaussian embedding. arXiv preprintarXiv:1412.6623
31. Wang B, Wang L, Wei Q (2018) Textzoo, a new benchmark for reconsidering text classification. arXiv preprint arXiv:1802.03656

32. Yu M, Dredze M (2014) Improving lexical embeddings with semantic knowledge. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 2: short papers), vol 2, pp 545–550

Journal : **Large 521** Dispatch : **4-2-2019** Pages : **11**
Article No. : **4071** ☐ LE ☐ TYPESET
MS Code : **NCAA-D-18-02469** ☑ CP ☑ DISK