Not All Synonyms Are Created Equal: Incorporating Similarity of Synonyms to Enhance Word Embeddings

Peiyang Liu* †, Wei Ye* ‡, Xiangyu Xi *†, Tong Wang†, Jinglei Zhang* †, Shikun Zhang*,
*National Engineering Research Center for Software Engineering, Peking University
†School of Software and Microelectronics, Peking University
email:{liupeiyang,jinglei.zhang,wye,xixy,zhangsk}@pku.edu.cn, 1186757843@qq.com

***Abstract*—Traditional word embedding approaches learn semantic information from the associated contexts of words on large unlabeled corpora, which ignores a fact that synonymy between words happens often within different contexts in a corpus, so this relationship will not be well embedded into vectors. Furthermore, existing synonymy-based models directly incorporate synonyms to train word embeddings, but still neglect the similarity between words and corresponding synonyms. In this paper, we explore a novel approach that employs the similarity between words and corresponding synonyms to train and enhance word embeddings. To this purpose, we build two Synonymy Similarity Models (SSMs), named SSM-W and SSM-M respectively, which adopt different strategies to incorporate the similarity between words and corresponding synonyms during the training process. We evaluated our models for both Chinese and English. The results demonstrate that our models outperform the baselines on seven word similarity datasets. For the analogical reasoning and text classification tasks, our models also surpass all the baselines including a synonymy-based model.**

***Index Terms*—word embedding, language model, synonyms, word similarity, text classification**

I. Introduction

Distributed representations of words, namely word embeddings, encode both semantic and syntactic information into a dense vector. The derived word embeddings have been used in many tasks such as text classification [1], information retrieval [2], sentiment analysis [3], etc. Most of these NLP tasks may also benefit from pre-trained word embeddings, such as CBOW [4], Skip-Gram [4] and GloVe [5], which are based on the distributional hypothesis [6], [7]: words that occur in the same contexts tend to have similar meanings. These methods ignore the truth that synonymy between words happens often within different contexts in a corpus. For example, in Fig. 1, according to WordNet [8], "Good"'s synonyms are "Well", "Right", "Honorable" and "Goodness", but the similar words calculated by CBOW are "Great", "Bad", "Lousy" and "Terrific". Obviously, CBOW embeds words only based on their syntactical structure but ignores the meaning of words. Since synonyms of words hold similar meanings, their embeddings are expected to be close enough. If we train words and corresponding synonyms separately (such as "Good" and "Well"), their latent correlation may be lost. Furthermore, we observe that some words regularly appear in our corpus while their synonyms do not, which causes a serious weakness that high-frequency words are well embedded but their low-frequency synonyms are not, although their meanings are similar. In order to solve these problems, synonymy-based models are proposed by researchers.

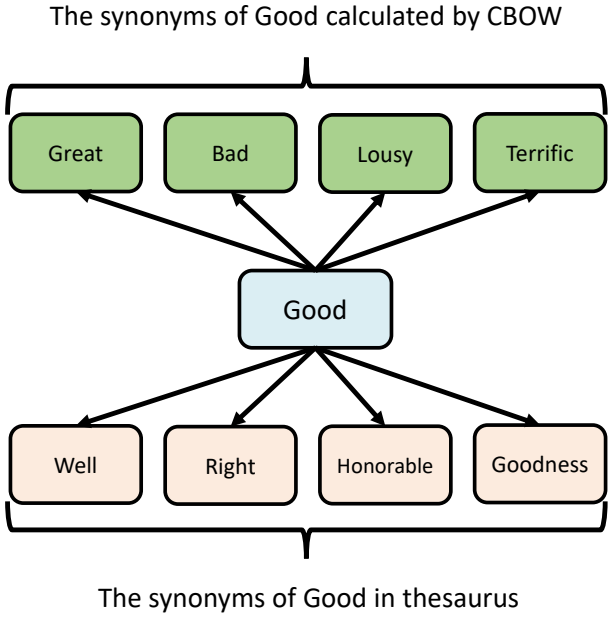

Fig. 1. A paradigm of Good's synonyms in thesaurus and calculated by CBOW.

The effectiveness of exploiting the internal synonymy between words has been confirmed by some previous work. For example, Yu groups English words into sets of synonyms called synsets [9], provides short, general definitions for them, and records various semantic relations between synsets. Bian incorporates synonymy knowledge from WordNet and Paraphrase Database into a joint model built upon Word2vec [10]. Zhang combines knowledge from multiple sources (syllables, POS tags, antonyms/synonyms, Freebase relations) with

‡Corresponding author.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

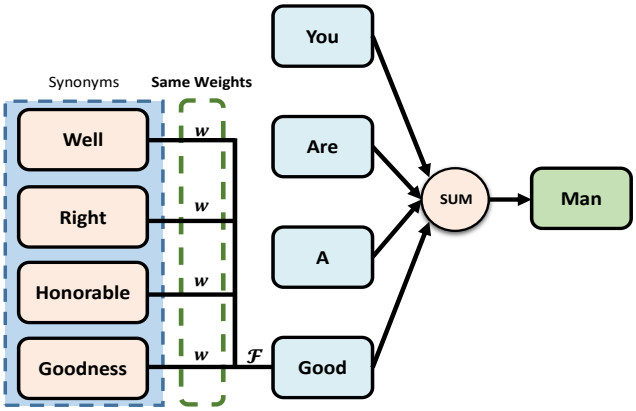

Fig. 2. A paradigm of Equal Synonymy Models based on CBOW. The sentence "You are a good man" is selected as an example. "Man" is the target word. When calculating the input vector of "Good", they first find out synonyms of "Good" in thesaurus, and add the vectors of all synonyms to the vector of "Good" with equal weights. $\mathcal{F}$ is the compositional function aggregates synonyms embeddings into target word embedding. ω is the same weight of each synonym.

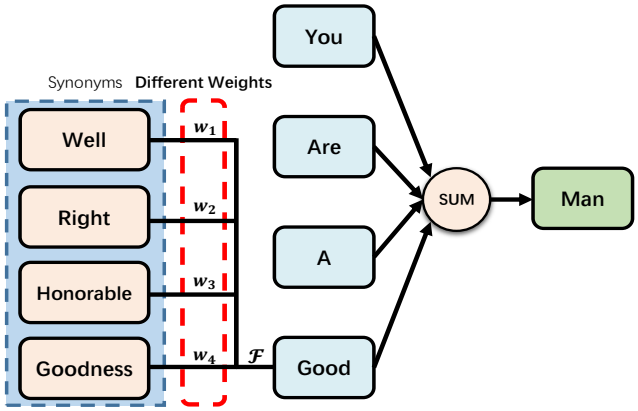

Fig. 3. A paradigm of SSM-W. In this model, all synonyms of "Good" are added together with different weights.

CBOW model [11]. Ono proposes a Bayesian Probabilistic Tensor Factorization (BPTF) model to combine thesauri information and existing word embeddings [12]. Hasegawa utilizes supervised synonym and antonym information from thesauri to enhance word embeddings [13]. Alsuhaibani incorporates visual features into word embeddings to represent the similarity of synonyms [14]. Sun learns word embeddings using a corpus and a knowledge base which contains information of synonymy between words [15].

As shown in Fig. 2, we refer to all of above synonymy-based models as *Equal Synonymy Models*. They further consider synonyms of words when training word embeddings, which achieves significant improvement. However, Equal Synonymy Models assume that all synonyms of a word have equal impacts to the word's embedding during training process, which ignores the similarity variance among synonym pairs. For instance, it is obvious that "Well" is semantically closer to "Good" than "Honorable". Our intuition is that a synonym which is semantically closer to the target word should have a greater impact on its embedding during the training process. Therefore, we explore a new way to incorporate the similarity values of synonym pairs into the training process and thus enhance word embedding.

In this paper, we consider different methods to change the input layer and update rules of CBOW [4], and propose two lightweight and efficient models, which are called Synonymy Similarity Models (SSMs), to encode synonymous properties into words as well as to enhance the semantic similarities among word embeddings.

For evaluation, we compared our SSMs with the state-of-the-art baselines on two basic NLP tasks, which are word similarity and analogical reasoning, and downstream tasks of text classification. The results show that our models outperform the baselines and can achieve satisfactory improvement on these tasks. In summary, the main contributions of this paper are summarized as follows.

- Rather than assuming that all synonyms of a word have equal impacts to the word's embedding, we propose to employ the similarity between a word and its corresponding synonyms to train the word embeddings. To validate the feasibility of our intuitive idea, we propose two specific models, SSM-W and SSM-M, with different strategies to incorporate the similarity values.
- We utilized a medium-sized corpus to train SSMs and the state-of-the-art baselines, and evaluated their performance on two basic NLP tasks, i.e., word similarity and analogical reasoning, and downstream text classification tasks. The results show that SSMs outperform the baselines on all of these tasks for both Chinese and English.

II. Background and Related Work

Considering the high efficiency of CBOW [4], we build SSMs upon it. In this section, we first review the background of CBOW, and then present some related work on recent synonymy-based word embedding methods.

CBOW with Negative Sampling With a sliding window, CBOW predict the target word according to the contextual words in the window. Given a sequence of tokens $T = \{t_1, t_2, ..., t_n\}$, in which n is the size of training corpus, the objective of CBOW is to maximize the following average log probability equation:

$$L = \frac{1}{n}\sum_{i=1}^{n} \log p(t_i|context(t_i)) \quad (1)$$

where $context(t_i)$ represents the context words of t_i in the slide window, and $p(t_i|context(t_i))$ is derived by softmax. Due to the huge size of vocabulary, it is difficult to calculate $p(t_i|context(t_i))$ in acceptable amount of time. Therefore, negative sampling and hierarchical softmax are proposed to solve this problem [4]. In order to train model efficiently, all of our models are trained based on negative sampling. In

terms of negative sampling, the log probability $\log p(t_O|t_I)$ is transformed to:

$$\log \delta(vec'(t_O)^T vec(t_I)) + \sum_{i=1}^{m} \log[1 - \delta(vec'(t_i)^T vec(t_I))] \quad (2)$$

where m represents the number of negative samples, and $\delta(\cdot)$ is the sigmoid function. The first item of Eq. (2) is the probability of target word on condition of the context. The second item indicates the probability of negative samples which hold different contexts with the target word.

Synonymy-based Word Embedding Recently, some more efficient word embedding methods are proposed by exploiting the synonyms of words. These synonymy-based models can be divided into two main branches.

The first branch directly adds the synonyms to word embeddings or optimizes a joint objective over distributional statistics and synonyms. [9], [10], [12], [14]. Yu proposed a prior-knowledge-enhanced word embedding model, which incorporates prior knowledge about synonyms from WordNet and Paraphrase Database into a joint model built upon CBOW [4] [9]. Bian incorporate synonyms and antonyms in to a CBOW model [10]. Ono uses supervised synonym and antonym information from thesaurus to enhance word embeddings [12]. Alsuhaibani incorporate a corpus and a knowledge base which contains information of synonymy between words to train word embeddings [14].

The other branch tries to use probabilistic graphical models to connect words with their synonyms, and further learns word embeddings. Zhang proposes a Bayesian Probabilistic Tensor Factorization (BPTF) model to combine information of thesaurus and existing word embeddings [11].

However, these synonymy-based models directly exploit the synonyms of words to train word embeddings, which ignore the similarity values between words and their corresponding synonyms. In contrast, we employ the similarity values of synonym pairs to provide deeper insights for training enhanced word embeddings.

III. Our Synonymy Similarity Models

We leverage two different strategies to modify the input layer and update rules of CBOW when incorporating the synonyms of words. We propose two specific models, named Synonymy Similarity Model-Weighted (SSM-W) and Synonymy Similarity Model-Max (SSM-M). Our intuition is that synonyms which hold a closer meaning to the corresponding word should have a greater impact on its word embedding during training. Inspired by the attention scheme, in the SSM-W model, the impact of a target word's each synonym on the target word's embedding is determined by its similarity with the target word. In contrast, in SSM-M, we only keep the synonym which is the most similar to the target word and discard all other synonyms. We present detailed description of SSM-W and SSM-M in the following subsections, respectively. At the end of this section, we introduce the update rules of our SSMs.

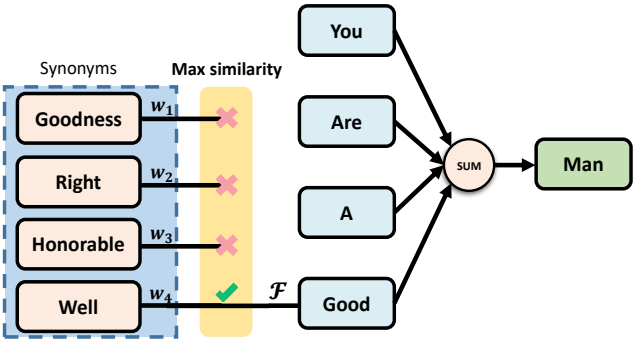

Fig. 4. A paradigm of SSM-M. The synonym with maximum similarity towards "Good" are selected.

A. SSM-W

The SSM-W model is built based on attention scheme. We observe that many words have more than one synonyms. For example, "Good" has synonyms "Well", "Right", "Honorable", etc. As shown in Fig. 3, the synonyms of "Good" hold different biases towards it. Therefore, we assign different weights to these synonyms, and measure these weights by calculating the normalized similarities between token t_i and the corresponding synonyms. For SSM-W, the modified embedding of t_i is

$$\hat{v}_{t_i} = \frac{1}{2}\left[v_{t_i} + \sum_{s \in S_i} w_{\langle t_i, s\rangle} \cdot v_s\right] \quad (3)$$

where v_{t_i} is the original vector of t_i, S_i is a set of synonyms of t_i, and $w_{\langle t_i,s\rangle}$ denotes the weight between t_i and the synonyms $s \in S_i$. We use $\cos(v_a, v_b)$ to denote the cosine similarity between v_a and v_b then $w_{\langle t_i,s\rangle}$ is expressed as follows:

$$w_{\langle t_i,s\rangle} = \frac{\cos(v_{t_i}, v_s)}{\sum_{x \in S_i} \cos(v_{t_i}, v_x)} \quad (4)$$

B. SSM-M

To further eliminate the impacts of some uncorrelated synonyms to a word. In SSM-M, we only select the synonym which is the most similar to the token t_i, As shown in Fig. 4, the synonym "Well" of "Good" is finally selected since the "Well" is semantically closest to "Good". For token t_i SSM-M is expressed as

$$\hat{v}_{t_i} = \frac{1}{2}\left[v_{t_i} + w_{\langle t_i, S^i_{max}\rangle} \cdot v_{S^i_{max}}\right] \quad (5)$$

Where S^i_{max} is the synonym with maximum similarity towards token t_i, and S^i_{max} is mathematically defined as

$$S^i_{max} = argmax_s \cos(v_{t_i}, v_s), s \in S_i \quad (6)$$

The normalized weight $w_{\langle t_i, S^i_{max}\rangle}$ can similarly be derived like Eq. (4).

C. Update Rules for SSMs

After modifying the input layer of CBOW, Eq. (1) can be expressed as:

$$\hat{L} = \frac{1}{n}\sum_{i=1}^{n} \log p(v_{t_i}| \sum_{t_j \in context(t_i)} \hat{v}_{t_j}) \quad (7)$$

Where $\hat{v}_{t_j}$ is the modified vector of $v_{t_j}(t_j \in context(t_i))$. Since the thesaurus describes top-level relations between words and the synonyms, these relations don't change during the training period. When the gradient is propagated back to the input layer, we update not just the word vector $v_{t_j}(t_j \in context(t_i))$, but the vectors of the synonyms in the vocabulary with the same weights as they are added to the vector v_{t_j}.

IV. Experimental Setup

Before we describe the experiment results, we first introduce experimental settings in this section.

A. Corpus and Synonym Map

We evaluate our model for both Chinese and English. For Chinese, we select a human-annotated corpus with news articles from *The People's Daily* for embedding learning. The corpus has 31 million words. To achieve higher quality for the word embeddings, we filter all digits and some punctuation marks out of the corpus, and use THULAC [15] to segment Chinese word.

For English, we utilize a medium-sized corpus, which stems from the website of the 2013 ACL Workshop on Machine Translation and is used in [16]. We chose the news corpus of 2009 whose size is about 1.7GB. It contains approximately 500 million tokens and 600,000 words in the vocabulary. As did in [16], to get better quality of the word embeddings, we filter all digits and some punctuation marks out of the corpus.

To create the synonym map, we need to obtain the synonyms of each word and interpret them with the lookup table. We built Chinese synonym map according to HIT-CIR Tongyici Cilin (Extended) [17], and English synonym map according to WordNet [8].

B. Baselines

For comparison, we chose three word-level state-of-the-art word embedding models including CBOW, Skip-Gram [4] and GloVe [5], and we also implemented a Equal Synonymy Model, which is a variant version of the previous work [9], where all synonyms of a word make equal impacts to the word's embedding during the training. This enables our evaluation to focus on the critical difference between our models and the Equal Synonymy Model. We utilize the source code of word2vec to train CBOW and Skip-Gram. GloVe is trained based on the code. We modified the source code of word2vec and train our SSMs and the Equal Synonymy Model.

C. Parameter Settings

Parameter settings have a great effect on the performance of word embeddings [18]. For fairness, all models are trained based on equal parameter settings. In order to accelerate the training process, CBOW, Skip-Gram and Equal Synonymy Model together with our SSMs are trained using the negative sampling technique. Since it is suggested that the number of negative samples in the range 5-20 is useful [19], we set the number of negative samples to be 20 in this paper. The Same as [20], dimension of word embedding is set to be 200, and the context window size is set to be 5 which is equal to the setting in [19].

D. Evaluation Benchmarks

To compare the quality of trained word embeddings with different models, we evaluated them on three standard tasks: word similarity, analogical reasonging, and text classification.

1) Word Similarity: We use this task to evaluate the ability of word embeddings to capture semantic information from corpus. For the task of Chinese word similarity, we employed two manually labeled datasets including wordsim-240 and wordsim-296 provided by [21]. Each dataset contains a list of word pairs with a human-labeled score on how related or similar the two words are. In wordsim-240, there are 240 pairs of Chinese words and human-labeled relatedness scores. Among the 240 word pairs, the words in 233 word pairs have appeared in the learning corpus and there are new words in the remaining 7 word pairs. In wordsim-296, the words in 280 word pairs have appeared in the learning corpus and the remaining 16 pairs have new words.

For English word similarity, we employed two manually labeled datasets including Wordsim-353 [22] and RG-65 [23] as well as other widely-used datasets including Rare-Word [24], SCWS [25], Men-3k [26] and WS-353-Related [27]. More details of these datasets are shown in Table II.

To evaluate the quality of word embeddings, we calculates the Spearman correlation [28] between the labeled scores and scores generated by the word embeddings.

2) Analogical Reasoning: This task consists of analogies such as "father is to man as mother is to woman". Embedding methods are expected to find a word x such that its vector x is closest to vec(woman) - vec(man) + vec(father) according to the cosine similarity. If the word "mother" is found, the model is considered having answered the problem correctly.

For Chinese, we used the dataset collected by [21], which consisting of 1125 analogies. It contains 3 analogy types: (1) capitals of countries (687 groups); (2) states/provinces of cities (175 groups); and (3) family words (240 groups). The learning corpus covers more than 97% of all the testing words.

For English, we used the Microsoft Research Syntactic Analogies dataset, which is divided into adjectives, nouns and verbs by Mikolov [29] with a size of 8000.

3) Text Classification: We use the text classification task to evaluate word embeddings on a more applied usage scenario.

For Chinese, we use datasets collected by [30], which contain four domains of Chinese reviews: notebook, car, cam-

DataSet	CBOW	Skip-Gram	GloVe	Equal Synonymy Model	SSM-W	SSM-M
Chinese WS-240 233 Pairs	55.35	55.13	47.27	57.77	**58.23**	58.15
Chinese WS-240 240 Pairs	55.89	55.42	49.28	56.92	57.31	**58.28**
Chinese WS-296 280 Pairs	61.28	59.82	49.87	61.55	**63.67**	63.29
Chinese WS-296 296 Pairs	58.67	52.78	44.85	60.83	63.21	**63.85**
English RG-65	56.50	62.81	59.92	60.45	62.49	**63.01**
English RW	40.58	36.42	33.40	40.12	**40.65**	40.38
English SCWS	**63.13**	60.20	47.98	60.44	61.91	61.77
English Men-3k	68.07	66.30	60.56	68.19	**68.25**	68.21
English Wordsim-353	58.77	61.94	49.40	60.08	62.46	**62.51**
English WS-353-REL	49.72	57.05	47.46	57.18	57.31	**57.34**

TABLE I
EVALUATION RESULTS ON WORD SIMILARITY (ρ * 100).

Name	Pairs	Name	Pairs
RG-65	65	RW	2034
SCWS	2003	Men-3k	3000
Wordsim-353	353	WS-353-REL	252

TABLE II
DETAILS OF ENGLISH DATASETS. THE COLUMN "PAIRS" SHOWS THE NUMBER OF WORD PAIRS IN EACH DATASET.

Method	Total	Capital	State	Family
CBOW	53.83	52.89	68.28	61.63
Skip-Gram	68.85	61.92	82.85	79.34
GloVe	65.81	66.32	54.93	63.37
Equal Synonymy Model	69.43	65.97	77.36	80.14
SSM-W	**69.49**	67.68	**83.21**	**82.18**
SSM-M	70.36	**68.24**	79.76	81.33

TABLE III
ACCURACY (%) FOR CHINESE ANALOGICAL REASONING TASK.

Method	Total	Adjectives	Nouns	Verbs
CBOW	12.85	9.63	11.37	17.54
Skip-Gram	12.90	9.92	11.51	17.29
GloVe	13.27	10.13	11.86	17.81
Equal Synonymy Model	12.83	9.83	11.45	17.23
SSM-W	13.27	10.08	**11.89**	17.85
SSM-M	**13.33**	**10.21**	11.77	**18.02**

TABLE IV
ACCURACY (%) FOR ENGLISH ANALOGICAL REASONING TASK.

Method	English	Chinese	Δ Eg	Δ Ch
CBOW	78.26	83.15	-	-
Skip-Gram	79.40	82.47	+1.14	-0.68
GloVe	77.01	79.27	-1.25	-3.88
Equal Synonymy Model	80.14	82.62	+1.88	-0.53
SSM-W	80.67	**85.57**	+2.41	**+2.42**
SSM-M	**81.28**	85.20	**+3.02**	+2.05
BERT	83.39	-	-	-
BERT+CBOW	82.81	-	-0.58	-
BERT+SSM-W	83.56	-	+0.17	-
BERT+SSM-M	**83.72**	-	**+0.33**	-

TABLE V
ACCURACY (%) FOR TEXT CLASSIFICATION TASK. Δ EG REPRESENTS DIFFERENCE BETWEEN BASELINE'S SCORE ON ENGLISH. Δ CH REPRESENTS DIFFERENCE ON CHINESE.

era, and phone. They manually labeled the sentiment polarity towards each aspect target as either positive or negative. It is a binary classification task. As what we do in corpus, we use THULAC [15] to segment Chinese word.

For English, we also conduct 4 text classification tasks using the 20 Newsgroups dataset. The dataset totally contains around 19,000 documents of 20 different newsgroups, and each corresponding to a different topic, such as guns, motorcycles, electronics and so on. For each task, we randomly select the documents of 10 topics and split them into training/validation/test subsets at the ratio of 6:2:2.

For each task, we trained, validated and tested an L2-regularized logistic regression (LR) classifier, which is implemented with the scikit-learn toolkit [31], which is an open-source Python module integrating many state-of-the-art machine learning algorithms.

V. EXPERIMENTAL RESULTS

A. The Results on Analogical Reasoning

We test our models and baselines on this datasets mentioned above. For Chinese, the results are displayed in Table III. For English, the results are displayed in Table IV.

B. The Results on Word Similarity

Word similarity is conducted to test the semantic information encoded in word embeddings, and the results are presented in Table I. From the table, we can observe that our models outperform the compared baselines on nine out of ten datasets.

From the evaluation results on Chinese WS-240, we observe that: (1) Equal Synonymy Model, SSM-W and SSM-M significantly outperform baseline methods on both 233 word pairs and 240 word pairs, which validates the effectiveness of exploiting the internal synonymy between words; (2) SSM-W and SSM-M perform better than Equal Synonymy Model, which indicates that corresponding synonyms should have different impacts to the target word; (3) The addition of 7

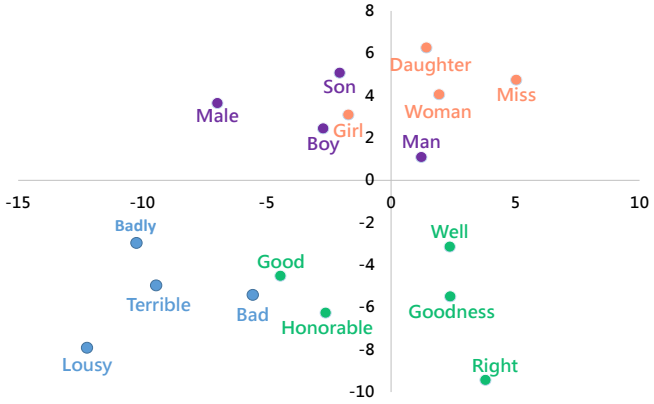

Fig. 5. The visualization of CBOW word embeddings.

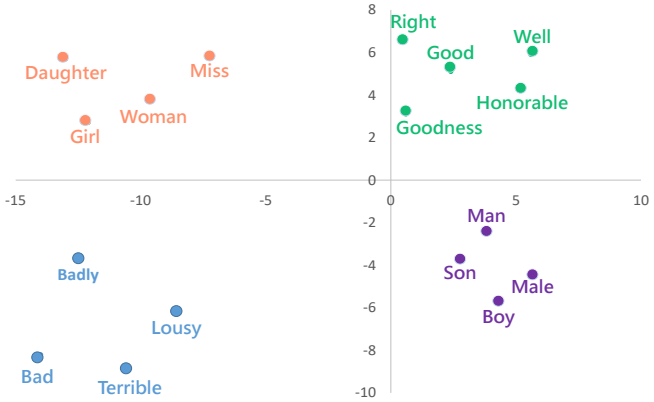

Fig. 6. The visualization of SSM-W word embeddings.

word pairs with new words does not cause significant change of correlations for all of above methods. The reason is that, the 7 word pairs are mostly unrelated. The default setting of 0 in all methods is basically consistent with the fact.

From the evaluation results on Chinese WS-296, we observe that the performance of baseline methods drop dramatically when adding 16 word pairs of new words, while the performance of Equal Synonymy Model and SSMs keeps stable. The reason is that the baseline methods cannot handle these new words appropriately. For example, "tiger" and "jaguar" are semantically relevant, but the relatedness is set to 0 in baseline methods simply because "jaguar" does not appear in the corpus, resulting in all baseline methods putting the word pair much lower than where it should be. In contrast, Equal Synonymy Model and SSMs compute the semantic relatedness of these word pairs much closer to human judgements. Since even if "jaguar" does not appear in the corpus, it's synonyms such as "tiger" do. Equal Synonymy Model and SSMs can easily cover at least one synonym of these new words and provide useful information about their semantic meanings for computing the relatedness.

From most of the English evaluation results, we observe that our SSMs get better score than baselines, which indicates

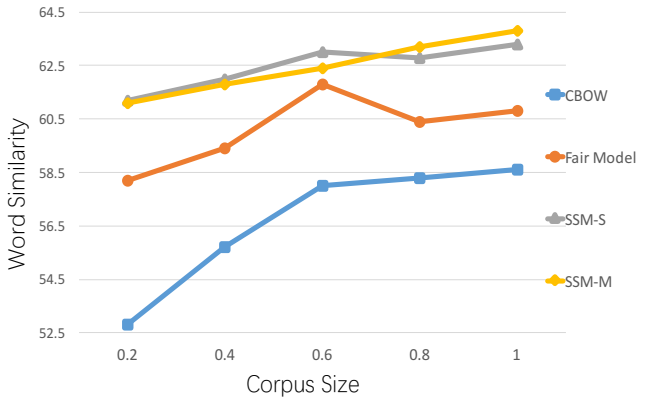

Fig. 7. Parameter analysis of corpus size. X-axis denotes the ratio of tokens used for training, and Y-axis denotes the Spearman rank (%) of word similarity.

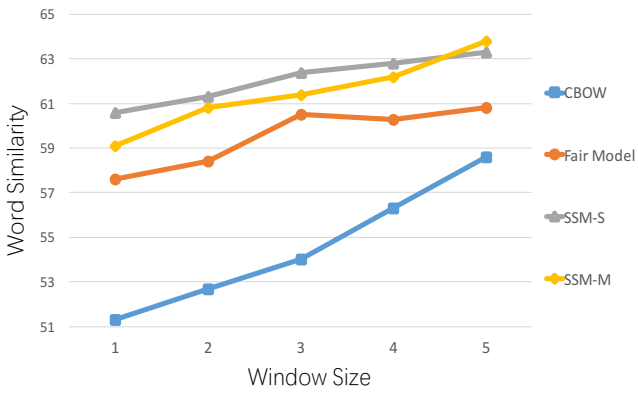

Fig. 8. Parameter analysis of window size. X-axis and Y-axis denote the window size and Spearman rank (%) of word similarity, respectively.

that our models can perform well in different languages. The reason of the improvement of SSMs on the evaluation of English and Chinese are generally the same. So We do not analyze the English evaluation results in more detail due to the page limits.

From the table III and table IV, we can observe that all of the synonymy-based model outperforms the baselines. Furthermore, it should be noted that our SSMs also achieve a better score than Equal Synonymy Model, which confirms the effectiveness of our intuitive idea again. In qualitative study, we found that Equal Synonymy Model are largely confused by polysemous words. For example, both "fu qin" and "lao zi" can be referred to the father in Chinese, but "lao zi" will also be used for someone calling himself arrogantly. If "lao zi" is assigned the same weight as other synonyms of "fu qin", we will get an inaccurate result in model training.

C. The Results on Text Classification

For Chinese and English text classification tasks, we report the classification accuracy over the test set. The average classification accuracy across the two tasks is utilized as the evaluation metric for different models. The results are

presented in Table V. Since we simply use the average embedding of words as the feature vector for classification, the overall classification accuracy of all models are merely around 80%. However, the classification accuracy of our SSMs still outperform all the baselines, especailly the Equal Synonymy Model. Moreover, it indicates that incorporating similarity of synonyms into word embeddings can contribute to enhancing the performance of downstream NLP tasks.

Note that language models like BERT [32] achieved great performance on downstream tasks. We also explore whether our model can make an improvement to BERT. As BERT for Chinese are only trained by char, we only do the experiment on English. We concatenate our models' embedding with BERT's embedding. To make a fair comparison, we also concatenate BERT's embedding with CBOW's embedding of the same dimension. As show in the last four rows of Table V, BERT+SSMs also get slightly better scores than original BERT and BERT+CBOW, which indicates that SSMs can integrate the knowledge from synonyms that BERT can not captured completely. Incorporating the idea behind SSMs into BERT directly is a valuable direction of future work.

D. Word Embedding Visualization

To visualize the embeddings of our models, we select several pairs of synonyms from the results of CBOW and SSM-W. The dimensions of the selected word embeddings are reduced from 200 to 2 using Principal Component Analysis (PCA). As show in Fig 5, some synonyms are not close enough, such as "Good" and "Well", whose meanings are similar. Some antonyms are embedded together, such as "Bad" and "Good", which are absolute opposite meaning. In our model, SSM-W, as show in Fig 6, synonyms get similar embedding, and antonyms are separated into different space. The visualization shows that embedding generated by our models can capture more accurate features of words, especially in the sense of semantics of synonyms.

E. The Impacts of Parameter Settings

Parameter settings can affect the performance of word embeddings. We analyze the impacts of corpus size and window size on the performance of word embeddings. In the analysis of corpus size, we hold the same parameter settings as before. The sizes of tokens used for training are separately 20%, 40%, 60%, 80% and 100% of the entire corpus mentioned above. We utilize the result of word similarity on Wordsim-296 as the evaluation criterion. From Fig. 7, we have the following observations. Firstly, the performance of our SSMs are better than CBOW and Equal Synonymy Model at each corpus size. Secondly, the performance of CBOW and Equal Synonymy Model are sensitive to the corpus size. In contrast, our SSMs' performance are more stable than others'. As we have shown in word similarity experiment, SSMs are able to increase the semantic information of word embeddings. It is worth noting that the worst performance of SSMs are nearly equal to the best performance of Equal Synonymy Model's. In the experiment with different window sizes, we observe that the performance of all word embeddings trained by different models has a trend to ascend with the increasing of window size as illustrated in Fig. 8. Our SSMs outperform others under all the preset conditions. Also, the worst performance of SSMs is nearly equal to the best performance of Equal Synonymy Model's.

VI. Conclusion

In this paper, we explored a new direction to employ the similarity of synonyms to train word embeddings. Different from previous works, we assume all corresponding synonyms make different impacts to the target word, so that we build our models based on attention scheme to assign different weights to corresponding synonyms. Two specific models named SSM-W and SSM-M are proposed by modifying the input layer and update rules of CBOW. To test the performance of our models, we chose CBOW, Skip-Gram and implemented a previous synonymy-based model as comparative baselines. We tested them on two basic NLP tasks of similarity and analogical reasoning, and downstream text classification tasks. The experimental results show that our models outperform the baselines on three tasks for both Chinese and English. In the future, we plan to incorporate SSMs with language models and perform evaluation on more downstream tasks.

VII. Ackowlegement

This research was supported by the National Key Research And Development Program of China (No. 2019YFB1405802)

References

[1] Liu Y, Liu Z, Chua T S, et al. Topical word embeddings[C]//Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015.

[2] Schütze H, Manning C D, Raghavan P. Introduction to information retrieval[C]//Proceedings of the international communication of association for computing machinery conference. 2008, 4.

[3] Shin B, Lee T, Choi J D. Lexicon integrated cnn models with attention for sentiment analysis[J]. arXiv preprint arXiv:1610.06272, 2016.

[4] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[5] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[6] Harris Z S. Distributional structure[J]. Word, 1954, 10(2-3): 146-162.

[7] Firth J R. A synopsis of linguistic theory, 1930-1955[J]. Studies in linguistic analysis, 1957.

[8] Miller G A. WordNet: a lexical database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.

[9] Yu M, Dredze M. Improving lexical embeddings with semantic knowledge[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2014: 545-550.

[10] Bian J, Gao B, Liu T Y. Knowledge-powered deep learning for word embedding[C]//Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, Heidelberg, 2014: 132-148.

[11] Zhang J, Salwen J, Glass M, et al. Word semantic representations using bayesian probabilistic tensor factorization[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014: 1522-1531.

[12] Ono M, Miwa M, Sasaki Y. Word embedding-based antonym detection using thesauri and distributional information[C]//Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2015: 984-989.

[13] Hasegawa M, Kobayashi T, Hayashi Y. Incorporating visual features into word embeddings: A bimodal autoencoder-based approach[C]//IWCS 2017—12th International Conference on Computational Semantics—Short papers. 2017.

[14] Alsuhaibani M, Bollegala D, Maehara T, et al. Jointly learning word embeddings using a corpus and a knowledge base[J]. PloS one, 2018, 13(3): e0193094.

[15] Sun M, Chen X, Zhang K, et al. Thulac: An efficient lexical analyzer for chinese[R]. Technical Report. Technical Report, 2016.

[16] Kim Y, Jernite Y, Sontag D, et al. Character-aware neural language models[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[17] Che W, Li Z, Liu T. Ltp: A chinese language technology platform[C]//Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations. Association for Computational Linguistics, 2010: 13-16.

[18] Levy O, Goldberg Y, Dagan I. Improving distributional similarity with lessons learned from word embeddings[J]. Transactions of the Association for Computational Linguistics, 2015, 3: 211-225.

[19] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.

[20] Dhillon P S, Foster D P, Ungar L H. Eigenwords: Spectral word embeddings[J]. The Journal of Machine Learning Research, 2015, 16(1): 3035-3078.

[21] Chen X, Xu L, Liu Z, et al. Joint learning of character and word embeddings[C]//Twenty-Fourth International Joint Conference on Artificial Intelligence. 2015.

[22] Finkelstein L, Gabrilovich E, Matias Y, et al. Placing search in context: The concept revisited[J]. ACM Transactions on information systems, 2002, 20(1): 116-131.

[23] Rubenstein H, Goodenough J B. Contextual correlates of synonymy[J]. Communications of the ACM, 1965, 8(10): 627-633.

[24] Luong T, Socher R, Manning C. Better word representations with recursive neural networks for morphology[C]//Proceedings of the Seventeenth Conference on Computational Natural Language Learning. 2013: 104-113.

[25] Huang E H, Socher R, Manning C D, et al. Improving word representations via global context and multiple word prototypes[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, 2012: 873-882.

[26] Bruni E, Tran N K, Baroni M. Multimodal distributional semantics[J]. Journal of Artificial Intelligence Research, 2014, 49: 1-47.

[27] Agirre E, Alfonseca E, Hall K, et al. A study on similarity and relatedness using distributional and wordnet-based approaches[C]//Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2009: 19-27.

[28] Spearman C. The proof and measurement of association between two things[J]. 1961.

[29] Mikolov T, Yih W, Zweig G. 11111[C]//Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013: 746-751.

[30] Peng H, Ma Y, Li Y, et al. Learning multi-grained aspect target sequence for Chinese sentiment analysis[J]. Knowledge-Based Systems, 2018, 148: 167-176.

[31] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python[J]. Journal of machine learning research, 2011, 12(Oct): 2825-2830.

[32] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.