Improving word vector model with part-of-speech and dependency grammar information

Chunhui Deng¹, Gangming Lai², Huifang Deng²

¹School of Computer Engineering, Guangzhou College of South China University of Technology, No. 1 Xue Fu Road, Huang, Guangzhou 510800, People’s Republic of China
²School of Computer Science and Engineering, South China University of Technology, University Town, Panyu, Guangzhou 510006, People’s Republic of China
✉ E-mail: hdengpp@qq.com

Abstract: Part-of-speech (POS) and dependency grammar (DG) are the basic components of natural language processing. However, current word vector models have not made full use of both POS information and DG information, and hence the models’ performances are limited to some extent. The authors first put forward the concept of POS vector, and then, based on continuous bag-of-words (CBOW), constructed four models: CBOW + P, CBOW + PW, CBOW + G, and CBOW + G + P to incorporate POS information and DG information into word vectors. The CBOW + P and CBOW + PW models are based on POS tagging, the CBOW + G model is based on DG parsing, and the CBOW + G + P model is based on POS tagging and DG parsing. POS information is integrated into the training process of word vectors through the POS vector to solve the problem of the POS similarity being difficult to measure. The POS vector correlation coefficient and distance weighting function are used to train the POS vector as well as the word vector. DG information is used to correct the information loss caused by fixed context windows. Dependency relations weight is used to measure the difference of dependency relations. Experiments demonstrated the superior performance of their models while the time complexity is still kept the same as the base model of CBOW.

1 Introduction

Word representation is the basic task of natural language processing (NLP). To simplify the processing, traditional NLP tasks use a one-hot representation method to represent words [1]. The one-hot representation uses high-dimensional vectors to represent words of only one non-zero dimension, which cannot capture any grammatical or semantic information. Rumelhart et al. [2] proposed distributed word vector representation (WVR), also known as word embedding representation, which can capture grammatical and semantic information by representing words in dense and low-dimensional vector space. However, by in-depth examining these WVRs or models, one can still find several shortcomings of them: (i) the words are equally treated in the context which cannot distinguish the difference between words [3–8]; (ii) the fixed context window is used to predict target word, and thus resulted in the loss of some important words in long and compound sentences [3–10]; (iii) it is over simplified to only use word frequency to select words in sub-sampling and negative sampling, which are used to speed up the training process and improve the quality of word vector [3–10].

To address the above issues, in this paper, we try in word vectors to apply the part-of-speech (POS) tagging information and the dependency grammar (DG) parsing information, which is already available from traditional NLP tasks, to further improve the performance of the word vector model.

POS tagging refers to correctly tagging each word in the result of word segmentation, i.e. POS tagging is the process of determining the POS tag of each word, which could be a noun, verb, adjective, or other types. On the other hand, the basic task of DG parsing is to determine the syntactic structure of a sentence or the dependency relationship between words in a sentence. Doing DG parsing on the results of word segmentation and POS tagging can further reveal the dependency relationship between words. To make full use of POS tagging and DG parsing, we combine POS information and DG information into a word vector model based on the popular continuous bag-of-words (CBOW) model [3, 11, 12]. POS information refers to the lexical information that can help to determine the POS tag of a target word through a form of POS vector. The concept of the POS vector is put forward for the first time by us in this paper to measure the similarity between POSs and to improve the sub-sampling technology and negative sampling technology. On the other hand, DG information means the grammar information that is contained in the extended part of a given context, which is used to compensate for the information loss due to the fixed context window of the word vector model. The dependency relation weight, which can bring DG information into the word vector model, is used to distinguish the differences of dependency relations. Specifically, we proposed four types of models using POS information and DG information: CBOW + P and CBOW + PW models, which are based on POS tagging, CBOW + G model, which is based on DG parsing, and CBOW + G + P model, which is based on POS tagging and DG parsing, respectively. In CBOW + P and CBOW + PW models, we optimise the hidden layer’s representation of context by using the POS vector to optimise the prediction of the target word. The POS vectors correlation coefficient and distance weighting function are used to train the POS vectors as well as the word vectors together. In the CBOW + G model, we use DG information to obtain long-distance dependency word information and increase the weight of the words related to the target word in the context window. Adding sentence-based DG information to the training process of the word vector model can reduce the problem of information loss caused by a fixed-size context window. Furthermore, since there is no data set to measure the quality of the POS vector, we construct a POS analogy data set with 55 sets of test data to measure its effect. Experiments on word similarity, word analogy, and POS analogy tasks presented in this paper show that the proposed word vector models significantly outperform other relevant models while the time complexity is still...
kept under the same order of magnitude as the popular CBOW base model.

This paper is organised as follows: Section 2 is a brief review of the related works. Section 3 states the four models proposed by this paper. Section 4 gives the experimental results and comparisons to other models. Also, Section 5 concludes this paper.

2 Related works

In recent years, Mikolov et al. [3] proposed the CBOW model and Skip-gram model, which are simple, effective, and easy to train in the large-scale corpus. Therefore, these models are popular in NLP tasks, such as text classification [13, 14], named entity recognition [15, 16], and word segmentation tasks [17, 18]. Turian et al. [19] classified distributed WVR into three categories: (i) matrix-based, (ii) clustering-based, and (iii) neural network-based. Matrix-based distributed WVR uses a matrix to represent a word by constructing a word-context matrix. Each row in the matrix corresponds to a word, each column corresponds to a context, and the elements in the matrix correspond to the co-occurring times of words and context. Clustering-based distributed WVR uses the clustering method to represent a word using the Huffman code. Neural network-based distributed WVR captures semantic association between words through multi-layer neural networks which can represent the complex context. Therefore, the neural network-based distributed WVR becomes the mainstream technology of WVR.

In this paper, we further divide the neural network-based distributed WVR into four types according to the different information incorporated into the model: (i) basic distributed WVR, (ii) distributed WVR with word structure information incorporated, (iii) distributed WVR with external knowledge base incorporated, and (iv) distributed WVR with grammatical information incorporated. Type 1 uses context information as the only training corpus, such as skip-gram [3], CBOW [3, 11–12], and GloVe [13] models. Type 2 incorporates structural information of words, including characters, radicals, and strokes. Chen et al. [5] proposed a character and word embeddings model that considers the joint training of Chinese characters and words and takes the character vector as part of the word vector. Yin et al. [6] proposed to add the target word’s radicals to the context representation and then used an improved CBOW model to train the words, characters, and radicals jointly. Cao et al. [7] proposed the context vector model to extract the strokes’ N-gram features of words and then used them to train in the model instead of words. Type 3 attempts to formalise the semantics information contained in the internal structure of words and then integrates them into the training process. Chen et al. [5] proposed the concept of sense vector, which divides polysemy into different sense vectors and improves the effect of word vector in English disambiguation task. Niu et al. [8] proposed Semeame attention over target model and other models by using sememes of HowNet to improve the WVR. They first used the attention mechanism to add sememes of HowNet to WVR learning [20]. Type 4 provides overall semantic information for word representation by lexical analysis and grammatical analysis of traditional NLP. In a sense, types 1–3 WVRs that fuse word structure information or external knowledge base belong to the way of expanding corpora to improve the quality of word vectors. While type 4 focuses on using traditional NLP’s lexical and grammatical analysis, which can reflect the dependency of POS information at the sentence level and provide semantic information for WVRs without any other expanded corpora needed. POS tagging and DG parsing are important parts of traditional NLP’s lexical and grammatical analysis.

To improve the expressive capability of word embedding, Liu et al. [9] and Pan et al. [10] tried incorporating POS into word embedding. Levy and Goldberg [21] first suggested that DG should be integrated into the training process of the CBOW model. POS information at the sentence level provides semantic information for learning word embeddings (PWE) model proposed by Liu et al. [9] and CWindow-POS (CWP) model proposed by Pan et al. [10] cannot make full use of POS information, as these models only modify words’ weight of the context window by just using the POS correlation weighted matrix. The dependency-based word embeddings (DBWE) model proposed by Levy and Goldberg [21] does not consider the diversity of dependency relations and limits the scope of DG to the context window. Therefore, we propose four types of models: CBOW + P, CBOW + PW, CBOW + G, and CBOW + G + P models to make use of POS information and DG information, which are already obtained in traditional NLP tasks, to enhance the performance of the word vector model.

3 Methodology

In this paper, the CBOW + P and CBOW + PW models based on POS tagging are proposed and the concept of POS vector is first formally put forward as far as we know. POS vector is introduced to formalise the POS information. The POS vector and the word vector are defined in the same vector space and trained in the same way so that the context’s POS information can help to predict the target word. On the other hand, we propose the CBOW + G model based on DG parsing and two strategies: (i) pre-training cosine distance (PTCD) and distiributed cosine similarity (DSCD) to integrate DG information into word vectors. The CBOW + G + P model is a combination of CBOW + G and CBOW + P models.

3.1 CBOW + P model

The PWE model [9] modifies the words’ weight of the context window by using the POS correlation weighted matrix, which is related to the distance between the target word and context words. Inspired by the PWE model, in this paper, we propose the CBOW + P model based on POS tagging, which further integrates POS information into the training process of word vectors and proposes the concept of POS vector to solve the problem of difficult measurement of POS similarity. On the other hand, we assume that the contribution of POS to the prediction of the target word is inversely proportional to context distance. We use a function that has a positive correlation with context distance to replace the POS correlation weighted matrix of the PWE model and, in this way, we can also reduce the training parameters. To measure the effectiveness of the POS vector, similar to the task of word analogy, we construct a data set to verify the effect of the POS vector. Surprisingly, the POS vector also exhibits analogical natures, such as $vn - v = a = an$. Hence, we call this data set a POS analogy test set.

The framework of the CBOW + P model is shown in Fig. 1. The CBOW + P model is a three-layer feed forward neural model, consisting of input layer, hidden layer, and output layer. The input layer is used to accept the context’s word vector and POS vector, which are both expressed by one-hot vector. The dimension of word vector is equal to the vocabulary size of the corpus, which is denoted as $|V|$, while the dimension of the POS vector is equal to the vocabulary size of the POS set, which is denoted as $|E|$. So, the input layer’s dimension is $4C \times (|V| + |E|)$, where $C$ is the windows size of context. The hidden layer represents the latent semantics of the context in the form of low-dimensional vectors, which is denoted as $N$, ranging from 50 to 300. The output layer represents the target word in the form of one-hot vector, with a dimension of $(|V| + |E|)$. The matrix dimensions between the output layers and the hidden layer are the parameters of the model. The POS tagging set in this paper follows the people’s daily annotated corpus (PFR) People’s Daily Annotation Corpus standard of the Institute of Computational Linguistics, Peking University [22]. We define $E$ as the POS set, $|E|$ as the size of the POS set, $x_i$ as the $i$th target word, $x_{i+k}$ as the $(i-k)$th and $(i+k)$th word of $x_i$ in the context window of length $k$, respectively, index $k(0 \leq k \leq l)$ ($C$ is the embedding size of context) as the distance between word $x_i$ and word $x_j$, $q(k)$ as the function, which has positive correlation with context distance $k$, $z_{i+k}(\in E)$ and $z_{j+k}(\in E)$ as the POS
vectors of \( x_{i-4} \) and \( x_{i+4} \), respectively, \( r_{z,z} \) as the correlation coefficient of the POS vector of word, which is initialised with an assigned random number between 0 and 1 and trained with the model. Also, when \( r_{z,z} = 0 \), the CBOW + P model reduces to the CBOW model.

The CBOW model ignored the POS information’s effect in predicting the target word. The word’s POS in the context of a target word determines the POS and part of the semantics of the target word. It is natural to assume that the farther away from the target word, the less the effect of the POS, i.e. the decisive effect of POS is inversely proportional to the distance \( k \) between the target word and the context word. The POS vector is added to the objective function \( L \) of the CBOW + P model, which regards POS information as part of the context of the target word, as shown in (1)

\[
L = \sum_{x_i \in D} \log P(x_i|x_{i-C}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+C})
\]

(1)

\[
\sum_{x_i \in D} \log P(x_i|x_{i-C}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+C})
\]

where \( z_{i-C}, \ldots, z_{i-1}, z_{i+1}, \ldots, z_{i+C} \) denote the POS vectors corresponding to the context words \( x_{i-C}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+C} \). \( P(x_i|x_{i-C}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+C}) \) represents the probability that the target word is \( x_i \) when the context word sequence is \( x_{i-C}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+C} \). Differing from the context representation of the CBOW model, the CBOW + P model adds POS vector \( z_{i-k} \), POS vector correlation coefficient \( r_{z,z} \), and distance weighting function \( \varphi(k) \) to the context representation. The calculation formula for the hidden layer variable (vector) of \( h \) is given in (2)

\[
h = \frac{1}{2C} \sum_{-C \leq k \leq C, k \neq 0} \frac{r_{z,z} + w_{i-k}}{\varphi(k)}
\]

(2)

Statistics on the frequency of POS appearing in Chinese Wikipedia (i.e. training corpus) show that the occurrence probability of each POS type is not equal. The frequency of POS being found in the training corpus can reflect the common extent of words, which can be used as another indicator in sub-sampling to calculate the drop probability \( Q(x_i) \) of word \( x_i \). Therefore, we propose that both the frequency of words being found in the training corpus and the frequency of POS should be considered in the sub-sampling to measure the common extent of words. The drop probability \( Q(x_i) \) of sub-sampling is shown in (3)

\[
Q(x_i) = 1 - \frac{t}{\sqrt{f(x_i) \cdot [g(z_i)]^{1/4}}}
\]

(3)

where \( f(x_i) \) represents the frequency of word \( x_i \) appearing in the training corpus; \( t \) is the preset threshold, which is generally set to \( 10^{-3}-10^{-6} \); \( g(z_i) \) is the frequency of POS \( z_i \) being found in the training corpus, which is called POS proportion, corresponding to the word \( x_i \). Its definition is given in (4), where \( E_z \) is the set of words with the same type of POS in POS set \( E \), the power of 1/4 is a trade-off value to adjust the impact of POS frequency

\[
g(z_i) = \frac{f(x_i)}{\sum_{x_i \in E_z} f(x_i)}
\]

(4)

On the other hand, the CBOW + P model also uses POS to optimise the calculation of negative sampling. When choosing negative samples, we will consider the impact of word frequency and POS proportion, CBOW + P model prefers to choose words with high frequency and high POS proportion. The calculation of the probability \( P(x_i) \) that words are selected as negative samples is given in (5)

\[
P(x_i) = \frac{\left( f(x_i) \cdot [g(z_i)]^{1/4} \right)^{3/4}}{\sum_{x_i \in E_z} \left( f(x_i) \cdot [g(z_i)]^{1/4} \right)^{3/4}}
\]

(5)

\[
P(x_i) = \frac{\left( f(x_i) \cdot [g(z_i)]^{1/4} \right)^{3/4}}{\sum_{x_i \in E_z} \left( f(x_i) \cdot [g(z_i)]^{1/4} \right)^{3/4}}
\]

3.2 CBOW + PW model

Based on the CBOW + P model, we further propose the CBOW + PW model. The difference is that the CBOW + P model takes POS as studying the object, while the CBOW + PW model takes a word as studying the object. The CBOW + PW model assumes that each word has its correlation coefficient of POS vector, which further refines the influence of POS in WVR. The CBOW + PW model assumes that each word in the vocabulary has its correlation coefficient of POS vector. Specifically, the difference between the CBOW + P model and the CBOW + PW model lies in the context representation of the target word. The calculation formula for the hidden layer variable (vector) of \( h \) is given in (6)

\[
h = \frac{1}{2C} \sum_{-C \leq k \leq C, k \neq 0} \frac{r_{w,z} + w_{i-k}}{\varphi(k)}
\]

(6)

where \( r_{w,z} \) represents the correlation coefficient of the POS vector corresponding to each word that is obtained in the training process, replacing \( r_{z,z} \) of the CBOW + P model. The definitions of other symbols are as stated for (1), not repeated here. The framework of the CBOW + PW model is shown in Fig. 2. The CBOW + PW model is also a three-layer feed forward neural model, consisting of the input layer, hidden layer, and output layer, the same as the CBOW + P model. The CBOW + PW model also used the improved sub-sampling and negative sampling technologies that the CBOW + P model used, as shown in (3) and (5).

3.3 CBOW + G model

The CBOW + P and CBOW + PW models use POS information to predict the target words. In traditional NLP, after POS tagging, DG parsing is often carried out to further reveal the diversity of the dependency relationship between words. Therefore, in this paper, we improve the word vector model further based on DG parsing to correct the information loss caused by the current fixed context window approach. At the same time, dependency weight,
which is introduced to represent the differences of dependency relations, is used to integrate DG information into the word vector model. The CBOW + G model proposed in this paper is also a three-layer model (as shown in Fig. 3), but with the dependency relations between words added in the input layer through DG parsing. The hidden layer and output layer are the same as CBOW + P and CBOW + PW.

The model proposed by Levy and Goldberg [21] does not consider the diversity of dependency relations and limits the scope of DG to the context window. In the CBOW + G model, DG information plays two main roles: (i) DG information takes a sentence as a unit rather than a context window as a unit. While a Chinese document usually consists of complex long sentences and compound sentences, too short or too long a context window will affect the effect of the model. Adding sentence-based DG information to the training process of the word vector model can capture long-distance dependency information and hence reduce the impact of the fixed-size context window. (ii) DG information improves the relative weight of the words related to the target word in the context window. Dependency relation exhibits an effect that is similar to the attention mechanism prevailing in recent years. We use the DG annotation system of the Language Technology Platform (LTP) of Harbin University of Technology [23].

We define the dependency relation set \( Q \) of the LTP DG annotation system as follows:

\[
Q = \{ \text{SBV, VOB, IOB, FOB, ATT, ADV, CMP, COO, POB, LAD, RAD, IS, WP, HED} \}
\]

There are 15 kinds of dependency relations of \( Q \). The formulation of the objective function is similar to that of the CBOW model, but the difference lies in the context representation, as shown in (7)

\[
r = \frac{1}{2C} \sum_{-C \leq k \leq C, k \neq 0} w_{i-k} \cdot \sum_{\text{Relation}(w_i, w_j) = Q_{m} \in J} r_{Q_{m}} \cdot w_j
\]

where \( k \) is an integer window-variable bounded by windows size of \( C \); \( w_{i-k} \) represents the \((i-k)\)th word of the context; \( \text{Relation}(w_i, w_j) \) represents the dependency relation between the \( j \)th target word \( w_j \) and its \( j \)th dependent word \( w_i \); \( J = \{ \text{Relation}(w_i, w_j) \mid \text{Relation}(w_i, w_j) \} \) is the dependency relation sequence of target word \( w_j \); \( Q_{m} \) is the \( m \)th relation of the dependency relation set \( Q \); \( r_{Q_{m}} \) is the dependency parameter of the target word \( w_j \) and its dependency word, which is used to distinguish the influence of different dependency relations. We define \( r_{Q_{m}} \) as the average cosine distance of a pair of words belonging to the same dependency relationship in the corpus. We propose two computational strategies: PTCD and NSCD. The calculation formula of the PTCD strategy is shown as

\[
r_{Q_{m}} = \frac{1}{|C_{m}|} \sum_{\text{Relation}(w_i, w_j) = Q_{m} \in C_{m}} \cos(w_j, w_i)
\]

where \( C_{m} \) is all pairs of words belonging to the same dependency category \( Q_{m} \) in the corpus; \( |C_{m}| \) is its size; \( \{ w_i, w_j \} \) is a pair of words in \( C_{m} \). On the other hand, the NSCD strategy uses the average cosine distance of all pairs of words belonging to the same dependency relationship to calculate the dependency weight \( r_{Q_{m}} \) as

\[
r_{Q_{m}} = \frac{1}{K_{m}} \sum_{\text{Relation}(w_i, w_j) \in \text{NEG(G(Q_{m}))}} \cos(w_j, w_i)
\]

where \( K_{m} = |\text{NEG(G(Q_{m}))}| \) represents the size of a negative sampling set of dependency relation; \( \{ w_i, w_j \} \) is a pair of words in \( \text{NEG(G(Q_{m}))} \); \( \text{Relation}(w_i, w_j) \) denotes the dependency relation between words \( \{ w_i, w_j \} \). We propose the CBOW + G model to capture the long-distance DG information and increase the proportion of words that are dependent on target words in context representation.

Take window size = 2 as an example. ‘record (记录)’ and ‘people (人)’ are not in the context window of the target word ‘study (研究)’. However, there is a coordinate (COO) relationship between ‘record (记录)’ and the target word ‘study (研究)’. DG parsing finds words that are closely related to the target word outside the context window, which is of great value in predicting the target word. On the other hand, different dependencies play different roles. For example, ‘、’ and ‘和 (and)’ have little value in predicting the target word ‘study (研究)’. To solve this problem, we pre-train the word vector and use NSCD or PTCD strategies to calculate \( r_{Q_{m}} \). For example, the average cosine distances of

![Fig. 2 CBOW + PW model](image-url)

![Fig. 3 CBOW + G model](image-url)
attribute and COO relations trained in Chinese Wikipedia are close to 0.5, while the average cosine distances of right adjunct and left adjunct are close to 0.

### 3.4 CBOW + G + P model

The CBOW + G + P model is a hybrid of the CBOW + P model and CBOW + G model. Experiments show that the CBOW + P model performs better in word similarity tasks, while the CBOW + G model does better in word analogy tasks. Therefore, we integrate the CBOW + P and CBOW + G models into the CBOW + G + P model, by taking their respective advantages. The calculation formula expressed in the hidden layer vector $h$ is as follows, which follows the same formalism as for the CBOW + P model and CBOW + G model:

$$
\begin{align*}
    h &= \frac{1}{2C} \sum_{-C \leq s \leq C, s \neq 0} w_{i-s} + \frac{1}{|J|} \sum_{\text{Relation}(w_{i}, j) = Q_{e} \in J} \rho_{Q_{e} w} \cdot w_{j} \\
    &+ \frac{1}{2C} \sum_{-C \leq s \leq C, s \neq 0} r_{z_{i-s}} \cdot z_{j-s},
\end{align*}
$$

Where $C$ is the window size, $J$ is the set of relations, and $Q_{e}$ is the entity.

### 4 Experiments

We quantitatively evaluate the quality of word vectors learned by CBOW + P, CBOW + PW, CBOW + G, and CBOW + G + P models on word similarity task, word analogy task, and POS analogy task proposed in this paper.

#### 4.1 Experiment settings

**Training corpus:** Chinese Wikipedia is used as the training corpus of this paper. In the preprocessing, LTP [23] and Jieba [24] are used for Chinese word segmentation, POS tagging, and DG parsing. After preprocessing, we obtained a 3.16 GB training corpus and 73,174,132 words. The vocabulary size is 1,581,744. The POS tagging standard is a POS tagging set of the PFR People’s Daily annotation corpus, which contains 48 POS tags, published by the Institute of Computational Linguistics, Peking University. Based on this POS tagging set, we propose a POS analogy test set with 55 sets of test data.

#### 4.2 Word similarity experiment

Word similarity task evaluates the capability of word vectors to reveal the semantic relevance of words. In this paper, two sets of Chinese word similarity data sets, Wordsim-240 and Wordsim-297, provided by Chen et al. [5] are selected for evaluation. Wordsim-240 data set contains 240 pairs of Chinese words and Wordsim-297 data set contains 297 pairs of Chinese words. A word pair of ‘OPEC–石油’ not appearing in Chinese Wikipedia is deleted from Wordsim-297 to become Wordsim-296. The similarity score of a pair of words is calculated by the cosine similarity of word vectors. The Spearman correlation coefficient proposed by Myers and Sirolli [25] is used in this paper to measure the difference between the human marker’s scores and similarity scores calculated by word embedding evaluation. The experimental results are shown in Tables 1 and 2, where +P denotes the CBOW + P model, +PW denotes the CBOW + PW model, +G denotes the CBOW + G model, and +G + P denotes the CBOW + G + P model.

The results show that the +P and +PW models proposed in this paper are superior to CBOW, PWE, and CWP models in word similarity task.
similarity tasks with the improvement rising up to 3.25%. On the other hand, the +G and +G+P models outperform CBOW and DBWE models in word similarity tasks with improvement rising up to 8.08%.

### 4.3 Word analogy experiment

This task evaluates the quality of word vectors by examining their capability of discovering linguistic rules between pairs of words. The data set proposed by Chen et al. [5], which is called the Analogy-1125 data set in this paper, includes three types of analogies: national capital, city state/province, and family vocabulary, with a total of 1125 analogies. As there is no Chinese analogy data set of POS vectors as far as we know, based on the POS tagging set of PFR People’s Daily annotation corpus, in this paper, we propose a data set consisting of 55 test data sets, which is called POS-Analogy-55 data set. (v, vn, a, an), (a, ad, g, dg), and (n, vn, v) are examples of the POS-Analogy-55 data set. The experimental results are shown in Tables 3 and 4.

The results show that the +P and +PW models proposed in this paper are superior to CBOW, PWE, and CWP models in the POS-Analogy-55 data set with the improvement rising up to 83.32%, but the performance on the Analogy-1125 data set is slightly inferior. On the other hand, the +G and +G+P models outperform CBOW and DBWE models with the improvement rate rising up to even 166.74% in the POS-Analogy-55 data set, and the improvement rate rising up to 3.01% in the Analogy-1125 data set, as shown in Table 4.

It can be seen from Tables 1-4 that the CBOW + P, CBOW + PW, CBOW + G, and CBOW + G + P models perform better than the original CBOW model. Since CBOW, PWE, and CWP do not output POS vectors, we use the average value of all word vectors belonging to the same type of POS as the vectors. The four proposed models can improve the quality of word vectors by using both POS information and DG information. PWE and CWP models only use POS weight information between the target word and context words. The POS vectors proposed in this paper are located in the same vector space as word vectors and formally represent the impact of POS in word meaning. The time complexity of the four models proposed in this paper is of the same magnitude as CBOW, and the number of predictions is O(V), where V is the size of the vocabulary. POS tagging and DG parsing are conducted in preprocessing.

### 5 Conclusions

In this paper, we proposed four types of models using POS information and DG information: CBOW + P, CBOW + PW, CBOW + G, and CBOW + G + P models. CBOW + P and CBOW + PW models optimise the vector representation of context information. CBOW + G model uses DG information to obtain long-distance dependency word information and increase the weight of the words related to the target word in the context window. To verify the effectiveness of the POS vector, we proposed a POS analogy data set consisting of 55 groups of test data. Experiments show that the proposed model outperforms the PWE and CWP model when using POS information and it also outperforms the DBWE model when taking into account the DG information. In addition, the time complexity of the proposed four models is of the same magnitude as the CBOW model. Future work may need to explore the following aspects: first, since the POS vector proposed in the paper presents its capability of discovering linguistic rules between word pairs, but how to apply this capability of it into the traditional NLP tasks in depth deserves further research. Second, dependency parameters are calculated directly from cosine distances, and perhaps more efficient methods should be used to calculate them. Third, a closer or deeper combination of POS information and DG information should be considered.

### 6 Acknowledgments

This work was supported in part by the Department of Education of Guangdong Province under Special Innovation Program (Natural Science), grant number 2015KTSCX183, and in part by the South China University of Technology under ‘Development Fund’ with fund number x2js-F8150310.
7 References

[1] Sun, F., Guo, J.-F., Lan, Y.-Y., et al.: ‘A survey on distributed word representation’, Chin. J. Comput., 2019, 42, (7), pp. 1605–1625

[2] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: ‘Learning representations by back-propagating errors’, Nature, 1986, 323, pp. 533–536

[3] Mikolov, T., Chen, K., Corrado, G., et al.: ‘Efficient estimation of word representations in vector space’, Workshop Track Proc. 1st Int. Conf. on Learning Representations, Scottsdale, AZ, USA, May 2013, pp. 1–12

[4] Pennington, J., Socher, R., Manning, C.D.: ‘Glove: global vectors for word representation’. Proc. 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP), Doha Qatar, October 2014, pp. 1532–1543

[5] Chen, X., Xu, L., Liu, Z., et al.: ‘Joint learning of character and word embeddings’. Proc. 24th Int Joint Conf. on Artificial Intelligence, Buenos Aires, Argentina, July 2015, pp. 1236–1242

[6] Yin, R., Wang, Q., Li, P., et al.: ‘Multi-granularity Chinese word embedding’. Proc. 2016 Conf. on Empirical Methods in Natural Language Processing, Austin, Texas, November 2016, pp. 981–986

[7] Cao, S., Lu, W., Zhou, J., et al.: ‘Cw2vec: learning Chinese word embeddings with stroke N-gram information’. Proc. 32nd AAAI Conf. on Artificial Intelligence, New Orleans, LA, USA, February 2018, pp. 5053–5061

[8] Niu, Y., Xie, R., Liu, Z., et al.: ‘Improved word representation learning with sememes’. Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, July 2017, pp. 2049–2058

[9] Liu, Q., Ling, Z., Jiang, H., et al.: ‘Part-of-speech relevance weights for learning word embeddings’, arXiv:1603.07695v1 [cs.CL], 2016

[10] Pan, B., Yu, C., Zhang, Q., et al.: ‘The improved model for word2vec based on part of speech and word order’, Acta Electron. Sin., 2018, 46, (8), pp. 1976–1982

[11] Continuous Bag of Words (CBOW), https://ikinc.online/tag/continuous-bag-of-words-cbow/, accessed 22 February 2020

[12] Wang, Q., Xu, J., Chen, H., et al.: ‘Two improved continuous bag-of-word models’. Proc. Int. Joint Conf. on Neural Networks, Anchorage, AK, USA, May 2017, pp. 2851–2856

[13] Lai, S., Xu, L., Liu, K., et al.: ‘Recurrent convolutional neural networks for text classification’. Proc. 29th AAAI Conf. on Artificial Intelligence, Austin, TX, USA, January 2015, pp. 2267–2273

[14] Zhang, X., Zhao, J., Leeun, Y.: ‘Character-level convolutional networks for text classification’. Advances in Neural Information Processing Systems – Proc. 29th Annual Conf. on Neural Information Processing Systems, Montreal, QC, Canada, December 2015, pp. 649–657

[15] Chi, J.P.C., Nicholls, E.: ‘Named entity recognition with bidirectional LSTM-CNN’. Trans. Assoc. Comput. Linguistics, 2016, 4, pp. 357–370

[16] Lample, G., Ballesteros, M., Subramanian, S., et al.: ‘Neural architectures for named entity recognition’, arXiv:1603.01360v3 [cs.CL], 2016

[17] Zhang, M., Zhang, Y., Fu, G.: ‘Transition-based neural word segmentation’. Proc. 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin Germany, August 2016, pp. 421–431

[18] Lee, J., Cho, K., Hofmann, T.: ‘Fully character-level neural machine translation without explicit segmentation’, Trans. Assoc. Comput. Linguistics, 2017, 5, pp. 365–378

[19] Turian, J., Ratinov, L., Bengio, Y.: ‘Word representations: a simple and general method for semi-supervised learning’. Proc. 48th Annual Meeting of the Association for Computational Linguistics, Uppsala Sweden, July 2010, pp. 384–394

[20] Bengio, Y., Courville, A., Vincent, P.: ‘Representation learning: a review and new perspectives’. IEEE Trans. Pattern Anal. Mach. Intell., 2013, 35, (8), pp. 1798–1828

[21] Levy, O., Goldberg, Y.: ‘Dependency-based word embeddings’. Proc. 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Long Papers), Baltimore, MD, USA, June 2014, pp. 302–308

[22] Yu, S.: ‘Foundational language resource data on web sites’, Terminology Stand. Inf. Technol., 2001, (4), pp. 19–23

[23] Che, W., Li, Z., Liu, T.: ‘LTP: a Chinese language technology platform’. Proc. 23rd Int. Conf. on Computational Linguistics: Demonstrations, Beijing China, August 2010, pp. 13–16

[24] Sun, J.: ‘Jieba (Chinese for ‘to stutter’) Chinese text segmentation’, https://github.com/fxsjy/jieba, accessed 20 February 2020

[25] Myers, L., Siros, M.J.: ‘Spearman correlation coefficients, differences between’, in Kotz, S., Read, C.B., Balakrishnan, N., et al.: ‘Encyclopedia of statistical sciences’ (John Wiley & Sons, Inc., USA, 2006), doi: 10.1002/0471667196.esc0505.pub2