HWE: Word Embedding with Heterogeneous Features

Jhih-Sheng Fan, Mu Yang, Peng-Hsuan Li and Wei-Yun Ma
Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan
{fann1993814,emfomy,jacobvsdanniel}@gmail.com and ma@iis.sinica.edu.tw

***Abstract*—Distributed word representation is widely used in Natural Language Processing. However, traditional approaches, which learn word representations from the co-occurrence information in large corpora, might not capture fine-grained syntactic and semantic information. In this paper, we propose a general and flexible framework to incorporate each type (e.g., word-sense, part-of-speech, topic) of contextual feature for learning feature-specific word embeddings in an explicit fashion, namely Heterogeneous Word Embedding (HWE). Experimental results on both intrinsic and extrinsic tasks show that HWE outperforms the baseline and various state-of-the-art models. Moreover, through the concatenation over HWE and the corresponding feature embeddings, each word would have different contextual representation under different contexts, which achieves even more significant improvement. Finally, we illustrate the insight of our model via visualization of the learned word embeddings.**

***Index Terms*—word embedding, distributed word representation, unsupervised learning.**

I. Introduction

Distributed word representation, which maps each word into a continuous vector, is an important research topic in natural language processing (NLP) [1], [2]. It has been proven to be beneficial in a wide range of applications, such as neural language model [3], [4], named entity recognition [5], syntactic parsing [6], and machine translation [7], [8].

Recently, several unsupervised approaches for learning word embeddings [9]–[12] have been proposed and have had great results in various NLP tasks. These approaches based on the contextual distribution of a large corpus for learning word representations, such as Word2Vec [11] and GloVe [12]. However, traditional approaches, which learn word representations from the co-occurrence information in large corpora, might not capture fine-grained syntactic and semantic information.

To address this issue, some recent approaches have been proposed to incorporate other information into word representations. For instance, several knowledge-joined word embedding models are proposed and achieved significant improvements in various NLP tasks [13]–[15]. These approaches exploit semantic relations between words defined in knowledge bases to learn more semantic word representation.

Another line of research is Topical Word Embedding (TWE) [16]–[18], in which topical word refers to a word taking a specific topic learned by LDA [19] as context. Their basic idea is to allow each word to have different embeddings in different contexts. For instance, in [17], topical word embeddings are obtained via concatenation over correspond word embeddings and topic embeddings. Their best TWE model (TWE-1) learns both kinds of embeddings separately and simultaneously. Since both kinds of embeddings are separately trained in the same input layer, the topic information is infused into word embeddings in an implicit fashion. For more explicitly infusing topic information into word embeddings, other variations were also investigated, such as TWE-2, in which each word-topic pair is regarded as a pseudoword, and topical word embeddings are thus learned directly. However, probably due to the problem of data sparseness, TWE-2 performs much worse than TWE-1.

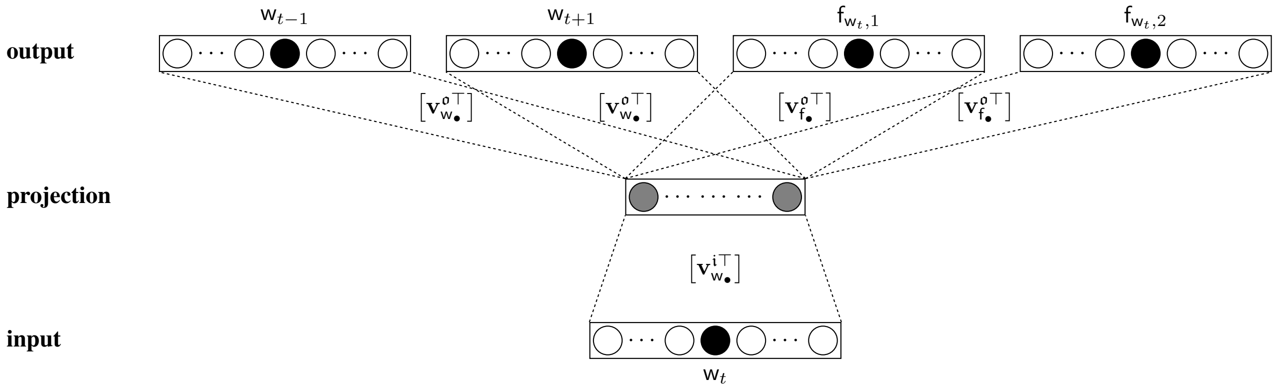
Inspired by topical word embeddings and aimed to provide an effective solution to the incorporation of the word and its contextual features in an explicit fashion, in this paper, we propose a new approach, namely Heterogeneous Word Embedding (HWE), to explicitly infuse the syntactic or semantic features into word embeddings: the word embeddings is in the input layer while its corresponding feature embeddings are in the output layer, shown in Figure 1. In other words, given a word, our model's goal is to predict its corresponding features and contextual words. The experimental results show that HWE outperforms TWE-1 and other state-of-the-art approaches on both intrinsic and extrinsic tasks, including word similarity measure, word analogy, dependency parsing, document classification, and sentiment analysis.

In addition, HWE is a flexible and general model, in which different kinds of syntactic or semantic features can be considered individually or simultaneously. Our experimental results show that different feature settings for different particular tasks would achieve more improvement. Moreover, through the concatenation of HWE and the corresponding feature embeddings, each word would have different contextual representation under different contexts, which achieves even more significant improvement.

In the end, we also illustrate the two-dimensional PCA projection of the word embeddings learned by HWE and the skip-gram, in order to analyze the insight meaning of the word representation.

Our main contributions are as follows:

- A general and flexible framework to infuse specific fine-grained information into word embeddings.
- For each specific NLP task, we investigate which features should be considered in the embedding training.
- The corresponding feature embeddings can be also learned and used in the proposed framework.



output
projection
input
w_{t-1}
w_{t+1}
$\mathsf{f}_{\mathsf{w}_t,1}$
$\mathsf{f}_{\mathsf{w}_t,2}$
$[\mathbf{v}^{\mathsf{o}\top}_{\mathsf{w}_\bullet}]$
$[\mathbf{v}^{\mathsf{o}\top}_{\mathsf{f}_\bullet}]$
$[\mathbf{v}^{\mathsf{i}\top}_{\mathsf{w}_\bullet}]$
w_t

Fig. 1: Heterogeneous Word Embedding (HWE) learning framework. In this example, the target word w_t in input layer predicts the two context words (w_{t-1}, w_{t+1}) and its two corresponding features ($\mathsf{f}_{\mathsf{w}_t,1}$, $\mathsf{f}_{\mathsf{w}_t,2}$).

II. Feature Choosing

With the development of NLP, we obtain word features easily by some resources or NLP toolkits, such as WordNet [20], Stanford Corenlp[1] [21]. To measure how the different types of features affect word embeddings training in the HWE model, so far we investigate three different feature types described as follows:

- Word Sense: The word sense contain the semantic information to identify word meaning clearly between different contexts. In this work, we collect word senses from WordNet, and apply word sense disambiguation (WSD) by pywsd[2] [22].
- POS: The part-of-speech (POS) tags contain syntactic roles of words. Taking POS tags as features are widely applied in many NLP tasks, such as dependency parsing [23] and named entity recognition (NER) [5]. In this work, we use a popular NLP toolkit, Stanford CoreNLP, to identify POS tags for words of every sentence in the training corpus.
- Topic: In this work, we use a general topic model Latent Dirichlet allocation (LDA) [19] to obtain the topic information of each word. The topic model is based on the word co-occurrence in documents to learn words' topic distributions. A word in different contexts may correspond to different topics. For example, in a document with the topic technology, the word 'apple' refers to a company, and in another document with the topic food, the word might refer to a fruit.

III. Method

In this section, we first briefly introduce the traditional skip-gram model [11], followed by two state-of-the-art approaches — Semantic Word Embedding (SWE) [14] and Topical Word Embedding (TWE-1) [17]. In this paper, we use these models as baselines. Finally, we introduce our HWE model and show how to compute the derivatives of the objective function for optimization.

[1]https://stanfordnlp.github.io/CoreNLP/

[2]https://github.com/alvations/pywsd/

A. Baselines

1) Skip-gram: The skip-gram model is based on the word-context relation in the corpus for learning word representations [11]. It predicts the context words of a given word; that is, it learns the word embeddings by maximizing the objective:

$$\mathcal{Q}_0 = \frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-c\leq j\leq c\\ j\neq 0}} \log \mathcal{P}(\mathsf{w}_{t+j}|\mathsf{w}_t), \tag{1}$$

where $\mathsf{w}_\bullet$ are the words, c is the size of the context window, and T is the length of the word sequence. All the following models (SWE, TWE, HWE) are approaches of the skip-gram.

2) SWE: Liu et al. proposed the Semantic Word Embedding (SWE) model [14], which constrains word embeddings using lexical semantic knowledge base. More specifically, for any word triplet $\mathsf{w}_i, \mathsf{w}_j, \mathsf{w}_k$ with similarity$(\mathsf{w}_i, \mathsf{w}_j) >$ similarity$(\mathsf{w}_i, \mathsf{w}_k)$, the inequality should be held in the embedding space

$$\mathrm{sim}\big(\mathbf{v}_{\mathsf{w}_i}, \mathbf{v}_{\mathsf{w}_j}\big) > \mathrm{sim}(\mathbf{v}_{\mathsf{w}_i}, \mathbf{v}_{\mathsf{w}_k}), \tag{2}$$

where $\mathbf{v}_{\mathsf{w}_\bullet}$ denote the embedding vectors of $\mathsf{w}_\bullet$.

In conclusion, SWE maximizes the following objective function:

$$\mathcal{Q} = \mathcal{Q}_0 - \beta\mathcal{D}, \tag{3}$$

where $\beta > 0$ is the penalty parameter, and $\mathcal{D}$ represents the inequity constraints.

3) TWE-1: Liu et al. proposed the Topical Word Embedding (TWE) model [17], in which a topical word is a word that takes a specific LDA-learned topic as context. Of their various TWE models, TWE-1, whose goal is predicting the context words of a given word and its topic, performs the best. Formally, given a word sequence $\{\mathsf{w}_1, \mathsf{w}_2, \ldots, \mathsf{w}_T\}$ and a topic sequence $\{\mathsf{z}_1, \mathsf{z}_2, \ldots, \mathsf{z}_T\}$, where $\mathsf{z}_\bullet$ is the inferred topic of the word $\mathsf{w}_\bullet$, TWE-1 maximizes the following objective function:

$$\mathcal{Q} = \mathcal{Q}_0 + \frac{1}{T}\sum_{t=1}^{T}\sum_{\substack{-c\leq j\leq c\\ j\neq 0}} \log \mathcal{P}(\mathsf{w}_{t+j}|\mathsf{z}_t). \tag{4}$$

B. Heterogeneous Word Embedding

The HWE model predicts a word's features and its contextual words, as shown in Figure 1. The word embeddings acquire the feature information more explicitly than TWE-1. We regard the topic information, or any information such as word sense or POS, as a type of features.

In the training phase, the objective function is

$$\mathcal{Q} = \mathcal{Q}_0 + \frac{1}{T}\sum_{t=1}^{T}\sum_{\mathsf{f}_{\mathsf{w}_t}\in\mathcal{F}_{\mathsf{w}_t}} \log \mathcal{P}(\mathsf{f}_{\mathsf{w}_t}|\mathsf{w}_t), \tag{5}$$

where $\mathcal{F}_{\mathsf{w}_t}$ is the feature set of the target word w_t. Due to the impracticality of the softmax function indicated by Mikolov et al., we adopt noise contrastive estimation (NCE) [24] as in [11] to approximate the probability function.

Let $\mathcal{F}$ be the set of all features in the corpus. For each $\mathsf{f}_{\mathsf{w}_t}$, by selecting several negative samples to form a negative sample set $\mathcal{F}^{\mathsf{n}}_{\mathsf{w}_t} \subset \mathcal{F}\setminus\{\mathsf{f}_{\mathsf{w}_t}\}$, we define the sample set $\mathcal{F}^{\mathsf{s}}_{\mathsf{w}_t}$ as the union of $\{\mathsf{f}_{\mathsf{w}_t}\}$ (the positive sample) and $\mathcal{F}^{\mathsf{n}}_{\mathsf{w}_t}$ (the negative samples). Here, for all $\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t} \in \mathcal{F}^{\mathsf{s}}_{\mathsf{w}_t}$, we use $\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}$ to distinguish the positive and the negative samples: $\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}} = 1$ for $\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t} = \mathsf{f}_{\mathsf{w}_t}$, and $\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}} = 0$ for $\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t} \neq \mathsf{f}_{\mathsf{w}_t}$.

Therefore, the probability function can be approximate as

$$\mathcal{P}(\mathsf{f}_{\mathsf{w}_t}|\mathsf{w}_t) \approx \prod_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}\in\mathcal{F}^{\mathsf{s}}_{\mathsf{w}_t}} \mathcal{L}\big(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}\big|\mathsf{w}_t\big). \tag{6}$$

Note that $\mathcal{L}$ is the likelihood function:

$$\mathcal{L}\big(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}\big|\mathsf{w}_t\big) = \sigma\Big(\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\cdot\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}\Big)^{\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}}\cdot\sigma\Big(-\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\cdot\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}\Big)^{1-\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}}, \tag{7}$$

where $\mathbf{v}^{\mathsf{i}}_\bullet$ and $\mathbf{v}^{\mathsf{o}}_\bullet$ are the input and the output embeddings, and σ is a sigmoid function.

To simplify the procedure of optimization, we transform the log-likelihood $\mathcal{L}(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}|\mathsf{w}_t)$ to the log format as following:

$$\begin{aligned}\log[\mathcal{L}(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}|\mathsf{w}_t)] = {}& \log\Big[\sigma\Big(\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\cdot\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}\Big)\Big]\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\\ &+\log\Big[\sigma\Big(-\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\cdot\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}\Big)\Big]\Big(1-\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\Big).\end{aligned} \tag{8}$$

Furthermore, we adopt the same method as [11] did to calculate the probability $\mathcal{P}(\mathsf{w}_{t+j}|\mathsf{w}_t)$ for our optimization.

C. Optimization

With previous inference as eq. (8), the embeddings is learned to adopt stochastic gradient descent (SGD) algorithm. First, we show how to compute partial derivative of the function $\log[\mathcal{L}(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}|\mathsf{w}_t)]$ with respect to the output embeddings $\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}$ and the input embeddings $\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}$:

$$\begin{aligned}\frac{\partial}{\partial\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}}\log[\mathcal{L}(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}|\mathsf{w}_t)] &= \Big[\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}} - \sigma\Big(\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\cdot\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}\Big)\Big]\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t};\\ \frac{\partial}{\partial\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}}\log[\mathcal{L}(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}|\mathsf{w}_t)] &= \Big[\lambda_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}} - \sigma\Big(\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\cdot\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}\Big)\Big]\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}.\end{aligned} \tag{9}$$

Next, we update the two embeddings:

$$\begin{aligned}\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}} &\leftarrow \mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}} + \alpha\frac{\partial}{\partial\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}}\log\big[\mathcal{L}\big(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}\big|\mathsf{w}_t\big)\big];\\ \mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t} &\leftarrow \mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t} + \alpha\sum_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}\frac{\partial}{\partial\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}}\log\big[\mathcal{L}\big(\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}\big|\mathsf{w}_t\big)\big],\end{aligned} \tag{10}$$

where $0 < \alpha < 1$ is a learning rate. Note that we update the input embedding $\mathbf{v}^{\mathsf{i}}_{\mathsf{w}_t}$ after updating the output embedding $\mathbf{v}^{\mathsf{o}}_{\mathsf{f}^{\mathsf{s}}_{\mathsf{w}_t}}$. For the following experiments tasks, we use the input word embeddings and the output feature embeddings.

IV. Experiments

In this section, we describe the experiment setting and report all the results to evaluate the proposed HWE model. We carry out five general natural language processing tasks — word similarity measure, word analogy, dependency parsing, document classification, and sentiment analysis. Here we compare the performance of HWE with a baseline model, the skip-gram, and two state-of-the-art models, TWE-1 and SWE. Moreover, we also investigate the performance of the replacement of the topic features with POS and word sense in TWE-1, denoted as TWE-1-POS and TWE-sense respectively, in order to get the comprehensive comparison. Please note that in the tables of evaluation results, unless particular statement in Tables IV and V, we use HWE-$\star$, such as HWE-POS, to refer to taking the embeddings of word (aimed to predict $\star$) other than the embeddings of $\star$ in the evaluation. Similarly, we use TWE-1-$\star$ to refer to taking the embeddings of word (learned together with $\star$ in the same input layer) other than the embeddings of $\star$ in the evaluation.

A. Experimental Setup

1) Training Corpora: We use the New York Times (NYT) 1994–97 portion of Gigaword v5.0 [25] as the training corpus to learn embeddings, containing about 436.5 million tokens and about 1.2 million unique terms.

2) Feature Preparation: For the word sense features, we extract 95,882 synsets from WordNet and assign each word to its corresponding synset identities as a word sense. Note that we do not perform word disambiguation for the training corpus; rather each word in a sentence may contain several word senses. For word topic features, we use MALLET's LDA [26] implementation[3], which uses Gibbs sampling [27], to extract the topic information corresponding to each word in a document. We use 400 topics for the larger NYT corpus.

3) Training Hyperparameter Setting: All models are implemented based on the skip-gram model with the same hyperparameters: a dimensionality of 300, a window size of 10, a number of negative samples 10, subsampling at 10^{-3}, and a single iteration for the NYT corpus. For SWE [14], we follow their best parameter setting ($\beta = 0.3$), and apply the

[3]http://mallet.cs.umass.edu/

Corpus	Skip-gram	HWE				TWE-1				SWE
	-	*POS*	*Sense**	*Sense-WSD*†	*Topic*	*POS*	*Sense**	*Sense-WSD*†	*Topic*	-
WS-353	56.22	54.76	**60.62**	58.42	58.81	55.64	60.36	53.56	56.64	55.26
WS-353-S	63.98	62.68	**68.24**	66.45	66.21	63.58	66.97	62.40	63.61	61.10
WS-353-R	53.42	51.34	**56.45**	54.25	55.94	52.80	55.38	50.06	55.37	51.80
MC 30	74.11	65.18	**77.13**	72.39	71.77	67.81	71.48	75.11	76.18	68.99
RG 65	63.45	52.40	**67.03**	65.94	63.78	57.84	58.62	63.17	66.02	64.42
Rare word	30.67	30.46	32.40	**32.91**	32.11	26.87	29.78	28.82	27.26	29.94
MEM	63.81	61.23	65.02	**65.66**	64.98	61.62	64.75	60.34	63.21	64.73
Mturk 287	65.15	66.59	62.44	66.52	66.59	65.67	64.18	66.26	**66.62**	65.11
Mturk 771	54.75	53.00	58.89	59.53	57.60	53.54	58.73	53.78	54.54	**60.95**
YP 130	41.18	32.95	**55.27**	48.02	43.70	37.73	35.68	37.68	37.17	43.13
SimLex 999	34.95	34.06	37.30	36.59	35.30	32.88	37.08	31.65	31.43	**45.43**
Verb 143	31.53	**36.53**	36.12	34.87	35.43	30.76	28.22	33.86	34.48	36.22

* Each word predicts multiple senses belong to itself from WordNet (without WSD).
† Each word predicts the corresponded sense with applying WSD to the sequences.

TABLE I: Spearman rank correlation (%) on the word similarity task. All vectors are 300-dimension word embeddings.

	Corpus	Skip-gram	HWE				TWE-1				SWE
		-	*POS*	*Sense*	*Sense-WSD*	*Topic*	*POS*	*Sense*	*Sense-WSD*	*Topic*	-
Semantic	Google	42.20	42.90	43.28	**43.72**	41.33	40.50	39.42	40.65	40.97	41.62
	-	-	-	-	-	-	-	-	-	-	-
Syntactic	Google	48.64	52.92	48.97	**58.42**	49.68	44.42	39.46	47.28	44.37	48.07
	MSR	41.13	47.66	41.47	**52.58**	42.21	36.33	29.62	39.76	38.28	39.88
Total	Google	45.72	48.37	46.39	**51.75**	45.89	42.64	39.44	44.27	42.83	45.14
	MSR	41.13	47.66	41.47	**52.58**	42.21	36.33	29.62	39.76	38.28	39.88

TABLE II: Accuracy (%) on the analogy task. All vectors are 300-dimension word embeddings.

synonym-antonym and hypernym-hyponym relations into the model to train the word embeddings.

B. Word Similarity

1) Task Description: We calculate the Spearman rank correlation between human judgment and the similarity scores computed using word embeddings on 12 datasets: MC-30 [28], MEN-3k [29], MTurk-287 [30], MTurk-771 [31], RG-65 [32], Rare-Word [33], SimLex-999 [34], Verb-143 [35], WordSim-353 (WS-353) [36]. WordSim-353-Similarity (WS-353-S), WordSim-353-Relatedness (WS-353-R), and YP-130 [37]. Note that the WS-353-S and WS-353-R are extracted from WS-353 [38].

2) Experimental Results: As shown in Table I, HWE-Sense outperforms other feature combination models in most datasets, as word sense is an important factor for word similarity tasks. Both our HWE-Sense models (Sense and Sense-WSD) usually outperform the TWE-Sense models, showing that TWE-Sense models do not completely leverage the advantage of senses in its implicit fashion. However, the HWE-Sense models underperforms SWE in a few tasks, probably due to the fact that our HWE-Sense models do not use fine-grained features for lexical knowledge such as hyponym, hypernym, and antonym.

C. Word Analogy

1) Task Description: We use two datasets: the Google analogy dataset [11], containing 19,544 word analogy questions, and the MSR analogy dataset [11], containing 8,000 word analogy questions. Google analogy questions can be divided into a semantic subset and a syntactic subset; the MSR analogy questions are syntactic.

2) Experimental Results: We present the word analogy results in Table II: the proposed HWE models achieve significantly better performance than the others. The HWE-Sense-WSD model performs the best on two types of analogy questions, and the HWE-POS model also yields the better performance on syntactic analogy questions. These features extracted by sequential labeling contain the implicit relations information from context words. All feature combinations of TWE-1 underperform the skip-gram model.

D. Dependency Parsing

1) Task Description: We use the Stanford neural network dependency parser (NNDEP) [39] to train a dependency parser using the English Penn Treebank (PTB) for evaluating embeddings. Here, we pre-train word embeddings for each model on the NYT corpus as the inputs of NNDEP to train the dependency parsers. Following the default settings for the NNDEP model, we use 50-dimensional pre-trained word embeddings and train the NNDEP model using 20,000

	Metric	Random	Skip-gram	HWE				TWE-1	SWE
		-	-	*POS*	*Sense*	*Sense-WSD*	*Topic*	*POS*	-
Dev	UAS	82.46	87.39	**87.77**	87.11	87.64	87.22	87.30	87.22
	LAS	77.57	83.97	**84.52**	83.68	84.35	83.79	83.92	83.81
Test	UAS	81.96	86.79	**87.27**	86.76	87.01	86.54	86.89	86.55
	LAS	77.43	83.68	**84.32**	83.60	83.87	83.40	83.77	83.43

TABLE III: UAS and LAS scores (%) on dependency parsing task. All vectors are 300-dimension word embeddings.

		Skip-gram		HWE-POS		
Word		✓	✓	✓	✓	✓
Feature (POS)			✓			✓
Dimension		50	50+50	50	100	50+50
Dev	UAS	87.39	89.47	87.77	88.23	**89.90**
	LAS	83.97	86.62	84.52	85.00	**87.11**
Test	UAS	86.79	89.14	87.27	87.85	**89.48**
	LAS	83.68	86.72	84.32	84.94	**87.16**

TABLE IV: Evaluation results (%) on dependency parsing task with 50/100-dimension vectors.

iterations, following the default setting for NNDEP.

2) Experimental Results: Table III shows the dependency parsing results with the NNDEP model. The proposed HWE-POS model performs the best, showing the syntactical information (POS) benefits dependency parsing.

As shown in Figure 2, the HWE-POS model converges faster than other models during training processing. However, as the iteration increases, the advantage of POS becomes insignificantly.

We also compare the influence of POS on the skip-gram model and HWE-POS. We train 50-dimensional word embeddings for each model. Furthermore, we concatenate the two 50-dimensional word and feature (POS) embeddings to yield 100-dimensional embeddings. Here we train the skip-gram feature embeddings using the POS sequence from the NYT corpus. Also, for comparison, we train extra HWE-POS 100-dimensional word embeddings.

As shown in Table IV, HWE-POS outperforms the skip-gram model on both word embeddings and concatenated embeddings cases. In HWE-POS, the word and feature embeddings share information with each other during the training. This relation between word and feature embeddings improves the quality of the dependency parser trained by HWE-POS. With this concatenation, HWE-POS trains a dependency parser that yields the best UAS and LAS scores.

E. Document Classification

1) Task Description: We use 20 Newsgroups[4], containing about 20,000 newsgroup documents and 20 categories. The training set contains 11,305 documents, and the testing set contains 7,532 documents.

[4]http://qwone.com/~jason/20Newsgroups/

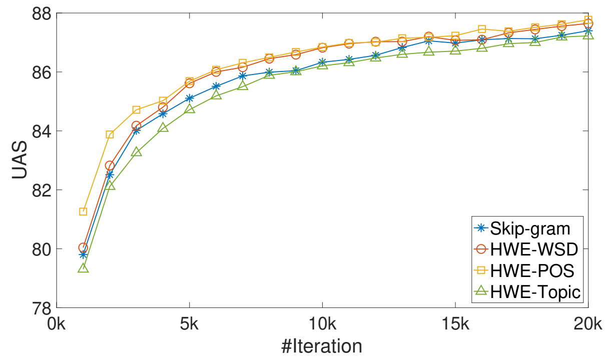


Fig. 2: The UAS score of each model on the development set versus the number of training iteration.

First, we set 100 topics for the smaller dataset for learning LDA topic information, and learn the embeddings from the training set. We generate document representations by summing up each word embeddings of a document (or, additionally, concatenating with the summation of the corresponding feature embeddings). With these document representations, we train a logistic regression classifier for document classification. We also compare TWE-1 [17] with the HWE models on the task.

2) Experimental Results: In Table V, HWE-Topic, which concatenates the word and corresponding topic embeddings, largely outperform other models since it leverages the word-topic feature from LDA for effective use with the context-word information. Also, most HWE models, which use only word embeddings to represent document feature vectors, are better than the skip-gram model in terms of accuracy.

F. Sentiment Analysis

1) Task Description: Following previous work, we use the Stanford Sentiment Treebank with binary classes (SST-2) [40] to evaluate different pre-trained word vectors [41]. SST-2 contains 6920/872/1821 sentences for train/validate/test, each of which is either annotated as positive or negative sentiment. Each sentence is represented by its average word vector for an SVM classifier. We use SVM with the Gaussian kernel. The hyperparameter γ is set to $1/k$, where k is the number of features. The hyperparameter c for each word embedding model is determined by the validation split with a grid search from 10^{-4} to 10^{4}.

	LDA	**Skip-gram**	**HWE-POS**		**HWE-Sense-WSD**		**HWE-Topic**		**TWE-1**	**SWE**
Word		✓	✓	✓	✓	✓	✓	✓	✓	✓
Feature	✓			✓		✓		✓	✓	
Dimension	100	300	300	300+300	300	300+300	300	300+300	300+300	300
Recall	65.65	68.78	69.50	67.88	70.09	71.82	68.71	**74.30**	72.47	70.52
Precision	65.02	73.70	71.78	72.92	71.83	74.74	72.61	**75.41**	74.52	72.79
F1	64.62	70.10	69.98	69.01	70.44	72.46	69.54	**74.53**	72.87	71.01
Accuracy	67.19	69.85	70.57	68.89	71.20	72.74	69.82	**75.32**	73.47	71.55

TABLE V: Evaluation results (%) of multi-class document classification with 100/300/600-dimension vectors.

Skip-gram	**HWE**				**TWE-1**				**SWE**
-	*POS*	*Sense*	*Sense-WSD*	*Topic*	*POS*	*Sense*	*Sense-WSD*	*Topic*	-
79.63	78.91	79.79	79.79	**80.45**	78.58	77.81	80.34	79.13	80.18

TABLE VI: SST-2 accuracy (%) on sentiment task. All vectors are 300-dimension word embeddings.

2) Experimental Results: The testing result for each word embedding model is shown in Table VI. HWE achieves the best performance (80.45%) over previous baselines the skip-gram (79.63%), TWE-1 (79.13%), and SWE (80.18%). With the same added knowledge of POS, Sense, Sense-WSD, or Topic, HWE performs better than TWE in 3 out of 4 experiments, showing the benefit of the proposed modeling of semantic information. Of this semantic knowledge, topic modeling is found to be the most useful for the sentence sentiment classification task.

V. Further Analysis

A. Visualization

First, we randomly select twelve verb pairs that belong to the *past to progressive* relation from the word analogy, such as 'ran' to 'running' and 'moved' to 'moving'. We then project those trained 300-dimension word embeddings into two dimension space with Principal Components Analysis (PCA). As shown in Figure 3, these relations trained by the HWE-POS model in Figure 3b are more smooth and consistent than those trained by the skip-gram in Figure 3a. In our HWE-POS model, the past verbs and the progressive verbs tend to be at the top-left and bottom-left side respectively, while these verbs are messy in the Skip-gram model. This shows that the HWE model takes advantage of the syntactically feature, Part-Of-Speech (POS), to correct the error relations. This phenomenon accounts for the good performance of HWE-POS model in the task of word analogy.

In addition, we visualize the effects of topic information for sentiment analysis besides the improvements shown in Section IV-F. We randomly select positive and negative affective terms proposed in [42]. Figure 3c and Figure 3d show the word vectors of the skip-gram and HWE-topic projected to $\mathbb{R}^2$ by PCA. As the skip-gram is known to cluster words that are functionally similar, many positive and negative words are clustered together. By using information provided by topic modeling, HWE is able to make them linearly separable.

B. Heterogeneous Features versus Model Convergence

With the same parameter setting as the dependency parsing task, we investigate the effect of the iteration to the performance. We train a 50-dimensional word embeddings for each model, without concatenating the feature embeddings.

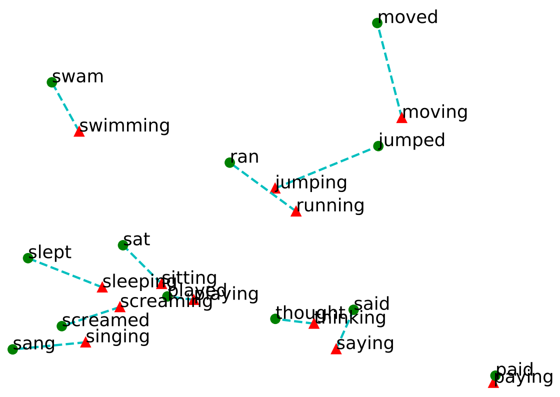
As shown in Figure 2, the HWE-POS model converges the fastest with fewer iterations, since the syntactical features contribute to this task significantly. HWE-POS and HWE-WSD reaches the better performance than the skip-gram model the with more iterations.

VI. Related Work

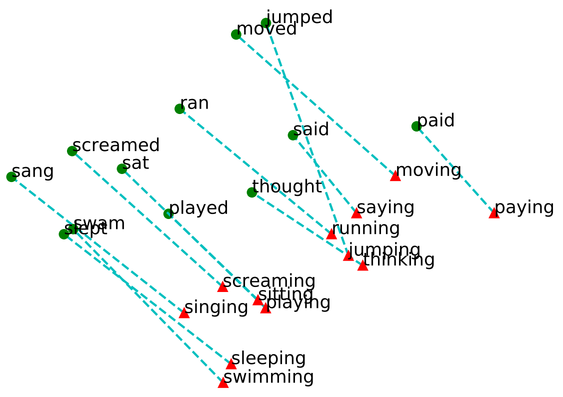
To enhance the quality of the word representations which learn from the co-occurrence information, some approaches have been proposed to incorporate other information into word representations.

Yu et al. [13] have proposed a new objective function which combines the CBOW with the synonymous words in knowledge to improve word embedding. To take advantage of knowledge more effectively, Liu et. al. [14] have proposed a more flexible method to incorporate semantic knowledge into the skip-gram. Their approach leverage semantic constraints in the optimization to learn more sensible word embeddings. Besides, a retrofitting approach, which adopts lexicons relation information in a post-processing stage, has been proposed by Faruqui et al. [15].

Niu et al. [16] have proposed a model Topic2Vec which incorporates topics information into the Word2vec for learning distributed representations of topics and words in the same semantic space. Liu et. al. [17] proposed Topical Word Embedding (TWE) which adopt a similar approach to combine topics information into the skip-gram. However, the TWE learn topic embeddings and word embeddings separately on the same corpus. Li et al. [18] refer that the TWE model, which combines the results of word embedding and LDA, lacks statistical foundations. Hence, they have proposed a generative model which combines word embedding and LDA, namely TopicVec. It leverages an embedding link function

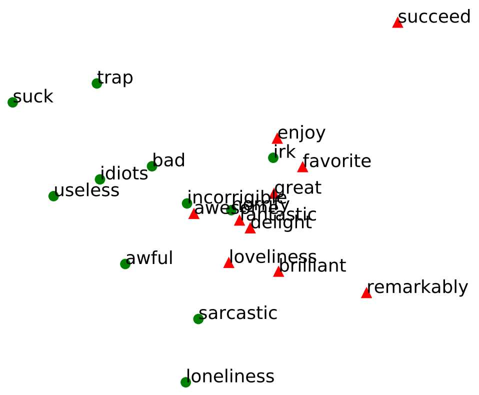


(a) Skip-gram (past verb to progressing verb)

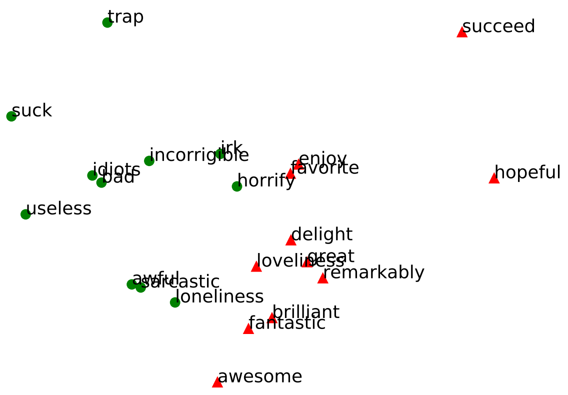


(b) HWE-POS (past verb to progressing verb)

Fig. 3: (Top) 2-dimensional PCA projection for word pairs holding the *past verb to progressing verb* relations.



(c) Skip-gram (negative vs. positive)



(d) HWE-Topic (negative vs. positive)

(Bottom) 2-dimensional PCA projection for the *negative and positive affective* terms.

which models the word distribution in a topic extraordinary for learning embeddings.

Liu et al. [43] refer that very few researches use the part-of-speech information for learning distributed word representation. Hence, they propose a new approach for learning word embeddings with weighted contexts based on part-of-speech (POS) relevance weights.

VII. Conclusion

We propose an effective and flexible method to improve word representations, termed HWE. The experimental results show that our model outperforms various state-of-the-art approaches. We also find that different feature settings for different particular tasks yield better performance. Moreover, through the concatenation of HWE and the corresponding feature embeddings, one can yield even more significant improvements. We conclude that HWE-sense works best for word similarity and semantic analogy, HWE-POS for syntactic analogy and dependency parsing, and HWE-Topic for document classification and sentiment analysis. The package and the trained embeddings is released on GitHub[5].

For future work, we aim to integrate more types of features in our model to learn word representations that are powerful enough to be used for multiple tasks. We would also like to investigate the effectiveness of using domain-specific features, which are expected to further improve tasks highly related to specific domains and knowledge, such as biomedical or law-related NLP tasks.

[5]https://github.com/emfomy/hwe/

References

[1] G. E. Hinton, J. L. McClelland, D. E. Rumelhart *et al.*, *Distributed representations*. Carnegie-Mellon University Pittsburgh, PA, 1984.

[2] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[3] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[4] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.

[5] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning," in *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 384–394.

[6] R. Socher, J. Bauer, C. D. Manning *et al.*, "Parsing with compositional vector grammars," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2013, pp. 455–465.

[7] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. M. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation." in *ACL (1)*, 2014, pp. 1370–1380.

[8] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[9] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[10] R. Lebret and R. Collobert, "Word emdeddings through hellinger pca," *arXiv preprint arXiv:1312.5542*, 2013.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[12] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[13] M. Yu and M. Dredze, "Improving lexical embeddings with semantic knowledge." in *ACL (2)*, 2014, pp. 545–550.

[14] Q. Liu, H. Jiang, S. Wei, Z.-H. Ling, and Y. Hu, "Learning semantic word embeddings based on ordinal knowledge constraints." in *ACL (1)*, 2015, pp. 1501–1511.

[15] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, "Retrofitting word vectors to semantic lexicons," in *The 2015 Annual Conference of the North American Chapter of the ACL*, 2015, p. 16061615.

[16] L. Niu, X. Dai, J. Zhang, and J. Chen, "Topic2Vec: learning distributed representations of topics," in *Asian Language Processing (IALP), 2015 International Conference on*. IEEE, 2015, pp. 193–196.

[17] Y. Liu, Z. Liu, T.-S. Chua, and M. Sun, "Topical word embeddings." in *AAAI*, 2015, pp. 2418–2424.

[18] S. Li, T.-S. Chua, J. Zhu, and C. Miao, "Generative topic embedding: a continuous representation of documents." in *ACL (1)*, 2016.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[20] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[21] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit." in *ACL (System Demonstrations)*, 2014, pp. 55–60.

[22] L. Tan, "Pywsd: Python implementations of word sense disambiguation (wsd) technologies [software]," https://github.com/alvations/pywsd, 2014.

[23] T. Koo, X. Carreras Pérez, and M. Collins, "Simple semi-supervised dependency parsing," in *46th Annual Meeting of the Association for Computational Linguistics*, 2008, pp. 595–603.

[24] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *arXiv preprint arXiv:1206.6426*, 2012.

[25] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda, "English gigaword fifth edition, linguistic data consortium," Technical report, Technical Report. Linguistic Data Consortium, Philadelphia, Tech. Rep., 2011.

[26] A. K. McCallum, "Mallet: A machine learning for language toolkit," 2002, http://mallet.cs.umass.edu.

[27] T. Griffiths, "Gibbs sampling in the generative model of latent dirichlet allocation," 2002.

[28] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity," *Language and cognitive processes*, vol. 6, no. 1, pp. 1–28, 1991.

[29] E. Bruni, N.-K. Tran, and M. Baroni, "Multimodal distributional semantics." *J. Artif. Intell. Res.(JAIR)*, vol. 49, no. 2014, pp. 1–47, 2014.

[30] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch, "A word at a time: computing word relatedness using temporal semantic analysis," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 337–346.

[31] G. Halawi, G. Dror, E. Gabrilovich, and Y. Koren, "Large-scale learning of word relatedness with constraints," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1406–1414.

[32] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Communications of the ACM*, vol. 8, no. 10, pp. 627–633, 1965.

[33] T. Luong, R. Socher, and C. D. Manning, "Better word representations with recursive neural networks for morphology." in *CoNLL*, 2013, pp. 104–113.

[34] F. Hill, R. Reichart, and A. Korhonen, "Simlex-999: Evaluating semantic models with (genuine) similarity estimation," *Computational Linguistics*, vol. 41, no. 4, pp. 665–695, 2015.

[35] S. Baker, R. Reichart, and A. Korhonen, "An unsupervised model for instance level subcategorization acquisition." in *EMNLP*, 2014, pp. 278–289.

[36] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 406–414.

[37] D. Yang and D. M. Powers, "Verb similarity on the taxonomy of wordnet," in *The Third International WordNet Conference: GWC 2006*. Masaryk University, 2006.

[38] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa, "A study on similarity and relatedness using distributional and wordnet-based approaches," in *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, pp. 19–27.

[39] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740–750.

[40] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2013, pp. 1631–1642. [Online]. Available: http://www.aclweb.org/anthology/D13-1170

[41] A. Komninos and S. Manandhar, "Dependency based embeddings for sentence classification tasks," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016, pp. 1490–1500. [Online]. Available: http://www.aclweb.org/anthology/N16-1175

[42] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 168–177.

[43] Q. Liu, Z.-H. Ling, H. Jiang, and Y. Hu, "Part-of-speech relevance weights for learning word embeddings," *arXiv preprint arXiv:1603.07695*, 2016.