GenSense: A Generalized Sense Retrofitting Model

Yang-Yin Lee[1*], Ting-Yu Yen[1*], Hen-Hsen Huang[1], Yow-Ting Shiue[1], and Hsin-Hsi Chen[1,2]
[1]Department of Computer Science and Information Engineering
National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei, 10617 Taiwan
[2]MOST Joint Research Center for AI Technology and All Vista Healthcare, Taiwan
{yylee, tyyen, hhhuang}@nlg.csie.ntu.edu.tw,
orina1123@gmail.com, hhchen@ntu.edu.tw

Abstract

With the aid of recently proposed word embedding algorithms, the study of semantic similarity has progressed and advanced rapidly. However, many natural language processing tasks need sense level representation. To address this issue, some researches propose sense embedding learning algorithms. In this paper, we present a generalized model from the existing sense retrofitting model. The generalization takes three major components: semantic relations between the senses, the relation strength and the semantic strength. In the experiments, we show that the generalized model outperforms the previous approaches in three aspects: semantic relatedness, contextual word similarity and semantic difference.

1 Introduction

The distributed representation of word model (word embedding) has drawn great interest in recent years due to its ability to acquire syntactic and semantic information from a large unannotated corpus (Mikolov et al., 2013; Pennington et al., 2014). With the pre-trained word embedding, some researches propose post-processing models that incorporate with the existing semantic knowledge into the word embedding model (Faruqui et al., 2015; Yu and Dredze, 2014). However, word embedding models use only one vector to represent a word, and are problematic in some natural language processing applications that require sense level representation (e.g., word sense disambiguation, semantic relation identification, etc.). As a result, some researches try to resolve the polysemy and homonymy issue and introduce sense level embedding, either act as pre-process (Iacobacci et al., 2015) or post-process (Jauhar et al., 2015) fashion.

In this research, we focus on the post-processing sense retrofitting approach and propose *GenSense*, a generalized sense embedding learning framework that retrofits a pre-trained word embedding via incorporating with the semantic relations between the senses, the relation strength and the semantic strength. Although some parts of the idea are not new, it is the first time of putting all the parts into a generalized framework. Our proposed *GenSense* for generating low-dimensional sense embedding is inspired from *sense retro* (Jauhar et al., 2015), but has three major differences. First, we generalize the semantic relations from positive relations (e.g., synonyms, hyponyms, paraphrase, etc.) to positive and negative relations (e.g., antonyms). Second, each relation incorporates with both the semantic strength and the relation strength. Within a semantic relation, there should be a weighting for each semantic strength. For example, although *jewel* has the synonyms *gem* and *rock*, it is clear that the similarity between (*jewel*, *gem*) is higher than (*jewel*, *rock*), and thus (*jewel*, *gem*) should have higher weight. Last, *GenSense* gives different relations with different relation strengths. For example, if the objective is to train a sense embedding that can distinguish between the positive and negative sense, then the weight for the negative relation (e.g., antonyms) should be higher, and vice versa. The experimental results suggest the relation strengths play a role in balancing the relations and are application dependent. With an objective that considers these three parts, the sense vectors can be learned and updated via running a belief propagation process on the relation constrained network.

*These authors contributed equally to this work.

Proceedings of the 27th International Conference on Computational Linguistics, pages 1662–1671
Santa Fe, New Mexico, USA, August 20-26, 2018.

In the experiments, we show that our proposed *GenSense* model outperforms the previous approaches in three types of datasets: semantic relatedness, contextual word similarity and semantic difference. While the generalized model of considering all the relations performs well in the semantic and relatedness tasks, we also find that the antonym relation is in favor of the semantic difference experiment. The remainder of this paper is organized as follows. Section 2 gives a survey on the related works. Section 3 defines the generalized sense retrofitting model. The experimental setup is in Section 4. Section 5 shows and discusses the experimental results. Section 7 concludes the remarks.

2 Related Works

The study of the representation of words has a long history. Early approaches include utilizing the term-document occurrence matrix from a large corpus and then perform dimension reduction techniques such as singular value decomposition (latent semantic analysis) (Bullinaria and Levy, 2007; Deerwester et al., 1990). Beyond that, recent word embedding approaches are more focus on neural-style (Dragoni and Petrucci, 2017; Mikolov et al., 2013; Pennington et al., 2014) and performs well on syntactic and semantic tasks. Apart from the unsupervised word embedding learning models, there are plenty of ontologies that contain lexical knowledge, such as WordNet (Fellbaum, 1998), Roget's 21st Century Thesaurus (Kipfer and Institute, 1993) or the paraphrase database (Pavlick et al., 2015). As a result, many researches combine the word embedding with ontological resources, either in a joint training (Bian et al., 2014; Liu et al., 2016; Yu and Dredze, 2014) or a post-processing (Faruqui et al., 2015) fashion. When the need for sense embedding is getting higher, some researches are inspired from the word level embedding learning model and propose sense level embedding (Iacobacci et al., 2015; Jauhar et al., 2015; Lee and Chen, 2017). Although some evidence shows that the sense embedding cannot improve every natural language processing task (Li and Jurafsky, 2015), the benefit of having a sense embedding for improving tasks that need sense level representation is still in great need (Azzini et al., 2012; Ettinger et al., 2016; Qiu et al., 2016).

3 Generalized Sense Retrofitting Model

Let $V = \{w_1, \dots, w_n\}$ be a vocabulary of a trained word embedding and $|V|$ be its size. The matrix $\hat{Q}$ will be the pre-trained collection of vector representations $\hat{q}_i \in \mathbb{R}^d$, where d is the dimensionality of a word vector. Each $w_i \in V$ is learned using a standard word embedding technique (e.g., GloVe (Pennington et al., 2014) or Word2Vec (Mikolov et al., 2013)). Let $\Omega = (T, E)$ be an ontology that contains the semantic relationship, where $T = \{t_1, \dots, t_m\}$ is a set of senses and $|T|$ is total number of senses. The edge $(i, j) \in E$ indicates a semantic relationship of interest (e.g., synonym) between t_i and t_j. In our scenario, the edge set E consists of several disjoint subsets of interest (i.e., $E = E_{s_1} \cup E_{s_2} \cup \dots \cup E_{s_k}$). For example, $(i, j) \in E_{s_1}$ if and only if t_j is the synonym of t_i. We use $\hat{q}_{t_j}$ to denote the word form vector of t_j (one should notice that $\hat{q}_{t_j}$ and $\hat{q}_{t_k}$ may map to the same vector representation even if $j \neq k$). Then the goal is to learn a new matrix $S = (s_1, \dots, s_m)$ such that each new sense vector is close to its word form vertex and its synonym neighbors. The basic form that considers only synonym relation for the objective of the sense retrofitting model is:

$$\sum_{i=1}^{m} \left[\alpha_1 \beta_{ii} \left\| s_i - \hat{q}_{t_i} \right\|^2 + \alpha_2 \sum_{(i,k) \in E_{s_1}} \beta_{ij} \| s_i - s_k \|^2 \right] \quad (1)$$

where α balances the importance of the word form vertex and the synonym, and βs control the strength of the semantic relations. From equation 1, the learned new sense vectors will close to its synonyms, meanwhile constraining its distance with its original word form vector. In addition, this equation can be further generalized to consider all the relations:

$$\sum_{i=1}^{m} \left[\alpha_1 \beta_{ii} \left\| s_i - \hat{q}_{t_i} \right\|^2 + \alpha_2 \sum_{(i,k) \in E_{s_1}} \beta_{ij} \| s_i - s_k \|^2 + \cdots \right] \quad (2)$$

Apart from the positive sense relation, we now introduce three types of special relations. The first one is the positive contextual neighbor relation s_2. $(i, j) \in E_{s_2}$ if and only if t_j has only one sense. In our model, we use the word form vector to represent the neighbors of the t_is in E_{s_2}. Those neighbors

are viewed as positive contextual neighbors as they learned from the context of a corpus (e.g., word2vec trained with Google News corpus) with positive meaning. The second is the negative sense relation s_3 (e.g., antonym). The negative senses are used in a subtraction fashion for pushing the sense away from the positive meaning. The last is the negative contextual neighbors s_4. Just like the positive contextual neighbors, the negative contextual neighbors were learned from the context of a corpus, but with negative meaning.

Figure 1 illustrates an example of the relation network. In Figure 1, *gay* may have two meanings: (1) bright and pleasant; promoting a feeling of cheer and (2) someone who is sexually attracted to persons of the same sex. If we focus on the first sense, then our model can attract s_{gay_1} to its word form vector $\hat{q}_{gay_1}$, its synonym s_{glad_1} its positive contextual neighbor $\hat{q}_{jolly}$. But in the same time, it will push s_{gay_1} from its antonym s_{sad_1} and its negative contextual neighbor $\hat{q}_{dull}$.

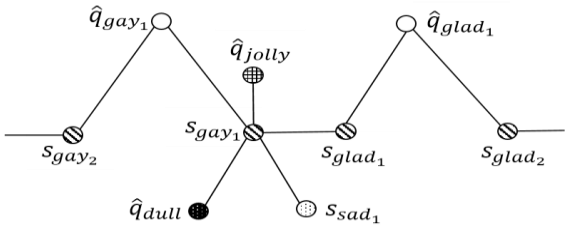

Figure 1. An illustration of the relation network. Different textures of the nodes represent different roles (e.g., synonym, antonym, etc.) in the GenSense model.

To formulize the above scenario and consider all the parts, the equation 2 would become:

$$\sum_{i=1}^{m}\left[\alpha_1\beta_{ii}\left\|s_i-\hat{q}_{t_i}\right\|^2+\alpha_2\sum_{(i,j)\in E_{s_1}}\beta_{ij}\left\|s_i-s_j\right\|^2+\alpha_3\sum_{(i,j)\in E_{s_2}}\beta_{ij}\left\|s_i-\hat{q}_j\right\|^2-\alpha_4\sum_{(i,j)\in E_{s_3}}\beta_{ij}\left\|s_i-s_j\right\|^2-\alpha_5\sum_{(i,j)\in E_{s_4}}\beta_{ij}\left\|s_i-\hat{q}_j\right\|^2\right] \quad (3)$$

We therefore apply an iterative updating method to the solution of the above convex objective function (Bengio et al., 2006). Initially, the sense vectors are set to their corresponding word form vectors (i.e., $s_i \leftarrow \hat{q}_{t_i}\ \forall i$). Then in the following iterations, the updating formula for s_i would be:

$$s_i=\frac{-\alpha_5\sum_{j:(i,j)\in E_{s_4}}\beta_{ij}\hat{q}_j-\alpha_4\sum_{j:(i,j)\in E_{s_3}}\beta_{ij}s_j+\alpha_3\sum_{j:(i,j)\in E_{s_2}}\beta_{ij}\hat{q}_j+\alpha_2\sum_{j:(i,j)\in E_{s_1}}\beta_{ij}s_j+\alpha_1\beta_{ii}\hat{q}_{t_i}}{-\alpha_5\sum_{j:(i,j)\in E_{s_4}}\beta_{ij}-\alpha_4\sum_{j:(i,j)\in E_{s_3}}\beta_{ij}+\alpha_3\sum_{j:(i,j)\in E_{s_2}}\beta_{ij}+\alpha_2\sum_{j:(i,j)\in E_{s_1}}\beta_{ij}+\alpha_1\beta_{ii}} \quad (4)$$

A formal description of our proposed GenSense method is shown in Algorithm 1. In Algorithm 1, the β parameters are retrieved from the ontology. The ϵ is a threshold for deciding whether to update the sense vector or not, which is used as a stopping criteria when the difference between the new sense vector and the original sense vector is too small. Experimentally, 10 iterations are sufficient to minimize the objective function from a set of starting vectors to produce effective sense retrofitted vectors.

4 Experiments

We evaluate GenSense with three types of experiments: semantic relatedness, contextual word similarities, and semantic difference. In testing phase, if a test dataset has missing words, we use the average of all sense vectors to represent the missing word. Note that our reported results of vanilla sense embedding may be slightly different from the other researches due to the treatment of missing words and the similarity computation method. Some researches use zero vector to represent the missing words, whereas some remove those missing words from the dataset. However, within this research the reported performance can be compared due to the same missing word processing method and the same similarity computation method.

Algorithm 1 GenSense

Input: A pre-trained word embedding $\hat{Q}$, a relation ontology $\Omega = (T, E)$, hyper-parameters α and parameters β, number of iterations max_it, the convergence criteria for sense vectors ϵ.

Output: A trained sense embedding S

for $i = 1$ **to** m **do**
  $s_i^{(0)} \leftarrow \hat{q}_{t_i}$
end for
for $it = 1$ **to** $max\_it - 1$ **do**
  for $i = 1$ **to** m **do**
    Compute s_i^{tmp} using equation (4).
    if $\left\| s_i^{tmp} - s_i^{(it-1)} \right\| \geq \epsilon$ **then**
      $s_i^{(it)} \leftarrow s_i^{tmp}$
    else
      $s_i^{(it)} \leftarrow s_i^{(it-1)}$
    end if
  end for
end for
return S

4.1 Experimental Setup

We adopt the GloVe model in our experiment (Pennington et al., 2014). The pre-trained GloVe word embedding is trained on Wikipedia and Gigaword-5 (6B tokens, 400k vocab, uncased, 50d vectors). Roget's 21st Century Thesaurus (Kipfer and Institute, 1993) (Roget) is selected for building ontology in our experiments as it contains the strength information of the senses. As Roget does not provide the ontology directly, we manually built a synonym ontology and an antonym ontology from the resource. The vocabulary from GloVe pre-trained word embedding is used for fetching and building the ontology from Roget. In Roget, there are three levels of synonym relations, we set βs to 1.0, 0.6 and 0.3 for the nearest to the farthest synonyms. The antonym relation is built in the same way. For each sense, β_{ii} is set to the sum of all the relation specific weights. Unless specifically address, αs are set to 1 in the experiments. We set the convergence criteria for sense vectors to $\epsilon = 0.1$ with the number of iterations of 10. With the capability of generalization, we run three types of the model: GenSense-syn (only considers the synonyms and positive contextual neighbors), GenSense-ant (only considers the antonyms and negative contextual neighbors) and GenSense-all (considers everything).

4.2 Semantic Relatedness

We downloaded four semantic relatedness benchmark datasets from the web: MEN (Bruni et al., 2014), MTurk (Radinsky et al., 2011), Rare Words (RW) (Luong et al., 2013) and WordSim353 (WS353) (Finkelstein et al., 2001). In MEN dataset, there are two versions of the word pairs: lemma and natural form. We show the natural form in the experimental result, but the performances on two datasets are similar. In each dataset, there is a list of word pairs together with their corresponding human rated scores. A higher score value indicates higher semantic similarity. For example, the score of (*journey*, *voyage*) is 9.29 and the score of (*king*, *cabbage*) is 0.23 in WS353. For measuring the semantic similarity between a word pair (w, w') in the datasets, we adopt the sense evaluation metrics AvgSim and MaxSim (Reisinger and Mooney, 2010):

$$\text{AvgSim}(w, w') \stackrel{\text{def}}{=} \frac{1}{K_w K_{w'}} \sum_{j=1}^{K_w} \sum_{k=1}^{K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right) \tag{5}$$

$$\text{MaxSim}(w, w') \stackrel{\text{def}}{=} \max_{1 \le j \le K_w, 1 \le k \le K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right) \tag{6}$$

where K_w and $K_{w'}$ denote the number of senses of w and w', respectively. The AvgSim can be seen as a *soft* metric as it averages all the similarity scores. Whereas the MaxSim can be seen as a *hard* metric as it only selects the senses with maximum similarity score. For measuring the performance of the sense embedding, we compute the spearman correlation between the human rated scores and the AvgSim/MaxSim scores. Table 1 shows a summary of the benchmark datasets and their relationship with the ontologies. In Table 1, row 3 shows the number of words that are both listed in the datasets and the ontology. The word count in Roget is 63,942.

	MEN	**MTurk**	**RW**	**WS353**
Pair count	3,000	287	2,034	353
Word count	751	499	2,951	437
Roget	707	416	2,371	412

Table 1. A summary of the semantic relatedness benchmark datasets.

4.3 Contextual Word Similarity

Although the semantic relatedness datasets are used in many researches, one major disadvantage is that the words in those word pairs do not have contexts. Therefore, we also conduct experiments with the Stanford's Contextual Word Similarities (SCWS) dataset (Huang et al., 2012). SCWS consists of 2,003 word pairs together with human rated scores. A higher score value indicates higher semantic similarity. Different from the semantic relatedness datasets, the words in the SCWS have their contexts and part-of-speech tags. That is, the human subjects can know the usage of the word when they rate the similarity. For each word pair, we compute its AvgSimC/MaxSimC scores from the learned sense embedding (Reisinger and Mooney, 2010):

$$\text{AvgSimC}(w, w') \stackrel{\text{def}}{=} \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{\text{K}} d_{c,w,k} d_{c',w',j} d\left(\pi_k(w), \pi_j(w')\right) \tag{7}$$

$$\text{MaxSimC}(w, w') \stackrel{\text{def}}{=} d\left(\hat{\pi}(w), \hat{\pi}(w')\right) \tag{8}$$

where $d_{c,w,k} \stackrel{\text{def}}{=} d\left(v(c), \pi_k(w)\right)$ is the likelihood of context c belonging to cluster π_k, and $\hat{\pi}(w) \stackrel{\text{def}}{=} \pi_{\arg\max_{1 \le k \le K} d_{c,w,k}}(w)$, the maximum likelihood cluster for w in context c. We use a window size of 5 for the words in the word pairs (i.e., 5 words prior to w/w' and 5 words after w/w'). Stop words are removed from the context. For measuring the performance, we compute the spearman correlation between the human rated scores and the AvgSimC/MaxSimC scores.

4.4 Semantic Difference

This task is defined to answer if a word has a closer semantic feature to a concept than another word (Krebs and Paperno, 2016). In this dataset, there are 528 concepts, 24,963 word pairs, and 128,515 items. Each word pair comes with a feature. For example, in the test $(airplane, helicopter)$: $wings$, choosing the first word if and only if $\cos(airplane, wings) > cos(helicopter, wings)$, otherwise, choose the second word. As this dataset does not provide context for disambiguation, we use the similar strategies from the semantic relatedness task:

$$\text{AvgSimD}(w, w') \stackrel{\text{def}}{=} \frac{1}{K_w K_{w'}} \sum_{j=1}^{K_w} \sum_{k=1}^{K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right) \tag{9}$$

$$\text{MaxSimD}(w, w') \stackrel{\text{def}}{=} \max_{1 \le j \le K_w, 1 \le k \le K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right) \tag{10}$$

In AvgSimD, we choose the first word iff $\text{AvgSimD}(w_1, w') > \text{AvgSimD}(w_2, w')$. In MaxSimD, we choose the first word iff $\text{MaxSimD}(w_1, w') > \text{MaxSimD}(w_2, w')$. The performance is determined by computing the accuracy.

5 Results and Discussion

Table 2 shows the spearman correlation ($\rho \times 100$) of AvgSim and MaxSim between human scores and sense embedding's scores on each benchmark dataset. Row 2 shows the performance of vanilla GloVe word embedding. Note that the MaxSim and AvgSim scores will be the same when there is only one sense for each word (word embedding). Row 3 shows the performance of the retro model (Jauhar et al., 2015).

	MEN	**MTurk**	**RW**	**WS353**	Macro	Micro
GloVe	65.7	61.9	30.3	50.3	52.1	51.9
retro	62.4/67.7	57.4/60.1	15.1/26.9	43.9/51.1	44.7/51.5	44.0/51.6
GenSense-syn	67.6/67.9	64.1/64.0	**33.8/33.6**	50.5/52.8	54.0/54.6	54.3/54.5
GenSense-ant	65.1/65.0	62.1/63.1	31.0/30.9	48.4/47.1	51.6/51.5	51.7/51.6
GenSense-all	**68.8/68.6**	**65.1/64.8**	33.3/33.2	**53.2/54.0**	**55.1/55.2**	**54.9/54.8**

Table 2. $\rho \times 100$ of (MaxSim / AvgSim) on semantic relatedness benchmark datasets.

From Table 2, we find that our proposed model outperforms the comparison models *retro* and GloVe in all the datasets. When comparing our model with retro, the spearman correlation scores of MaxSim of each dataset grows at least 6.4. In RW, the spearman correlation score of GenSense exceed *retro* by 18.2. We also discover a significant growth of spearman correlation between GenSense-syn and GenSense-all. Surprisingly, the model that only adopts synonyms and positive contextual information can outperform *retro* and GloVe. After utilizing antonym knowledge from Roget, its performance can further be improved in all but RW dataset. This result supports an assumption that the antonyms in Roget are quite informative and useful. Moreover, GenSense can adapt information from synonyms and antonyms to boost its performance. Although our model can pull sense vectors away from its reverse sense with the help of antonym and negative contextual information. This shift cannot guarantee the new sense vectors will move to a better place with only negative relations. As a result, the GenSense-ant does not perform as well as GenSense-syn. Table 2 also shows the macro-averaged and micro-averaged results in the rightmost two columns. In both of the additional evaluation metrics, we find that the GenSense model outperforms *retro* with a large margin. These two metrics suggest the robustness of our proposed model when comparing to the *retro* model.

We also conduct an experiment to test how much benefit we can get from the relation strength. We run GenSense-syn over the Roget ontology with a grid of $(\alpha_1, \alpha_2, \alpha_3)$ parameters. Each parameter is tuned from 0.0 to 1.0 with a 0.1 step size. Table 3 shows the results with MaxSim metric and Table 4 shows the results with AvgSim metric. Note that the $\alpha_1/\alpha_2/\alpha_3$ parameter combinations of the worst or the best case may be more than one. In that case, we only report one $\alpha_1/\alpha_2/\alpha_3$ setting in Table 3 and Table 4 due to the space limitation. From Table 3, we find that the default setting can achieve relatively good results when comparing to the best case. Another point worth mentioning is that the worst performance happens under .1/1./.1 setting except the WS353 dataset. Similar results can be found in Table 4's AvgSim metric. The results demonstrate the importance of the original word vector and the synonyms sense vectors in the model.

	MEN	$\alpha_1/\alpha_2/\alpha_3$	**MTurk**	$\alpha_1/\alpha_2/\alpha_3$	**RW**	$\alpha_1/\alpha_2/\alpha_3$	**WS353**	$\alpha_1/\alpha_2/\alpha_3$
GenSense default	67.6	1./1./1.	64.1	1./1./1.	33.8	1./1./1.	50.5	1./1./1.
GenSense worst	52.4	.1/1./.1	50.3	.1/1./.1	25.1	.1/1./.1	34.8	.1/1./.8
GenSense best	68.1	.8/.5/.8	64.4	.4/.3 /.4	35.8	.1/.1/1.	52.0	1./.6/.3

Table 3. $\rho \times 100$ of MaxSim on semantic relatedness benchmark.

	MEN	$\alpha_1/\alpha_2/\alpha_3$	MTurk	$\alpha_1/\alpha_2/\alpha_3$	RW	$\alpha_1/\alpha_2/\alpha_3$	WS353	$\alpha_1/\alpha_2/\alpha_3$
GenSense default	67.9	1./1./1.	64.0	1./1./1.	33.6	1./1./1.	52.8	1./1./1.
GenSense worst	60.1	.1/1./.1	58.7	.1/1./.1	30.5	.1/1./.1	43.3	.1/1./1.
GenSense best	68.1	.5/.5/.8	64.2	.3/.5/.4	35.8	.1/.1/1.	53.1	.5/.5/.8

Table 4. $\rho \times 100$ of AvgSim on semantic relatedness benchmark datasets.

Figure 2 shows the $\rho \times 100$ of MaxSim on the semantic relatedness benchmark datasets as function of vector dimension. All GloVe pre-trained models are trained on the 6 billion tokens corpus of 50d, 100d, 200d and 300d. We use the GenSense-all model on the GloVe pre-trained models. Figure 2 shows the proposed GenSense-all outperforms GloVe in all the datasets of all the tested dimensions. In GloVe's original paper, they showed GloVe's performance (in terms of accuracy) is proportional to the dimension in the range within 50d and 300d. In this experiment, we show that both GloVe and GenSense-all's performance is proportional to the dimension in the range within 50d and 300d in terms of $\rho \times 100$ of MaxSim. Similar results can be found in the AvgSim metric.

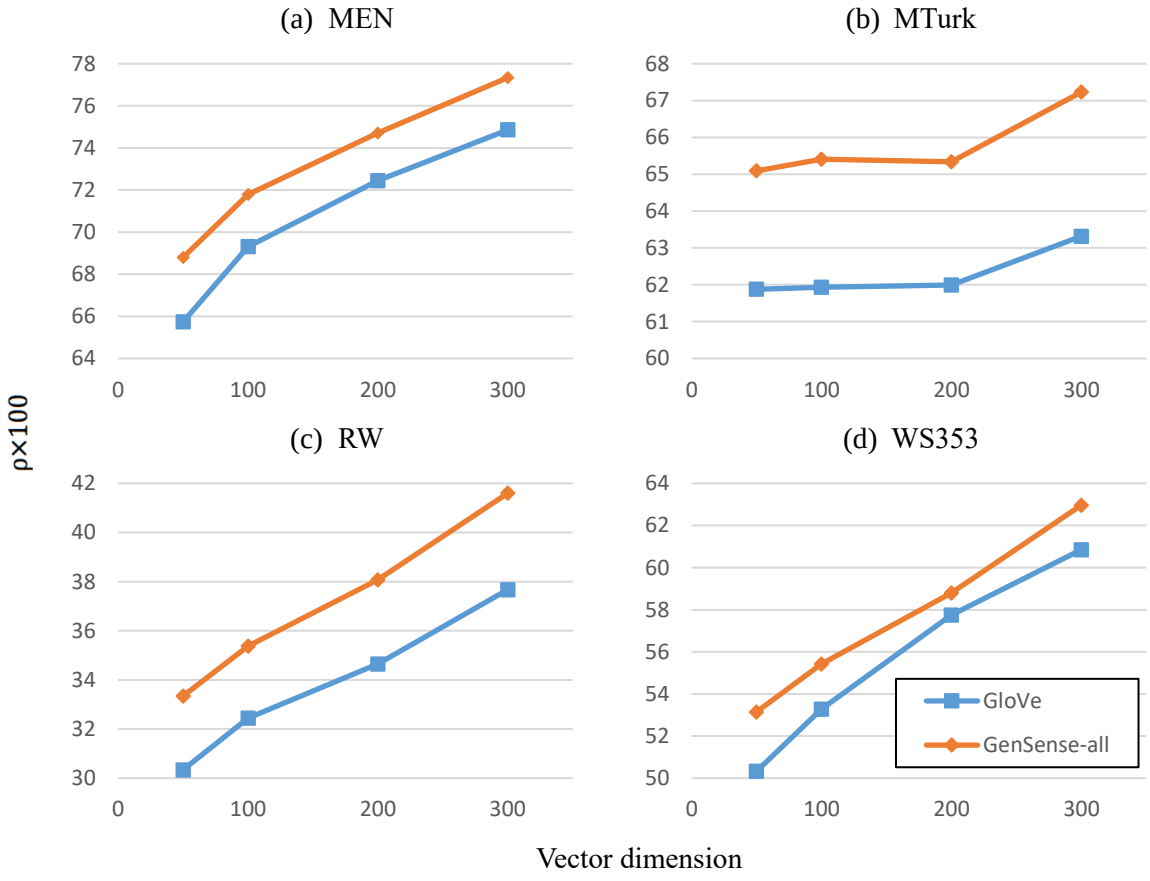

Figure 2. $\rho \times 100$ of MaxSim on semantic relatedness benchmark datasets as function of vector dimension. GloVe model is compared.

Table 5 shows the selected MEN's word pairs and its corresponding GenSense-all, GloVe and retro scores for case study. For GenSense-all, GloVe and retro, we use the MaxSim scores and then sort and re-scale to MEN's score distribution. From Table 5, we find that Gensense-all can improve pre-trained word-embedding model (in terms of closeness to MEN's score, smaller is better) in the following situations: (1) both words have a few senses (*lizard*, *reptiles*), (2) both words have many senses (*stripes*, *train*) and (3) one word has many senses and one word has a few senses (*rail*, *railway*). Sometimes, the retro model increases the closeness to MEN's score. In other words, GenSense-all can handle all the possible situations well and outperform *retro*.

Word pair	#senses	GenSense-all	GloVe	retro
(*rail*, *railway*)	(15, 2)	1	3	36
(*stripes*, *train*)	(20, 17)	2	6	23
(*pregnant*, *women*)	(3, 4)	0	8	17
(*curve*, *dance*)	(10, 7)	2	6	25
(*blue*, *happy*)	(16, 4)	0	5	21
(*dripping*, *round*)	(5, 25)	0	4	24
(*nails*, *wolf*)	(10, 6)	2	6	26
(*action*, *truck*)	(12, 3)	0	9	19
(*lizard*, *reptiles*)	(2, 1)	7	13	28
(*amphibians*, *lizard*)	(3, 2)	9	16	34

Table 5. Selected MEN's word pairs and their scores difference with GenSense-all, GloVe and retro models. (the smaller the better).

Table 6 shows the spearman correlation ($\rho \times 100$) of Stanford's Contextual Word Similarity dataset. With the sense level information, both GenSense and retro can outperform the word embedding model GloVe. The GenSense model performs slightly better than *retro*. Again, we find that the retrofitting model cannot benefit with only negative relation information.

	SCWS
GloVe	52.9
retro	54.2/55.9
GenSense-syn	**54.8/56.0**
GenSense-ant	52.9/52.7
GenSense-all	54.2/55.3

Table 6. $\rho \times 100$ of (MaxSimC / AvgSimC) on SCWS dataset.

Table 7 shows the results of the semantic difference experiment. From Table 7, we find that GenSense outperforms *retro* and GloVe. The accuracy of *retro* declines in this experiment. This finding demonstrates the effectiveness and robustness of our proposed framework. Surprisingly, the antonym relation plays an important role when computing the semantic difference.

	Accuracy	Precision	Recall
GloVe	58.5	53.3	59.4
retro	57.5/57.3	52.2/51.9	58.0/52.0
GenSense-syn	57.8/57.6	52.5/52.3	61.2/59.8
GenSense-ant	58.0/**58.7**	52.7/**53.3**	59.7/**61.7**
GenSense-all	**58.7**/57.6	**53.3**/52.3	**62.4**/61.0

Table 7. $(\text{Accuracy}, \text{Precision}, \text{Recall}) \times 100$ of (MaxSimD / AvgSimD) on the semantic difference dataset.

6 Conclusion

In this paper we present *GenSense*, a generalized framework for learning sense embedding. The generalization takes in three parts: (1) we extend the synonym relation to positive contextual neighbor relation, antonym relation and negative contextual neighbor relation; (2) within each relation, we consider the semantic strength; and (3) we use relation strength between relations to balance different components. We then conduct experiments in three types of experiments: semantic relatedness, contextual word similarity, and semantic difference and show that the GenSense model outperforms the previous approaches. In the future, one of the possible applications is to apply the generalized sense

representation learnt by the proposed method in downstream natural language processing applications to conduct extrinsic evaluations. We release the source code and the pre-trained model as resource for the research community. [12] Other versions of the sense retrofitted embeddings can be found in the website.

Acknowledgements

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-107-2634-F-002-019, MOST-107-2634-F-002-011 and MOST-106-2923-E-002-012-MY3.

Reference

Antonia Azzini, Célia da Costa Pereira, Mauro Dragoni, and Andrea GB Tettamanzi. 2012. A neuro-evolutionary corpus-based method for word sense disambiguation. *IEEE Intelligent Systems*, 27(6):26–35.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 2006. Label Propagation and Quadratic Criterion. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press.

Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Knowledge-powered deep learning for word embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 132–148. Springer.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *The Journal of Artificial Intelligence Research*, 49:1–47.

John A Bullinaria and Joseph P Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.

Scott C. Deerwester, Susan T Dumais, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Mauro Dragoni and Giulio Petrucci. 2017. A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Transactions on Affective Computing*, 8(4):457–470.

Allyson Ettinger, Philip Resnik, and Marine Carpuat. 2016. Retrofitting sense-specific word vectors using parallel text. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1378–1383.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* pages 873–882, Jeju, Republic of Korea.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association*

[1] http://nlg.csie.ntu.edu.tw/nlpresource/GenSense
[2] https://github.com/y95847frank/GenSense

for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 95–105, Beijing, China.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 683–693.

B.A. Kipfer and Princeton Language Institute. 1993. *Roget's 21st Century Thesaurus in Dictionary Form: The Essential Reference for Home, School, or Office*. Dell Pub.

Alicia Krebs and Denis Paperno. 2016. Capturing discriminative attributes in a distributional space: Task proposal. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 51–54.

Guang-He Lee and Yun-Nung Chen. 2017. MUSE: Modularizing unsupervised sense embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 327–337, Copenhagen, Denmark.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal.

Xiaojie Liu, Jian-Yun Nie, and Alessandro Sordoni. 2016. Constraining word embeddings by prior knowledge–application to medical information retrieval. In *Asia Information Retrieval Symposium*, pages 155–167.

Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevich, and Chris Callison-Burch Ben Van Durme. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 425–430, Beijing, China.*

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.

Lin Qiu, Kewei Tu, and Yong Yu. 2016. Context-dependent sense embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 183–191.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346.

Joseph Reisinger and Raymond J Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 545–550.