

Multiplex Word Embeddings for Selectional Preference Acquisition

Hongming Zhang^{♣*}, Jiaxin Bai^{♣*}, Yan Song[♣], Kun Xu[♡],
Changlong Yu[♣], Yangqiu Song[♣], Wilfred Ng[♣], and Dong Yu[♡]

[♣]Department of CSE, The Hong Kong University of Science and Technology

[♣]Sinovation Ventures

[♡]Tencent AI Lab

{hzhangal, jbai, cyuaq, yqsong, wilfred}@cse.ust.hk
clksong@gmail.com, {kxkunxu, dyu}@tencent.com

Abstract

Conventional word embeddings represent words with fixed vectors, which are usually trained based on co-occurrence patterns among words. In doing so, however, the power of such representations is limited, where the same word might be functionalized separately under different syntactic relations. To address this limitation, one solution is to incorporate relational dependencies of different words into their embeddings. Therefore, in this paper, we propose a multiplex word embedding model, which can be easily extended according to various relations among words. As a result, each word has a center embedding to represent its overall semantics, and several relational embeddings to represent its relational dependencies. Compared to existing models, our model can effectively distinguish words with respect to different relations without introducing unnecessary sparseness. Moreover, to accommodate various relations, we use a small dimension for relational embeddings and our model is able to keep their effectiveness. Experiments on selectional preference acquisition and word similarity demonstrate the effectiveness of the proposed model, and a further study of scalability also proves that our embeddings only need 1/20 of the original embedding size to achieve better performance.

1 Introduction

Representing words as distributed representations is an important way for machines to process lexical semantics, which attracts much attention in natural language processing (NLP) in the past few years (Mikolov et al., 2013; Pennington et al., 2014; Song et al., 2017, 2018b; Song and Shi, 2018) with respect to its usefulness in many downstream tasks, e.g., parsing (Chen and Manning, 2014), machine translation (Zou

et al., 2013), coreference resolution (Lee et al., 2018), etc. Conventional word embeddings, e.g., word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), leverage the co-occurrence information among words to train a unified embedding for each word. Such models are popular and the resulting embeddings are widely used owing to their effectiveness and simplicity. However, these embeddings are not helpful for scenarios requiring words functionalizing separately under different situations, where selectional preference (SP) (Wilks, 1975) is a typical scenario.

In general, SP refers to that, given a word (predicate) and a dependency relation, human beings have certain preferences for the words (arguments) connecting to it. Such preferences are usually carried in dependency syntactic relations, for example, the verb ‘sing’ has plausible object words ‘song’ or ‘rhythm’ rather than other nouns such as ‘house’ or ‘potato’. With such characteristic, SP is proven to be important in natural language understanding for many cases and widely applied over a variety of NLP tasks, e.g., sense disambiguation (Resnik, 1997), semantic role classification (Zapirain et al., 2013), and coreference resolution (Hobbs, 1978; Zhang et al., 2019b,c), etc.

Conventional SP acquisition methods are either based on counting (Resnik, 1997) or complex neural network (de Cruys, 2014), and the SP knowledge acquired in either way can not be directly leveraged into downstream tasks. On the other hand, the information captured by word embeddings can be seamlessly used in downstream tasks, which makes embedding a potential solution for the aforementioned problem. However, conventional word embeddings using one unified embedding for each word are not able to distinguish different relations types (such as various syntactic relations, which is crucial for SP) among words. For example, such embeddings

*Equal contribution.

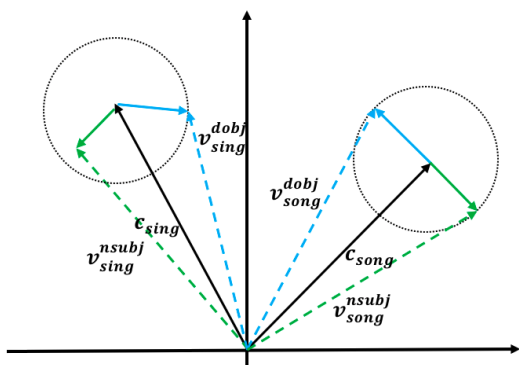


Figure 1: Illustration of the multiplex embeddings for ‘sing’ and ‘song’. Black arrows present center embeddings for words’ overall semantics; blue and green arrows refer to words’ relational embeddings for relation-dependent semantics. All relational embeddings for each word are designed to near its center embedding. *nsubj* and *dobj* relations are used as examples.

treat ‘food’ and ‘eat’ as highly relevant words but never distinguish the function of ‘food’ to be a subject or an object to ‘eat’. To address this problem, the dependency-based embedding model (Levy and Goldberg, 2014) is proposed to treat a word through separate ones, e.g., ‘food@dobj’ and ‘food@nsubj’, under different syntactic relations, with the skip-gram (Mikolov et al., 2013) model being used to train the final embeddings. However, this method is limited in two aspects. First, sparseness is introduced because each word is treated as two irrelevant ones (e.g., ‘food@dobj’ and ‘food@nsubj’), so that the overall quality of learned embeddings is affected. Second, the resulting embedding size is too large¹, which is not appropriate either for storage or usage.

Therefore, in this paper, we propose a multiplex word embedding (MWE) model, which can be easily extended to various relations between two words. A multiplex network embedding model was originally proposed for modeling multiple relations among people in a social network (Zhang et al., 2018). Interestingly, we found it also useful in capturing various relations among different words. One example is shown in Figure 1. ‘sing’ and ‘song’ are highly related to each other with the ‘predicate-objective’ rather than the ‘predicate-subject’ relation. In our model, each word has a group of embeddings, including a center embedding representing its general semantics, and

¹Assuming one has 200,000 words in the vocabulary and 20 dependency relations, and follow conventional approaches (Mikolov et al., 2013; Pennington et al., 2014) to set the embedding dimension to 300, the resulting embedding size will be about 10 Gabytes.

several embeddings representing their relation-dependent semantics. To ensure that the embeddings of the same word (under different relations) are similar to each other, we limit the Euclidean norm of relation-dependent embeddings within a small range, as shown in Figure 1. Moreover, considering that there could be many relations among words, if using a conventional dimension setting for embeddings to encode relations, the overall embedding size would be too big to be used in downstream tasks. To deal with it, we propose to use a small dimension for relation-dependent embeddings and use a transformation matrix for each relation to project them into the same space of the center embeddings. Thus the two types of embeddings can be jointly trained and the quality of the relation-dependent ones are guaranteed.

Experiments are conducted on SP acquisition over different dependency relations to evaluate whether the learned embeddings effectively capture words’ semantics over these relations. In addition, the word similarity measurement is used to assess how well words’ general semantics are learned in our model. Both evaluations confirm the superiority of our model, where the SP information is effectively preserved and the words’ overall semantics are enhanced. Particularly, further analysis also indicates that our MWE embeddings are more powerful than all existing embedding methods in SP acquisition with around 1/20 in their size comparing to previous embeddings (Levy and Goldberg, 2014). All code and resulting embeddings are available at: <https://github.com/HKUST-KnowComp/MWE>.

2 Multiplex Word Embeddings

2.1 Model Overview

As introduced in Section 1, encoding selectional preference information into embeddings can be conducted by modeling word association patterns under different dependency relations. Similar to (Levy and Goldberg, 2014), the proposed MWE model also distinguishes different relations among words and learns separate embeddings for them on each dependency edge.

Formally, let \mathcal{W} be the vocabulary set containing n words w_1, w_2, \dots, w_n and \mathcal{R} the relations set containing m relations r_1, r_2, \dots, r_m , the proposed model is expected to produce $m + 1$ embeddings for each word, where there are m relational embeddings representing relations and a center em-

bedding \mathbf{c} for the general semantics. Particularly, each relational embedding has an overall and a local version, denoted as \mathbf{v} and \mathbf{u} , where \mathbf{u} records the difference between \mathbf{v} and \mathbf{c} . For both the relational embeddings (\mathbf{v} and \mathbf{u}) and the center embedding, we use a similar way as that in word2vec (Mikolov et al., 2013) that each one has two variants to represent the semantics of a word w being the head or tail in different relations. We denote the head and tail embeddings as $\mathbf{u}_{h,w}^r \in R^s$ and $\mathbf{u}_{t,w}^r \in R^s$ for the local embeddings of word w under relation type r and $\mathbf{c}_{w,h} \in R^d$ and $\mathbf{c}_{w,t} \in R^d$ for the center embedding, respectively, where s is the dimension for \mathbf{u} and d the dimension for \mathbf{c} .

Although learning on similar information as that in (Levy and Goldberg, 2014), to reduce the sparseness of introducing m relational embeddings for each word, we do not treat each word as multiple separate prototypes. Instead, we use its center embedding to transfer information among different relations and the sum of the center embedding and a local embedding to represent the final embedding for the corresponding relation². Moreover, considering there are various relations among words, we use a lower dimension s for storage compression, with $s < d$. Thus, a transformation matrix \mathbf{X} is introduced to transform \mathbf{u} into the same vector space of \mathbf{c} . As a result, the final embeddings \mathbf{v} for head and tail of w under relation r are formulated as:

$$\begin{aligned}\mathbf{v}_{h,w}^r &= \mathbf{c}_{h,w} + \mathbf{X}_h^{rT} \mathbf{u}_{h,w}^r, \\ \mathbf{v}_{t,w}^r &= \mathbf{c}_{t,w} + \mathbf{X}_t^{rT} \mathbf{u}_{t,w}^r.\end{aligned}\quad (1)$$

2.2 Learning the MWE model

To train $\mathbf{v}_{h,w}^r$ and $\mathbf{v}_{t,w}^r$ for each w under r , we adopt the negative sampling strategy to conduct the learning process. Specifically, for each r , we use a relation tuple set \mathcal{T}_r , with each tuple $t = (w_h, r, w_t) \in \mathcal{T}_r$, where w_h is the head word and w_t the tail word. For each t , we randomly generate two negative tuples by replacing w_h and w_t with the randomly selected fake head w'_h and tail w'_t respectively. Then the learning process is expected to distinguish the positive tuple against the

² Conceptually shown in Figure 1, to ensure the resulting final embeddings not too far away from the center embeddings, we add restriction to the Euclidean norm of the local relational embeddings.

negative ones. Therefore, formally, we maximize

$$\mathcal{J} = \frac{1}{|\mathcal{T}_r|} \sum_{t \in \mathcal{T}_r} \log \frac{e^{f(w_h, r, w_t)}}{e^{f(w_h, r, w_t)} + e^{f(w_h, r, w'_t)} + e^{f(w'_h, r, w_t)}}, \quad (2)$$

over all tuples in \mathcal{T}_r for each r , with $f(\cdot)$ evaluating w_h and w_t being positive samples under r through

$$f(w_h, r, w_t) = \mathbf{v}_{h,w_h}^r \cdot \mathbf{v}_{t,w_t}^r. \quad (3)$$

For each (w_h, w_t) , we use cross entropy as the loss function

$$\begin{aligned}E &= -\log \sigma(\mathbf{v}_{h,w_h}^r \cdot \mathbf{v}_{t,w_t}^r) - \log \sigma(-\mathbf{v}_{h,w_h}^r \cdot \mathbf{v}_{t,w'_t}^r) \\ &\quad - \log \sigma(-\mathbf{v}_{h,w'_h}^r \cdot \mathbf{v}_{t,w_t}^r),\end{aligned}\quad (4)$$

to measure the learning effect for Eq. (2), with σ denoting the sigmoid function.

Combined with Eq. (1), the training process is thus to update c , u^r , and X^r with the gradients passed from the loss function using stochastic gradient descent (SGD).

In detail, take w_h as an example, its center embedding \mathbf{c}_{h,w_h} is updated by

$$\begin{aligned}\mathbf{c}_{h,w_h} &= \mathbf{c}_{h,w_h} - \lambda \cdot \eta \cdot \frac{\partial E}{\partial \mathbf{v}_{h,w_h}^r} \\ &= \mathbf{c}_{h,w_h} - \lambda \cdot \eta \cdot e(w_h, w_t) \cdot \mathbf{v}_{t,w_t}^r,\end{aligned}\quad (5)$$

where $e(w_h, w_t) = \sigma(\mathbf{e}_{h,w_h}^r \cdot \mathbf{e}_{t,w_t}^r) - t_k$, with $t_k = 1$ if (w_h, w_t) is positive sample and $t_k = 0$ for negative ones. η is the discounting learning rate. λ is an alternating weight to control the contribution of the gradient, ranging from 0 to 1.

Moreover, X^r and \mathbf{u}^r are updated as follows:

$$\begin{aligned}\mathbf{u}_{h,w_h}^r &= \mathbf{u}_{h,w_h}^r - (1 - \lambda) \cdot \eta \cdot \frac{\partial E}{\partial \mathbf{v}_{h,w_h}^r} \cdot \frac{\partial \mathbf{X}_h^r \mathbf{u}_{h,w_h}^r}{\partial \mathbf{u}_{h,w_h}^r} \\ &= \mathbf{u}_{h,w_h}^r - (1 - \lambda) \cdot \eta \cdot e(w_h, w_t) \cdot \mathbf{X}_h^r \cdot \mathbf{v}_{t,w_t}^r,\end{aligned}\quad (6)$$

$$\begin{aligned}\mathbf{X}_h^r &= \mathbf{X}_h^r - (1 - \lambda) \cdot \eta \cdot \frac{\partial E}{\partial \mathbf{v}_{h,w_h}^r} \cdot \frac{\partial \mathbf{X}_h^r \mathbf{u}_{h,w_h}^r}{\partial \mathbf{X}_h^r} \\ &= \mathbf{X}_h^r - (1 - \lambda) \cdot \eta \cdot e(w_h, w_t) \cdot \mathbf{u}_{h,w_h}^r \cdot \mathbf{v}_{t,w_t}^r,\end{aligned}\quad (7)$$

Meanwhile, \mathbf{c}_{t,w_t} , \mathbf{u}_{t,w_t}^r , and \mathbf{X}_t^r are updated in the same way of \mathbf{c}_{h,w_h} , \mathbf{u}_{h,w_h}^r , and \mathbf{X}_h^r following Eqs. (5), (6), and (7).

The above learning process is conducted on all m relations at the same time. During the training, the Euclidean norm of $\mathbf{X}^T \mathbf{u}$ for all embeddings is constrained to have a maximum semantic

Algorithm 1 Multiplex Word Embedding

Input: Relation specific tuple sets $\mathcal{T}_1, \dots, \mathcal{T}_m$, d , s , a , and η .

Output: \mathbf{c}_w , \mathbf{u}_w^r , and \mathbf{X}^r .

Initialize \mathbf{c}_w , \mathbf{u}_w^i and \mathbf{X}^i randomly.

for Each Iteration k **do**

 Update λ .

for Each tuple $t_p = (w_h, r, w_t) \in \mathcal{T}_r$ **do**

 Randomly generate the two negative examples $t_{n1} = (w'_h, r, w_t)$ and $t_{n2} = (w_h, r, w'_t)$.

$\mathcal{T}_{tmp} = [t_p, t_{n1}, t_{n2}]$

for $t \in \mathcal{T}_{tmp}$ **do**

$e(w_h, w_t) = \sigma(\mathbf{v}'_{h,w_h}{}^T \cdot \mathbf{v}^i_{t,n_t}) - t_k$

 Update \mathbf{c}_{h,w_h} , \mathbf{u}^r_{h,w_h} , \mathbf{X}^r_h , \mathbf{c}_{t,w_t} , \mathbf{u}^r_{t,w_t} , and \mathbf{X}^r_t based on Eqs. (5), (6), and (7).

if $\|\mathbf{X}^r_h{}^T \mathbf{u}^r_{h,w_h}\| > a$ **then**

 | Update \mathbf{X}^r_h and \mathbf{u}^r_{h,w_h} .

end

if $\|\mathbf{X}^r_t{}^T \mathbf{u}^r_{t,w_t}\| > a$ **then**

 | Update \mathbf{X}^r_t and \mathbf{u}^r_{t,w_t} .

end

end

end

end

drifting range a , whose value controls the distance between the local relational embeddings and the center embedding. Specifically, if $\|\mathbf{X}^T \mathbf{u}\|$ equals to a' , which is greater than a , we scale down \mathbf{X}^T and \mathbf{u} with $\sqrt{a' \cdot \frac{1}{ka}}$, where k is a scaling parameter set to 0.8 throughout this work.

The learning processing is summarized in Algorithm 1. For complexity analysis, it is obvious drawn that m relation types and n words result in $O(mn)$ and $O((d + s * m) * n)$ for time and space complexities, respectively.

2.3 The Alternating Optimization Strategy

To balance the learning process between the overall semantics and the relation-specific information, we adopt an alternating optimization strategy³ (Bezdek and Hathaway, 2003) to adjust λ based on different stage of the training instead of using a fixed weight for λ . Specifically, considering \mathbf{c} is more reliable and possesses more information while \mathbf{u} is learned with shared \mathbf{c} , we alter-

³Alternating optimization was originally proposed to train different parameters in a sequential manner and applied in various areas.

natively update \mathbf{c} and \mathbf{u} upon the convergence of \mathbf{c} . As a result, we set λ to 1 in the first half of the training process and 0 afterwards.

3 Experiments

Experiments are conducted to evaluate how our embeddings are performed on SP acquisition and word similarity measurement.

3.1 Implementation Details

We use the English Wikipedia⁴ as the training corpus. The Stanford parser⁵ is used to obtain dependency relations among words. For the fair comparison, we follow existing work and set $d = 300$, $s = 10$, and $a = 1$. Following (Keller and Lapata, 2003) and (de Cruys, 2014), we select three dependency relations (*nsubj*, *dojb*, and *amod*) as follows:

- *nsubj*: The preference of subject for a given verb. For example, it is plausible to say ‘dog barks’ rather than ‘stone barks’. The verb is viewed as the predicate (head) while the subject as the argument (tail).
- *dojb*: The preference of object for a given verb. For example, it is plausible for ‘eat food’ rather than ‘eat house’. The verb is viewed as the predicate (head) while the object as the argument (tail).
- *amod*: The preference of modifier for a given noun. For example, it is plausible to say ‘fresh air’ rather than ‘solid air’. The noun is viewed as the predicate (head) while the adjective as the argument (tail).

3.2 Baselines

We first compare the proposed multiplex embedding with the following embedding models. As it is trivial to apply these embedding models in downstream tasks, we label these models as downstream friendly.

- **word2vec**, the embedding model proposed by Mikolov et al. (2013). We use the skip-gram model for this baseline.
- **GloVe** (Pennington et al., 2014), learning word embeddings by matrix decomposition on word co-occurrences.

⁴<https://dumps.wikipedia.org/enwiki/>

⁵<https://stanfordnlp.github.io/CoreNLP/>

Model	Downstream	Keller			SP-10K			
		dobj	amod	average	nsubj	dobj	amod	average
word2vec	Friendly	0.29	0.28	0.29	0.32	0.53	0.62	0.49
GloVe	Friendly	0.37	0.32	0.35	0.57	0.60	0.68	0.62
D-embeddings	Friendly	0.19	0.22	0.21	0.66	0.71	0.77	0.71
ELMo	Friendly	0.23	0.06	0.15	0.09	0.29	0.38	0.25
BERT (static)	Friendly	0.11	0.05	0.08	0.25	0.32	0.27	0.28
BERT (dynamic)	Friendly	0.19	0.23	0.21	0.35	0.45	0.51	0.41
PP	Unfriendly	0.66	0.26	0.46	0.75	0.74	0.75	0.75
DS	Unfriendly	0.53	0.32	0.43	0.59	0.65	0.67	0.64
NN	Unfriendly	0.16	0.13	0.15	0.70	0.68	0.68	0.69
MWE	Friendly	0.63	0.43[†]	0.53[†]	0.76	0.79[†]	0.78	0.78[†]

Table 1: Results on different SP acquisition evaluation sets. As Keller is created based on the PP distribution and has relatively small size while SP-10K is created based on random sampling and has a much larger size, we treat the performance on SP-10K as the main evaluation metric. Spearman’s correlation between predicated plausibility and annotations are reported. The best performing models are denoted with bold font. † indicates statistical significant ($p < 0.005$) overall baseline methods.

- **D-embeddings**, the model proposed by Levy and Goldberg (2014) uses a skip-gram framework to encode dependencies into embeddings with multi-prototypes of words.

To investigate whether the SP knowledge can be captured by the pretrained contextualized word embedding models, we also treat following pretrained models as baselines.

- **ELMo** (Peters et al., 2018), a pretrained language model with contextual awareness. We use its static representations of words as the word embedding.
- **BERT** (Devlin et al., 2019), a pretrained bi-directional contextualized word embedding model with state-of-the-art performance on many NLP tasks.

Besides those embedding methods, we also compare with following conventional SP acquisition methods to demonstrate the effectiveness of the proposed multiplex embedding model, as it is still unclear how to leverage these methods in downstream tasks, we label these methods as downstream unfriendly.

- **Posterior Probability (PP)** (Resnik, 1997), a counting based method for the selectional preference acquisition task.
- **Distributional Similarity (DS)** (Erk et al., 2010), a method that uses the similarity of the

SP Evaluation Set	#W	#P
Keller (Keller and Lapata, 2003)	571	360
SP-10K (Zhang et al., 2019a)	2,500	6,000

Table 2: Statistics of Human-labeled SP Evaluation Sets. #W and #P indicate the numbers of words and pairs, respectively. As different datasets have different SP relations, we only report statistics about ‘nsubj’, ‘dobj’, and ‘amod’ (if available).

embedding of the target argument and average embedding of observed golden arguments in the corpus to predict the preference strength.

- **Neural Network (NN)** (de Cruys, 2014), an NN-based method for the SP acquisition task. This model achieves the state-of-the-art performance on the pseudo-disambiguation task.

For word2Vec and GloVe, we use their released code. For D-embedding, we follow their original paper using the Gensim package⁶. The dimensions of the aforementioned embeddings are set to 300 according to their original settings. For ELMo and BERT, we use their pre-trained models. As BERT was not originally designed for word level semantics tasks, for the selectional preference acquisition task, we compare with two variations of the original BERT model: (1) BERT (static), which extracts static embedding from the BERT model in a similar way as that from the ELMo model; (2) BERT (dynamic), which takes a pair of

⁶<https://radimrehurek.com/gensim/models/word2vec.html>

Model		noun	verb	adjective	overall
word2vec		0.41	0.28	0.44	0.38
Glove		0.40	0.22	0.53	0.37
D-embedding		0.41	0.27	0.38	0.36
nsubj	h	0.46	0.29	0.54	0.43
	t	0.45	0.25	0.48	0.40
	h+t	0.44	0.23	0.50	0.40
	[h,t]	0.47	0.27	0.51	0.42
dobj	h	0.46	0.27	0.45	0.41
	t	0.45	0.23	0.46	0.40
	h+t	0.45	0.20	0.45	0.38
	[h,t]	0.46	0.25	0.48	0.42
amod	h	0.47	0.25	0.52	0.37
	t	0.46	0.24	0.50	0.38
	h+t	0.46	0.24	0.52	0.38
	[h,t]	0.47	0.26	0.52	0.38
center	h	0.51	0.33	0.57	0.48
	t	0.51	0.30	0.56	0.47
	h+t	0.52	0.31	0.54	0.46
	[h,t]	0.51	0.32	0.57	0.48

Table 3: Spearman’s correlation of different embeddings for the WS measurement. ‘nsubj’, ‘dobj’, ‘amod’ represents the embeddings of the corresponding relation and ‘center’ indicates the center embeddings. h, t, h+t, and [h,t] refer to the head, tail, sum of two embeddings, and the concatenation of them, respectively. The best scores are marked in bold fonts.

words as input and produces a plausibility score for that pairs of words. The main difference between the static and dynamic version of BERT is that in the dynamic version, the contextual information can be fully utilized, which is more similar to the training objective of BERT.

3.3 Selectional Preference Acquisition

The first task in our experiment is SP acquisition. Currently, there are two methods that we can use to evaluate the quality of extracted SP knowledge: Pseudo-disambiguation (Ritter et al., 2010) and human labeled datasets (McRae et al., 1998; Keller and Lapata, 2003). However, as shown in (Zhang et al., 2019a), pseudo-disambiguation selects positive SP tuples from the corpus and generate negative SP tuples by randomly replacing the head/tail. As a result, pseudo-disambiguation only evaluates how good the model fits the corpus rather than evaluating the SP acquisition models based on ground truth. Thus, in this section, we evaluate different SP acquisition methods with ground truth (human labeled datasets). Two representative datasets, Keller (Keller and Lapata, 2003) and SP-10K (Zhang et al., 2019a), are selected as the benchmark datasets.

In each dataset, for each SP relation, various

Model	WS	Dimension	Training Time
ELMo	0.434	512	≈40
BERT	0.486	768	≈300
MWE	0.476	300	4.17

Table 4: Comparison of MWE against language models on the WS task. Overall performance, embedding dimension, and training time (days) on a single GPU are reported.

word pairs are provided. For each of the word pairs, the datasets also provide the annotated plausibility score of how likely a preference exists between that word pair under the corresponding SP relation. Thus the job for all the models is to predict the plausibility score for all word pairs under different SP relation settings. After that, we follow the conventional setting that uses the Spearman’s correlation to assess the correspondence between the predicted plausibility scores and human annotations on all word pairs for all SP relations. The statistics of these datasets are shown in Table 2.

For embedding based methods (word2vec, GloVe, D-embedding⁷, and MWE⁸), we follow previous work (Mikolov et al., 2013; Levy and Goldberg, 2014) and use the cosine similarity between embeddings of head and tail words to predict their relations. For conventional SP acquisition methods (PP, DS, and NN), we follow their original paper to compute the plausibility scores.

The experimental results are shown in Table 1. As Keller is created based on the PP distribution and have relatively small size while SP-10K is created based on random sampling and has a much larger size, we treat the performance on SP-10K as the major evaluation. Our embeddings significantly outperform other baselines, especially embedding based baselines. The only exception is PP on the Keller dataset due to its biased distribution. In addition, there are other interesting observations. First, compared with ‘dobj’ and ‘nsubj’, ‘amod’ is simpler for word2vec and GloVe. The reason behind is that conventional embeddings only capture the co-occurrence information, which is enough to predict the selectional preference of

⁷For D-embedding, since it provides embeddings for different relations (e.g., ‘food@nsubj’, ‘food@dobj’), we follow their work and directly use embeddings over certain relations to predict the score for the tested pairs. For example, given the test tuple (‘eat’, dobj, ‘food’), we compute the cosine similarity of ‘eat’ and ‘food@dobj’ rather than ‘food@subj’.

⁸For the proposed model MWE, we use the cosine similarity between v_{h, w_h} and v_{t, w_t} as the predicted plausibility.

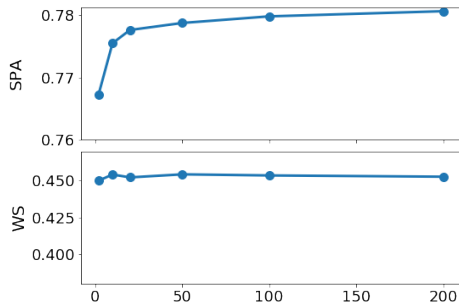


Figure 2: Effect of the different s on SP acquisition (SP-10K) and WS tasks.

‘amod’ rather than ‘nsubj’ or ‘dobj’⁹. Second, even though large-scale contextualized word embedding models like ELMo and BERT have been proved useful in many other tasks, they are still limited in learning specific and detailed semantics and thus perform inferior to our model in the SP acquisition task. For Example, ELMo and BERT can know that there is a strong semantic connection between ‘eat’ and ‘food’, but they do not know whether ‘food’ is a plausible subject or object of ‘eat’. Third, surprisingly, although NN achieves the state-of-the-art performance on the pseudo-disambiguation task, its performance is not satisfying against human annotation, especially on Keller, which is probably because the NN model overfits the training data, whose distribution is different from human SP knowledge.

3.4 Word Similarity Measurement

In addition to SP acquisition, we also evaluate our embeddings on word similarity (WS) measurement to test whether the learned embedding can effectively capture the overall semantics. We use SimLex-999 (Hill et al., 2015) as the evaluation dataset for this task because it contains different word types, i.e., 666 noun pairs, 222 verb pairs, and 111 adjective pairs. We follow the conventional setting that uses the Spearman’s correlation to assess the correspondence between the similarity scores and human annotations on all word pairs. Evaluations are conducted on the final embeddings \mathbf{v} for each relation and the center ones.

Results are reported in Table 3 with several observations. First, our model achieves the best overall performance and significantly better on nouns, which can be explained by that nouns appear in all

⁹The only possible SP relation between nouns and adjectives is ‘amod’, while multiple SP relations could exist between nouns and verbs, and co-occurrence information cannot effectively distinguish them.

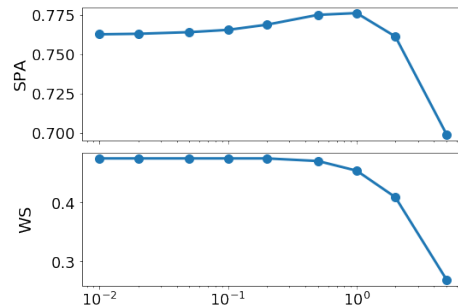


Figure 3: Effect of different α on SP acquisition (SP-10K) and WS tasks.

three relations while most of the verbs and adjectives only appear in one or two relations. This result is promising since it is analyzed by Solovyev et al. (2017) that two-thirds of the frequent words are nouns; thus there are potential benefits if our embeddings are used in downstream NLP tasks. Second, the center embeddings achieve the best performance against all the other relation-dependent embeddings, which demonstrates the effectiveness of our model in learning relation-dependent information over words and also enhancing their overall semantics.

We also compare MWE with pre-trained contextualized word embedding models in Table 4 for this task, with overall performance, embedding dimensions, and training times reported. It is observed that that MWE outperforms ELMo and achieves comparable results with BERT with smaller embedding dimension and much less training complexities.

4 Analysis

With the experiments on SP acquisition and word similarity measurement, we conduct further analysis for different settings for hyper-parameters and learning strategies, as well as the scalability of our model.

4.1 Effect of the Local Embedding Dimension

We investigating the performance of MWE on SPA (SP acquisition) and WS tasks with different s . As shown in Figure 2, the performance of our model is in an increasing trend when we increase the value of s . When the dimension reaches 10, the performance almost reaches the top, which confirms that the local relational embeddings can be effective in small dimensions (about 1/30 of the center dimension) when using center embeddings as the constraint.

Training Strategy	Averaged SPA	Overall WS
$\lambda = 1$	0.762	0.476
$\lambda = 0$	0.073	0.018
$\lambda = 0.5$	0.493	0.323
Alternating optimization	0.775	0.476

Table 5: Comparisons of different training strategies.

4.2 Effect of the Semantic Drifting Range

Similar to s , we also investigate the model performance with the influence of different constraint a . It is observed in Figure 3 that, the constraint gets looser with the increase of a . For SP acquisition, in a small range, the model performance gradually improves along with the increasing value of a because these embeddings are more flexible to capture intense relation-dependent information. However, once the range passes a threshold (1 in our experiment), the embeddings get farther away from their general semantics and start to overfit, which is also observed in the WS experiment, and thus the performance drops monotonically.

4.3 Effect of Alternating Optimization

An analysis is also conducted to demonstrate the effectiveness of the alternating optimization strategy. As shown in Table 5, we compare our model with several different strategies. The first one is to put all weights to the center embedding (fix λ to 1), which never updates the local relational embeddings. As a result, it can achieve similar performance on word similarity measurement but is inferior in SP acquisition because no relation-dependent information is preserved. The second strategy is to put all weights to the local relational embeddings (fix λ to 0). In this case, it performs very poor owing to the loss of general semantics. Last but not least, the alternating optimization also outperforms the setting with a fixed $\lambda = 0.5$, which can be explained by alternating optimization can first get a good overall semantic representation and then acquire the SP knowledge on top of that. To summarize, the alternating optimization provides an effective solution to training our model with a smart process in adjusting the contribution of different embeddings as well as stabilizing their optimization.

4.4 Scalability

Scalability of embeddings is an important factor in real applications. Practically, GPU or similar

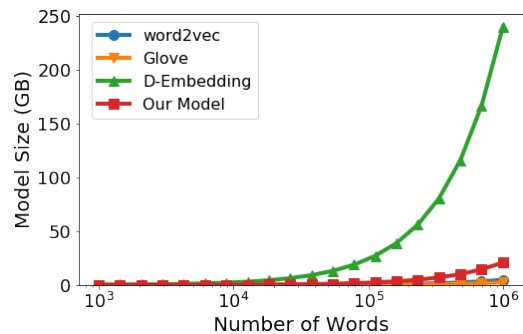


Figure 4: Scalability on embedding size against number of words.

computation support is required to accommodate embeddings to applying neural models. Normally, the current most affordable GPUs typically have a memory size limitation around 10 GB. Thus, to ensure the learned embeddings can be used in downstream tasks, their size becomes an important concern when evaluating different embedding models, especially when there exist many different relations between words in our model.

Assuming that there are 20 relations among words, the size of all embedding models with the increasing word numbers in Figure 4. word2Vec and GloVe have the smallest size because they only train one embedding for each word while sacrificing the relation-dependent information among them. As a comparison, D-embeddings are trained on multiple prototypes for each word to record relation information, thus their size dramatically explodes with the increasing of the vocabulary. Consequently, it is easily computed for D-embeddings that 200,000 words will result in a 10GB model, which is unfeasible to be used in downstream tasks. Effectively with the small dimension for our local relational embeddings, relation information can be preserved in a small-sized model, which shows a compatible space requirement with the conventional embeddings.

5 Related Work

Learning word embeddings has become an important research topic in NLP (Bengio et al., 2003; Turney and Pantel, 2010; Collobert et al., 2011; Song and Shi, 2018), with the capability of embeddings demonstrated in different languages (Song et al., 2018a) and tasks such as parsing (Chen and Manning, 2014), machine translation (Zou et al., 2013), coreference resolution (Lee et al., 2018), etc. Conventional word embeddings (Mikolov et al., 2013; Pennington et al., 2014) often leverage

word co-occurrence patterns, resulting in a major limitation that they coalesce different relationships between words into a single vector space. To address this limitation, dependency-based embedding model (Levy and Goldberg, 2014) was proposed to represent each word with several separate embeddings, and then suffers from its sparseness and the huge size of the resulting embeddings. Alternatively, our MWE model uses a set of (constrained and small) embeddings for each word to encode its general and relation-dependent semantics to resolve the sparseness and the size problem.

Another line of work related to this paper is the research on SP, which is considered important in language understanding and has been proved helpful in various downstream tasks (Tang et al., 2016; Resnik, 1997; Hobbs, 1978; Inoue et al., 2016; Zafirain et al., 2013). Several studies attempted to acquire SP automatically from raw corpora (Resnik, 1997; Rooth et al., 1999; Erk et al., 2010; Santus et al., 2017). However, they are designed specifically for SP acquisition and are limited in leveraging SP knowledge for downstream tasks. Compared to them, the learned embeddings from our model are useful resources that incorporate SP knowledge and can be seamlessly leveraged in other NLP models.

6 Conclusion

In this paper, we present a multiplex word embedding model encoding selectional preference information from word dependency relations. Different from conventional embeddings, for each word, we proposed to use a set of embeddings to represent its general (center embedding) and relation-dependent (local relational embedding) semantics, where their combination present the final embedding for each word under particular relations. Experiments on SP acquisition and word similarity measurement illustrated that our model better encodes SP knowledge than different baselines without harming its capability in representing general semantics. Analysis is also conducted with respect to different settings and optimization strategies, as well as the effect of model size, which further illustrates the validity and effectiveness of the proposed model and its learning process.

Acknowledgements

This paper was supported by the Early Career Scheme (ECS, No. 26206717) from Research

Grants Council in Hong Kong and the Tencent AI Lab Rhino-Bird Focused Research Program.

References

- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.
- James C Bezdek and Richard J Hathaway. 2003. Convergence of alternating optimization. *Neural, Parallel & Scientific Computations*, 11(4):351–368.
- Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*, pages 740–750.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12:2493–2537.
- Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *Proceedings of EMNLP*, pages 26–35.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- Katrin Erk, Sebastian Padó, and Ulrike Padó. 2010. A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, 36(4):723–763.
- Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.
- Jerry R Hobbs. 1978. Resolving pronoun references. *Lingua*, 44(4):311–338.
- Naoya Inoue, Yuichiroh Matsubayashi, Masayuki Ono, Naoaki Okazaki, and Kentaro Inui. 2016. Modeling context-sensitive selectional preference with distributed representations. In *Proceedings of COLING*, pages 2829–2838.
- Frank Keller and Mirella Lapata. 2003. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29(3):459–484.
- Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of NAACL-HLT*, pages 687–692.
- Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of ACL*, pages 302–308.

- Ken McRae, Michael J Spivey-Knowlton, and Michael K Tanenhaus. 1998. Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, 38(3):283–312.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.
- Philip Resnik. 1997. Selectional preference and sense disambiguation. *Tagging Text with Lexical Semantics: Why, What, and How?*
- Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *Proceedings of ACL*, pages 424–434.
- Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of ACL*, pages 104–111.
- Enrico Santus, Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017. Measuring thematic fit with distributional feature overlap. In *Proceedings of EMNLP*, pages 648–658.
- Valery D Solovyev, Vladimir V Bochkarev, and Anna V Shevlyakova. 2017. Dynamics of core of language vocabulary. *arXiv preprint:1705.10112*.
- Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning word representations with regularization from prior knowledge. In *Proceedings of CoNLL*, pages 143–152.
- Yan Song and Shuming Shi. 2018. Complementary learning of word embeddings. In *Proceedings of IJCAI*, pages 4368–4374.
- Yan Song, Shuming Shi, and Jing Li. 2018a. Joint learning embeddings for chinese words and their components via ladder structured networks. In *Proceedings of IJCAI*, pages 4375–4381.
- Yan Song, Shuming Shi, Jing Li, and Haisong Zhang. 2018b. Directional skip-gram: Explicitly distinguishing left and right context for word embeddings. In *Proceedings of NAACL-HLT*, pages 175–180.
- Haiqing Tang, Deyi Xiong, Min Zhang, and Zhengxian Gong. 2016. Improving statistical machine translation with selectional preferences. In *Proceedings of COLING*, pages 2154–2163.
- Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.
- Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial intelligence*, 6(1):53–74.
- Beñat Zepirain, Eneko Agirre, Lluís Màrquez, and Mihai Surdeanu. 2013. Selectional preferences for semantic role classification. *Computational Linguistics*, 39(3):631–663.
- Hongming Zhang, Hantian Ding, and Yangqiu Song. 2019a. SP-10K: A large-scale evaluation set for selectional preference acquisition. In *Proceedings of ACL*, pages 722–731.
- Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. 2018. Scalable multiplex network embedding. In *Proceedings of IJCAI*, pages 3082–3088.
- Hongming Zhang, Yan Song, and Yangqiu Song. 2019b. Incorporating context and external knowledge for pronoun coreference resolution. In *Proceedings of NAACL-HLT*, pages 872–881.
- Hongming Zhang, Yan Song, Yangqiu Song, and Dong Yu. 2019c. Knowledge-aware pronoun coreference resolution. In *Proceedings of ACL*, pages 867–876.
- Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP*, pages 1393–1398.