Natural Language Engineering (2023), **29**, pp. 1097–1125
doi:10.1017/S1351324922000523

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

On generalization of the sense retrofitting model

Yang-Yin Lee[1,*†], Ting-Yu Yen[1†], Hen-Hsen Huang[2], Yow-Ting Shiue[1] and Hsin-Hsi Chen[1]

[1]Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan and [2]Institute of Information Science, Academia Sinica, Taipei, Taiwan
*Corresponding author: E-mail: eric890006@gmail.com

(Received 10 August 2020; revised 2 November 2022; accepted 8 November 2022; first published online 31 March 2023)

Abstract

With the aid of recently proposed word embedding algorithms, the study of semantic relatedness has progressed rapidly. However, word-level representations are still lacking for many natural language processing tasks. Various sense-level embedding learning algorithms have been proposed to address this issue. In this paper, we present a generalized model derived from existing sense retrofitting models. In this generalization, we take into account semantic relations between the senses, relation strength, and semantic strength. Experimental results show that the generalized model outperforms previous approaches on four tasks: semantic relatedness, contextual word similarity, semantic difference, and synonym selection. Based on the generalized sense retrofitting model, we also propose a standardization process on the dimensions with four settings, a neighbor expansion process from the nearest neighbors, and combinations of these two approaches. Finally, we propose a Procrustes analysis approach that inspired from bilingual mapping models for learning representations that outside of the ontology. The experimental results show the advantages of these approaches on semantic relatedness tasks.

Keywords: Sense embedding; Retrofitting; Generalization; Semantic relatedness

1. Introduction

Models for the distributed representation of words (word embeddings) have drawn great interest in recent years because of their ability to acquire syntactic and semantic information from large unannotated corpora (Mikolov *et al.* 2013a; Pennington, Socher, and Manning 2014; Sun *et al.* 2016). Likewise, more and more ontologies have been compiled with high-quality lexical knowledge, including WordNet (Miller 1998), Roget's 21st Century Thesaurus (Roget) (Kipfer 1993), and the paraphrase database (PPDB) (Pavlick *et al.* 2015). Based on lexical knowledge, early linguistic approaches such as the Leacock Chodorow similarity measure (Leacock and Chodorow 1998), the Lin similarity measure (Lin 1998), and the Wu–Palmer similarity measure (Wu and Palmer 1994) have been proposed to compute semantic similarity. Although these linguistic resource-based approaches are somewhat logical and interpretable, they do not scale easily (in terms of vocabulary size). Furthermore, approaches based on modern neural networks outperform most linguistic resource-based approaches with better linearity.

While the recently proposed contextualized word representation models (Peters *et al.* 2018; Devlin *et al.* 2019; Radford *et al.* 2019) can have different representations given the context of a target word, evidence showed that the contextualized word representation may perform worse than static word embedding in some semantic relatedness datasets (Ethayarajh 2019). Moreover,

[†]These authors contributed equally to this work.

they may not be able to incorporate the knowledge in the ontologies into the models. On the contrary, some researches proposed models to incorporate word embedding models and lexical ontologies, using either joint training or post-processing (Yu and Dredze 2014; Faruqui *et al*. 2015). However, these 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 and semantic relation identification). One way to take into account such polysemy and homonymy is to introduce sense-level embedding, via either pre-processing (Iacobacci, Pilehvar, and Navigli 2015) or post-processing (Jauhar, Dyer, and Hovy 2015).

In this work, we focus on a post-processing sense retrofitting model GenSense (Lee *et al*. 2018), which is a generalized sense embedding learning framework that retrofits a pre-trained word embedding (i.e., Word2Vec Mikolov *et al*. 2013a, GolVe Pennington *et al*. 2014) with semantic relations between the senses, the relation strength, and the semantic strength.[a] The GenSense for generating low-dimensional sense embedding is inspired from the Retro sense model (Jauhar *et al*. 2015) but has three major differences. First, it generalizes semantic relations from positive relations (e.g., synonyms, hyponyms, paraphrasing Lin and Pantel 2001; Dolan, Quirk, and Brockett 2004; Quirk, Brockett, and Dolan 2004; Ganitkevitch, Van Durme, and Callison-Burch 2013; Pavlick *et al*. 2015) to both positive and negative relations (e.g., antonyms). Second, each relation incorporates both semantic strength and 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*); thus, a good model should assign a higher weight to (*jewel, gem*). Last, GenSense assigns different relation strengths to different relations. For example, if the objective is to train a sense embedding that distinguishes between positive and negative senses, then the weight for negative relations (e.g., antonyms) should be higher, and vice versa. Experimental results suggest that relation strengths play an important role in balancing relations and are application dependent. Given an objective function that takes into consideration these three parts, sense vectors can be learned and updated using a belief propagation process on the relation constrained network. A constraint on the update formula is also considered using a threshold criterion.

Apart from the GenSense framework, some work suggests using a standardization process to improve the quality of vanilla word embeddings (Lee *et al*. 2016). Thus, we propose a standardization process on GenSense's embedding dimensions with four settings, including (1) performing standardization after all of the iteration process (*once*); (2) performing standardization after every iteration (*every time*); (3) performing standardization before the sense retrofitting process (*once*); and (4) performing standardization before each iteration of the sense retrofitting process (*every time*). We also propose a sense neighbor expansion process from the nearest neighbors; this is added into the sense update formula to improve the quality of the sense embedding. Finally, we combine the standardization process and neighbor expansion process in four different ways: (1) GenSense with neighbor expansion, followed by standardization (*once*); (2) GenSense with neighbor expansion, followed by standardization in each iteration (*each time*); (3) standardization and then retrofitting of the sense vectors with neighbor expansion (*once*); and (4) in each iteration, standardization and then retrofitting of the sense vectors with neighbor expansion (*each time*).

Though GenSense can retrofit sense vectors connected within a given ontology, words outside of the ontology are not learned. To address this issue, we introduce a bilingual mapping method (Mikolov, Le, and Sutskever 2013b; Xing *et al*. 2015; Artetxe, Labaka, and Agirre 2016; Smith *et al*. 2017; Joulin *et al*. 2018) for learning the mapping between the original word embedding and the learned sense embedding that utilize the Procrustes analysis. The goal of orthogonal Procrustes analysis is to find a transformation matrix W such that the representations before the

[a]Besides Word2Vec and GloVe, embeddings trained by others on different corpora and different parameters may also be used.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

sense retrofitting are close to the representations after the sense retrofitting. After obtaining W, we can apply the transformation matrix to the senses that are not retrofitted.

In the experiments, we show that the GenSense model outperforms previous approaches on four types of datasets: semantic relatedness, contextual word similarity, semantic difference, and synonym selection. With an experiment to evaluate the benefits yielded by the relation strength, we find a 87.7% performance difference between the worst and the best cases in WordSim-353 semantic relatedness benchmark dataset (Finkelstein *et al.* 2002). While the generalized model which considers all the relations performs well in the semantic relatedness tasks, we also find that antonym relations perform particularly well in the semantic difference experiment. We also find that the proposed standardization, neighbor expansion, and the combination of these two processes improve performance on the semantic relatedness experiment.

The remainder of this paper is organized as follows. Section 2 introduces some related works. Section 3 describes our proposed generalized sense retrofitting model. In Section 4, we show the Procrustes analysis on GenSense. Section 5 describes the details of the experiments. The results discussions are shown in Section 6. In Section 7, we show the limitation of the model and point out some future directions. Finally, we conclude this research in Section 8.

2. Related work

The study of word representations has a long history. Early approaches include utilizing the term-document occurrence matrix from a large corpus and then using dimension reduction techniques such as singular value decomposition (Deerwester *et al.* 1990; Bullinaria and Levy 2007). Beyond that, recent unsupervised word embedding approaches (sometimes referred to as corpus-based approaches) based on neural networks (Mikolov *et al.* 2013a; Pennington *et al.* 2014; Dragoni and Petrucci 2017) have performed well on syntactic and semantic tasks. Among these, Word2Vec word embeddings were released using the continuous bag-of-words (CBOW) model and the skip-gram model (Mikolov *et al.* 2013a). The CBOW model predicts the center word using contextual words, while the skip-gram model predicts contextual words using the center word. GloVe, another widely adopted word embedding model, is a log-bilinear regression model that mitigates the drawbacks of global factorization approaches (such as latent semantic analysis; Deerwester *et al.* 1990) and local context window approaches (such as the skip-gram model) on the word analogy and semantic relatedness tasks (Pennington *et al.* 2014). The global vectors in GloVe for word embedding are trained using unsupervised learning on aggregated global word–word co-occurrence statistics from a corpus; this encodes the relationship between words and yields vectorized representations for words that satisfy ratios between words. The objective functions of Word2Vec and GloVe are slightly different. Word2Vec utilizes negative sampling to make words that do not frequently co-occur more dissimilar, whereas GloVe instead uses a weighting function to adjust the word–word co-occurrence counts; Word2Vec does not use this method. To deal with the out-of-vocabulary (oov) issue in word representations, FastText (Bojanowski *et al.* 2017) is a more advanced model which leverages subword information. For example, when considering the word *asset* with tri-gram, it will be represented by the following character tri-gram: *<as, ass, sse, set, et>*. This technique not only resolved the oov issue but also better represented low frequency words.

Apart from unsupervised word embedding learning models, there exist ontologies that contain lexical knowledge such as WordNet (Miller 1998), Roget's 21st Century Thesaurus (Kipfer 1993), and PPDB (Pavlick *et al.* 2015). Although these ontologies are useful in some applications, different ontologies contain different structure. In Roget, a synonym set contains all the words of the same sense and has its unique definition. For example, the word *free* in Roget has at least two adjective senses: (*free, complimentary, comp, unrecompensed*) and (*free, available, clear*). The definition of the first sense (*without charge*) is different from the second sense (*not busy; unoccupied*).

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

The synonym's relevance can be different in each set: the ranking in the first sense of *free* is (*free, complimentary*) > (*free, comp*) > (*free, unrecompensed*). PPDB is an automatically created massive resource of paraphrases of three types: lexical, phrasal, and syntactic (Ganitkevitch *et al*. 2013; Pavlick *et al*. 2015). For each type, there are several sizes with different trade-offs in terms of precision and recall. Each pair of words is semantically equivalent in some degree. For example, (*automobile, car, auto, wagon, . . .*) is listed in the coarsest size, while the finest size contains only (*automobile, car, auto*).

As the development of these lexical ontologies and word embedding models has matured, many have attempted combining them either with joint training (Bian, Gao, and Liu 2014; Yu and Dredze 2014; Bollegala *et al*. 2016; Liu, Nie, and Sordoni 2016; Mancini *et al*. 2017) or post-processing (Faruqui *et al*. 2015; Ettinger, Resnik, and Carpuat 2016; Mrkšic *et al*. 2016; Lee *et al*. 2017; Lengerich *et al*. 2017; Glavaš and Vulić 2018). When the need for sense embedding becomes more obvious, some researches focus on learning sense-level embedding with lexical ontology.

Joint training for sense embedding utilizes information contained in the lexical database during the intermediate word embedding generation steps. For example, as the SensEmbed model utilizes Babelfy to annotate the Wikipedia corpus, it generates sense-level representations (Iacobacci *et al*. 2015). NASARI uses WordNet and Wikipedia to generate word-based and synset-based representations and then linearly combines the two embeddings (Camacho-Collados, Pilehvar, and Navigli 2015). Mancini *et al*. (2017) proposed to learn word and sense embeddings in the same space via a joint neural architecture.

In contrast, this research focuses on post-processing approach. For retrofitting on word embedding, a new word embedding model is learned by retrofit (refine) the pre-trained word embedding with the lexical database's information. One of the advantages of post-processing is the non-necessity of training a word embedding from scratch, which often takes huge amounts of time and computation power. Faruqui *et al*. (2015) proposed an objective function for retrofitting which minimizes the Euclidean distance of synonymic or hypernym–hyponym relation words in WordNet, while at the same time it preserves the original word embedding's structure. Similar to retrofitting, a counter-fitting model is proposed to not only minimize the distance between vectors of words with synonym relations but also maximize the distance between vector of words with antonym relations (Mrkšic *et al*. 2016). Their qualitative analysis shows that before counter-fitting, words are related but not similar. After counter-fitting, the closest words are similar words.

For post-processing sense models, the Retro model (Jauhar *et al*. 2015) applies graph smoothing with WordNet as a retrofitting step to tease the vectors of different senses apart. Li and Jurafsky (2015) proposed to learn sense embedding through Chinese restaurant processes and show a pipelined architecture for incorporating sense embeddings into language understanding. Ettinger *et al*. (2016) proposed to use parallel corpus to build sense graph and then perform retrofitting on the constructed sense graph. Yen *et al*. (2018) proposed to learn sense embedding through retrofitting on sense and contextual neighbors jointly; however, the negative relations were not considered in their model. Remus and Biemann (2018) used unsupervised sense inventory to perform retrofitting on word embedding to learn the sense embedding, though the quality of the unsupervised sense inventory is questionable. Although it has been shown that sense embedding does not improve every natural language processing task (Li and Jurafsky 2015), there is still a great need for sense embedding for tasks that need sense-level representation (i.e., synonym selection, word similarity rating, and word sense induction) (Azzini *et al*. 2011; Ettinger *et al*. 2016; Qiu, Tu, and Yu 2016). A survey on word and sense embeddings can be found in Camacho-Collados and Pilehvar (2018). After the proposal of the transformer model, researchers either utilize the transformer model directly (Wiedemann *et al*. 2019) or trained with ontologies to better capture the sense of a word in specific sentences (Shi *et al*. 2019; Loureiro and Jorge 2019); Scarlini, Pasini, and Navigli 2020).

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

3. Generalized sense retrofitting model

3.1. The GenSense model

The GenSense model is to learn a better sense representation such that each new representation is close to its word form representation, its synonym neighbors, and its positive contextual neighbors, while actively pushing away from its antonym neighbors and its negative contextual neighbors (Lee *et al.* 2018). Let $V = \{w_1, ..., w_n\}$ be a vocabulary of a trained word embedding and $|V|$ be its size. The matrix $\hat{Q}$ is 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.* 2013a). Let $\Omega = (T, E)$ be an ontology that contains the semantic relationship, where $T = \{t_1, ..., t_m\}$ is a set of senses and $|T|$ the total number of senses. In general, $|T| > |V|$ since one word may contain more than one sense. For example, in WordNet the word *gay* has at least two senses *gay.n.*01 (the first noun sense; homosexual, homophile, homo, gay) and *gay.a.*01 (the first adjective sense; cheery, gay, sunny). 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. For example, the set of synonyms and antonyms as there is no case of a semantic relationship of a pair of senses belong to synonym and antonym at the same time, and thus $E = E_{r_1} \cup E_{r_2} \cup ... \cup E_{r_k}$. If r_1 denotes the synonym relationship, then $(i, j) \in E_{r_1}$ 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.[b] Then the goal is to learn a new matrix $S = (s_1, ..., s_m)$ such that each new sense vector is close to its word form vertex and its synonym neighbors. The basic form that considers only synonym relations for the objective of the sense retrofitting model is

$$\sum_{i=1}^{m} \left[\alpha_1 \beta_{ii} \|s_i - \hat{q}_{t_i}\|^2 + \alpha_2 \sum_{(i,k) \in E_{r_1}} \beta_{ij} \|s_i - s_k\|^2 \right] \tag{1}$$

where the α's balance the importance of the word form vertex and the synonym, and the β's control the strength of the semantic relations. When $\alpha_1 = 0$ and $\alpha_2 = 1$, the model only considers the synonym neighbors and may be too deviate from the original vector. From Equation (1), a learned sense vector approaches its synonyms, meanwhile constraining its distance with its original word form vector. In addition, this equation can be further generalized to consider all relations as

$$\sum_{i=1}^{m} \left[\alpha_1 \beta_{ii} \|s_i - \hat{q}_{t_i}\|^2 + \alpha_2 \sum_{(i,k) \in E_{r_1}} \beta_{ij} \|s_i - s_k\|^2 + \dots \right]. \tag{2}$$

Apart from the positive sense relation, we now introduce three types of special relations. The first is the positive contextual neighbor relation r_2. $(i, j) \in E_{r_2}$ if and only if t_j is the synonym of t_i and the surface form of t_j has only one sense. In the model, we use the word form vector to represent the neighbors of the t_i's in E_{r_2}. These neighbors are viewed as positive contextual neighbors, as they are learned from the context of a corpus (e.g., Word2Vec trained on the Google News corpus) with positive meaning. The second is the negative sense relation r_3. $(i, j) \in E_{r_3}$ if and only if t_j is the antonym of t_i. The negative senses are used in a subtractive fashion to push the sense away from the positive meaning. The last is the negative contextual neighbor relation r_4. $(i, j) \in E_{r_4}$ if and only if t_j is the antonym of t_i and the surface form of t_j has only one sense. As with the positive contextual neighbors, negative contextual neighbors are learned from the context of a corpus, but with negative meaning. Table 1 summarizes the aforementioned relations.

In Figure 1, which contains an example of the relation network, the word gay may have two meanings: (1) *bright and pleasant; promoting a feeling of cheer* and (2) *someone who is sexually*

[b] Note that $\hat{q}_{t_j}$ and $\hat{q}_{t_k}$ may be mapped to the same vector representation even if $j \neq k$. For example, let t_j be *gay.n.01* and t_k be *gay.a.01*. Then both $\hat{q}_{t_j}$ and $\hat{q}_{t_k}$ are mapped to the word form vector of *gay*.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 1. Summary of the semantic relations. $sf(t_i)$ is the word surface form of sense t_i

	Relation	Definition
r_1	Synonym	$(i,j) \in E_{r_1}$ iff t_j is the synonym of t_i
r_2	Positive contextual neighbor	$(i,j) \in E_{r_2}$ iff t_j is the synonym of t_i and $sf(t_j)$ has only one sense
r_3	Antonym	$(i,j) \in E_{r_3}$ iff t_j is the antonym of t_i
r_4	Negative contextual neighbor	$(i,j) \in E_{r_2}$ iff t_j is the antonym of t_i and $sf(t_j)$ has only one sense

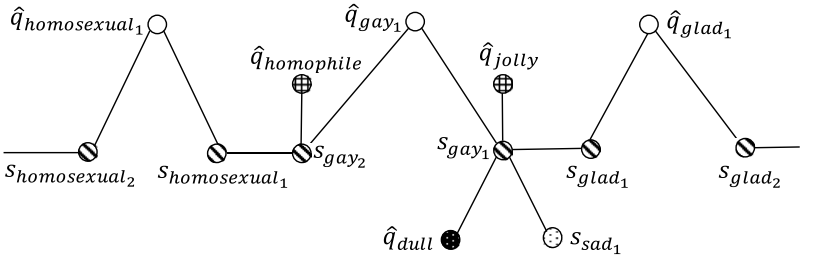

Figure 1. An illustration of the relation network. Different node textures represent different roles (e.g., synonym and antonym) in the GenSense model.

attracted to persons of the same sex. If we focus on the first sense, then our model attracts s_{gay_1} to its word form vector $\hat{q}_{gay_1}$, its synonym s_{glad_1}, and its positive contextual neighbor $\hat{q}_{jolly}$. At the same time, it pushes s_{gay_1} from its antonym s_{sad_1} and its negative contextual neighbor $\hat{q}_{dull}$.

Formalizing the above scenario and considering all its parts, Equation (2) becomes

$$\sum_{i=1}^{m}\Bigg[\alpha_1\beta_{ii}\|s_i-\hat{q}_{t_i}\|^2+\alpha_2\sum_{(i,j)\in E_{r_1}}\beta_{ij}\|s_i-s_j\|^2+\alpha_3\sum_{(i,j)\in E_{r_2}}\beta_{ij}\|s_i-\hat{q}_j\|^2 + \alpha_4\sum_{(i,j)\in E_{r_3}}\beta_{ij}\|s_i+s_j\|^2+\alpha_5\sum_{(i,j)\in E_{r_4}}\beta_{ij}\|s_i+\hat{q}_j\|^2\Bigg]. \tag{3}$$

We therefore apply an iterative updating method to the solution of the above convex objective function (Bengio, Delalleau, and Le Roux 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 is

$$s_i=\frac{\left[\begin{array}{l}-\alpha_5\sum\limits_{j:(i,j)\in E_{r_4}}\beta_{ij}\hat{q}_j-\alpha_4\sum\limits_{j:(i,j)\in E_{r_3}}\beta_{ij}s_j \\ +\alpha_3\sum\limits_{j:(i,j)\in E_{r_2}}\beta_{ij}\hat{q}_j+\alpha_2\sum\limits_{j:(i,j)\in E_{r_1}}\beta_{ij}s_j+\alpha_1\beta_{ii}\hat{q}_{t_i}\end{array}\right]}{\left[\begin{array}{l}\alpha_5\sum\limits_{j:(i,j)\in E_{r_4}}\beta_{ij}+\alpha_4\sum\limits_{j:(i,j)\in E_{r_3}}\beta_{ij} \\ +\alpha_3\sum\limits_{j:(i,j)\in E_{r_2}}\beta_{ij}+\alpha_2\sum\limits_{j:(i,j)\in E_{r_1}}\beta_{ij}+\alpha_1\beta_{ii}\end{array}\right]} \tag{4}$$

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

A formal description of the GenSense method is shown in Algorithm 1 (Lee *et al.* 2018), in which the β parameters are retrieved from the ontology, and ε is a threshold for deciding whether to update the sense vector or not, and as such is used as a stopping criterion when the difference between the new sense vector and the original sense vector is small. Empirically, 10 iterations are sufficient to minimize the objective function from a set of starting vectors to produce effective sense-retrofitted vectors. Based on the GenSense model, the next three subsections will describe three approaches to further improve the sense representations.

Algorithm 1. GenSense

Input: Pre-trained word embedding $\hat{Q}$, relation ontology $\Omega = (T, E)$, hyper-parameters α, parameters β, number of iterations *max_it*, convergence criterion for sense vectors ε
Output: Trained sense embedding S

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

3.2. Standardization on dimensions

Although the original GenSense model considers the semantic relations between the senses, the relation strength, and the semantic strength, the literature indicates that the vanilla word embedding model benefits from standardization on the dimensions (Lee *et al.* 2016). In this approach, let $1 \leq j \leq d$ be the d dimensions in the sense embedding. Then for every sense vector $s_i \in \mathbb{R}^d$, the z-score is computed on each dimension as

$$s'_{ij} = \frac{s_{ij} - \mu}{\sigma}, \forall i, j \tag{5}$$

where μ is the mean and σ is the standard deviation of dimension j. After this process, the sense vector is then divided by its norm to ensure a summation of 1:

$$s''_i = \frac{s'_i}{\left\| s'_i \right\|}, \forall i \tag{6}$$

where $\|s_i\|$ is the norm of the sense vector s_i. As this standardization process can be placed in multiple places, we consider the following four situations:

(1) GenSense-Z: the standardization process is performed after every iteration.
(2) GenSense-Z-last: the standardization process is performed only at the end of the whole algorithm.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

(3) Z-GenSense: the standardization process is performed at the beginning of each iteration.

(4) Z-first-GenSense: the standardization process is performed only once, before iteration.

The details of this approach are shown in Algorithms 2 and 3. Note that although further combinations or adjustments of these situations are possible, in the experiments we analyze only these four situations.

Algorithm 2. Standardization Process

Input: Sense embedding S
Output: Standardized sense embedding S
Function `standardization`(S)
  for $i = 1$ **to** m **do**
    for $j = 1$ **to** d **do**
      $s_{ij}^{tmp} \leftarrow \frac{s_{ij}-\mu}{\sigma}$
    end
  end
  for $i = 1$ **to** m **do**
    $s_i \leftarrow \frac{s_i^{tmp}}{\|s_i^{tmp}\|}$
  end
  return S
end

3.3. Neighbor expansion from nearest neighbors

In this approach, we utilize the nearest neighbors of the target sense vector to refine GenSense. Intuitively, if the sense vector s_i's nearest neighbors are uniformly distributed around s_i, then they may not be helpful. In contrast, if the neighbors are clustered and gathered in a distinct direction, then utilization of these neighbors is crucial. Figure 2 contains examples of nearest neighbors that may or may not be helpful. In Figure 2(a), *cheap*′s neighbors are not helpful since they are uniformly distributed and thus make the effect of the neighbors canceled. In Figure 2(b), *love*′s neighbors are helpful since they are gathered in the same quadrant and make the new sense vector of *love* closer to its related senses.

In practice, we pre-build the sense embedding k-d tree for rapid lookups of the nearest neighbors of vectors (Maneewongvatana and Mount 1999). After building the k-d tree and take into consideration of the nearest neighbor term, the update formula for s_i becomes

$$s_i = \frac{\begin{bmatrix} \alpha_6 \sum\limits_{j:(i,j)\in NN(s_i)} \beta_{ij} s_j & -\alpha_5 \sum\limits_{j:(i,j)\in E_{r_4}} \beta_{ij}\hat{q}_j - \alpha_4 \sum\limits_{j:(i,j)\in E_{r_3}} \beta_{ij} s_j \\ +\alpha_3 \sum\limits_{j:(i,j)\in E_{r_2}} \beta_{ij}\hat{q}_j & +\alpha_2 \sum\limits_{j:(i,j)\in E_{r_1}} \beta_{ij} s_j + \alpha_1 \beta_{ii}\hat{q}_{t_i} \end{bmatrix}}{\begin{bmatrix} \alpha_6 \sum\limits_{j:(i,j)\in NN(s_i)} \beta_{ij} & +\alpha_5 \sum\limits_{j:(i,j)\in E_{r_4}} \beta_{ij} + \alpha_4 \sum\limits_{j:(i,j)\in E_{r_3}} \beta_{ij} \\ +\alpha_3 \sum\limits_{j:(i,j)\in E_{r_2}} \beta_{ij} & +\alpha_2 \sum\limits_{j:(i,j)\in E_{r_1}} \beta_{ij} + \alpha_1 \beta_{ii} \end{bmatrix}} \tag{7}$$

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Algorithm 3. GenSense with Standardization Process

Input: Pre-trained word embedding $\hat{Q}$, relation ontology $\Omega = (T, E)$, hyper-parameters α, parameters β, number of iterations *max_it*, convergence criterion for sense vectors ε, condition *cond*

Output: Trained sense embedding S

1 **for** $i = 1$ **to** m **do**
2 $\quad s_i^{(0)} \leftarrow \hat{q}_{t_i}$
3 **end**
4 **if** *cond* = *Z-first-GenSense* **then**
5 $\quad S \leftarrow$`standardization(`S`)`
6 **end**
7 **for** $it = 1$ **to** $max\_it - 1$ **do**
8 $\quad$ **if** *cond* = *Z-GenSense* **then**
9 $\quad\quad S \leftarrow$`standardization(`S`)`
10 $\quad$ **end**
11 $\quad$ **for** $i = 1$ **to** m **do**
12 $\quad\quad$ Compute s_i^{tmp} using Equation (4).
13 $\quad\quad$ **if** $\left\| s_i^{tmp} - s_i^{(it-1)} \right\| \geq \varepsilon$ **then**
14 $\quad\quad\quad s_i^{(it)} \leftarrow s_i^{tmp}$
15 $\quad\quad$ **else**
16 $\quad\quad\quad s_i^{(it)} \leftarrow s_i^{(it-1)}$
17 $\quad\quad$ **end**
18 $\quad$ **end**
19 $\quad$ **if** *cond* = *GenSense-Z* **then**
20 $\quad\quad S \leftarrow$`standardization(`S`)`
21 $\quad$ **end**
22 **end**
23 **if** *cond* = *GenSense-Z-last* **then**
24 $\quad S \leftarrow$`standardization(`S`)`
25 **end**
26 **return** S

where $NN(s_i)$ is the set of N nearest neighbors of s_i and α_6 is a newly added parameter for weighting the importance of the nearest neighbors. Details of the proposed neighbor expansion approach are shown in Algorithm 4. The main procedure of Algorithm 4 is similar to that of Algorithm 1 (GenSense) with two differences: (1) in line 4 we need to build the k-d tree and (2) in line 7 we need to compute the nearest neighbors for Equation (7).

3.4. Combination of standardization and neighbor expansion

With the standardization and neighbor expansion approaches, a straightforward and natural way to further improve the sense embedding's quality is to combine these two approaches. In this study, we propose four combination situations:

(1) GenSense-NN-Z: in each iteration, GenSense is performed with neighbor expansion, after which the sense embedding is standardized.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

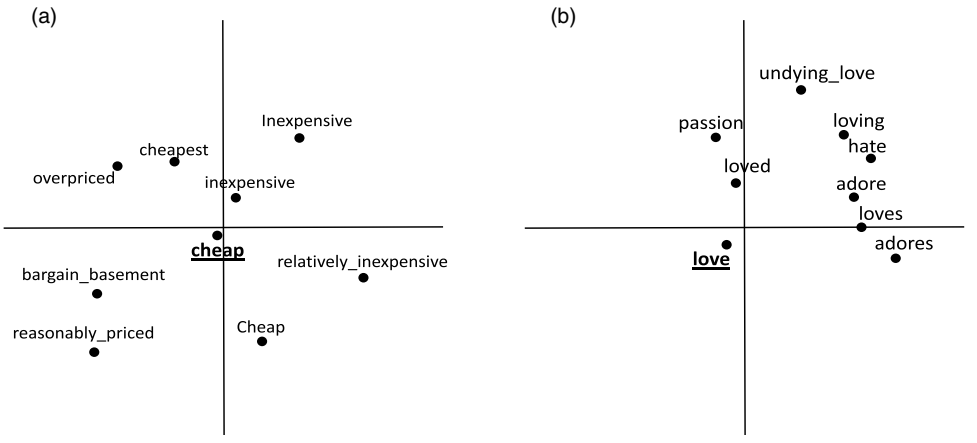

Figure 2. Nearest neighbors of *cheap* (a) and *love* (b). In (a), *cheap*'s neighbors are uniformly distributed and thus not helpful. In (b), *love*'s neighbors are gathered in the quadrant I, and thus can be attracted toward them.

Algorithm 4. GenSense-NN

Input: Pre-trained word embedding $\hat{Q}$, relation ontology $\Omega = (T, E)$, hyper-parameters α, parameters β, number of iterations *max_it*, convergence criterion for sense vectors ε, number of nearest neighbors N

Output: Trained sense embedding S

for $i = 1$ **to** m **do**
  $s_i^{(0)} \leftarrow \hat{q}_{t_i}$
end
Build k-d tree according to $\hat{Q}$
for $it = 1$ **to** $max\_it - 1$ **do**
  for $i = 1$ **to** m **do**
    $NN(s_i) \leftarrow \{s | s \in N$ nearest neighbors of $s_i\}$
    Compute s_i^{tmp} using Equation (7).
    if $\left\| s_i^{tmp} - s_i^{(it-1)} \right\| \geq \varepsilon$ **then**
      $s_i^{(it)} \leftarrow s_i^{tmp}$
    else
      $s_i^{(it)} \leftarrow s_i^{(it-1)}$
    end
  end
end
return S

(2) GenSense-NN-Z-last: in each iteration, GenSense is performed with neighbor expansion. The standardization process is performed only after the last iteration.

(3) GenSense-Z-NN: in each iteration, the sense embedding is standardized and GenSense is performed with neighbor expansion.

(4) GenSense-Z-NN-first: standardization is performed only once, before the iteration process. After that, GenSense is performed with neighbor expansion.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

As with standardization on the dimensions, although different combinations of the standardization and neighbor expansion approaches are possible, we analyze only these four situations in our experiments.

4. Procrustes analysis on GenSense

Although the GenSense model retrofits sense vectors connected within a given ontology, words outside of the ontology are not learned. We address this by introducing a bilingual mapping method (Mikolov *et al.* 2013b; Xing *et al.* 2015; Artetxe *et al.* 2016; Smith *et al.* 2017; Joulin *et al.* 2018) for learning the mapping between the original word embedding and the learned sense embedding. Let $\{x_i, y_i\}_{i=1}^{n}$ be the pairs of corresponding representations before and after sense retrofitting, where $x_i \in \mathbb{R}^d$ is the representation before sense retrofitting and $y_i \in \mathbb{R}^d$ is that after sense retrofitting.[c] The goal of orthogonal Procrustes analysis is to find a transformation matrix W such that Wx_i approximates y_i:

$$\min_{W \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i=1}^{n} l\left(Wx_i, y_i\right), \text{ subject to } W^T W = I_d \tag{8}$$

where we select the typical square loss $l_2(x, y) = \|x - y\|_2^2$ as the loss function. When constraining W to be orthogonal (i.e., $W^T W = I_d$, where I_d is an identity matrix with d-dimensionality and the dimensionality of the representation before and after retrofitting is the same), selecting the square loss function makes formula 8 a least-squares problem, solvable with a closed-form solution. From formula 8,

$$\begin{aligned}
&\arg\min_{W \in \mathbb{R}^{d \times d}} \frac{1}{n} \sum_{i=1}^{n} l\left(Wx_i, y_i\right) \\
&= \arg\min_{W \in \mathbb{R}^{d \times d}} \sum_{i=1}^{n} \left\| Wx_i - y_i \right\|_2^2 \\
&= \arg\min_{W \in \mathbb{R}^{d \times d}} \sum_{i=1}^{n} \left\| Wx_i \right\|_2^2 - 2y_i^T Wx_i + \left\| y_i \right\|_2^2 \\
&= \arg\min_{W \in \mathbb{R}^{d \times d}} \sum_{i=1}^{n} y_i^T Wx_i.
\end{aligned} \tag{9}$$

Let $X = (x_1, ..., x_n)$ and $Y = \left(y_1, ..., y_n\right)$. Equation (9) can be expressed as

$$\arg\max_{W \in \mathbb{R}^{d \times d}} \mathrm{Tr}\left(Y^T WX\right) \tag{10}$$

where $\mathrm{Tr}(\cdot)$ is the trace operator $\mathrm{Tr}(A) = \sum_{i=1}^{n} a_{ii}$. We first rearrange the matrices in the trace operator

$$\mathrm{Tr}\left(Y^T WX\right) = \mathrm{Tr}\left(WXY^T\right); \tag{11}$$

then, using the singular value decomposition $XY^T = U\Sigma V^T$, formula 11 becomes

$$\mathrm{Tr}\left(WU\Sigma V^T\right) = \mathrm{Tr}\left(V^T WU\Sigma\right). \tag{12}$$

[c]Theoretically, the dimensionality of x_i and y_i can be different. However, in this study we only consider the special case the dimensionality of x_i and y_i be the same.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Since V^T, W, and U are orthogonal matrices, $P = V^T WU$ must be an orthogonal matrix. From Equation (12), it follows that

$$\mathrm{Tr}\left(V^T WU\Sigma\right) = \mathrm{Tr}(P\Sigma) = \sum_{i=1}^{n} p_{ii}\sigma_i \leq \sum_{i=1}^{n} \left|p_{ii}\right| \sigma_i \leq \sum_{i=1}^{n} \sigma_i. \tag{13}$$

From Equation (13), $\left|p_{ii}\right| \leq 1$; given the orthogonality of P, $P = I_n$. As a result, $V^T WU = I$ and $W = VU^T$.

4.1. Inference from Procrustes method

After obtaining the transformation matrix W, the next step is to infer the sense representations that cannot be retrofitted from the GenSense model (out-of-ontology senses). For a bilingual word embedding, there are two methods for representing these mappings. The first is finding a corresponding mapping of the word that is to be translated:

$$t(i) \in \underset{j\in\{1,..,n,n+1,...,N\}}{\arg\min} \left\| Wx_i - y_i \right\|_2^2 \tag{14}$$

for word i, where $t(i)$ denotes the translation and $\{1, ..., n, n+1, ..., N\}$ denotes the vocabulary of the target language. Though this process is commonly used in bilingual translation, there is a simpler approach for retrofitting. The second method is simply to apply the transformation matrix on the word that is to be translated. However, in bilingual embedding this method yields only the mapped vector and not the translated word in the target language. One advantage with GenSense is that all corresponding senses (those before and after applying GenSense) are known. As a result, we simply apply the transformation matrix to the senses that are not retrofitted (i.e., Wx_i for sense i). In the experiments, we show only the results of the second method, as it is more natural within the context of GenSense.

5. Experiments

We evaluated GenSense using four types of experiments: semantic relatedness, contextual word similarity, semantic difference, and synonym selection. In the testing phase, if a test dataset had missing words, we used the average of all sense vectors to represent the missing word. Note that the results we report for vanilla sense embedding may be slightly different from other work due to the handling of missing words and the similarity computation method. Some work uses a zero vector to represent missing words, whereas some removes missing words from the dataset. Thus, the reported performance should be compared given the same missing word processing method and the same similarity computation method.

5.1. Ontology construction

Roget's 21st Century Thesaurus (Kipfer 1993) was used to build the ontology in the experiments as it includes strength information for the senses. As Roget does not directly provide an ontology, we used it to manually construct synonym and antonym ontologies. We first fetched all the words with their sense sets. Let Roget's word vocabulary be $V = \{w_1, ..., w_n\}$ and the initial sense vocabulary be $\left\{w_{11}, w_{12}, ..., w_{1m_1}, ..., w_{n1}, w_{n2}, ..., w_{nm_n}\right\}$, where word w_i has m_i senses, including all parts of speeches (POSes) of w_i. For example, *love* has four senses: (1) noun, adoration; very strong liking; (2) noun, person who is loved by another; (3) verb, adore, like very much; and (4) verb, have sexual relations. The initial sense ontology would be $O = \left\{W_{11}, W_{12}, ..., W_{1m_1}, ..., W_{n1}, W_{n2}, ..., W_{nm_n}\right\}$. In the ontology, each word w_i had m_i senses,

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

of which w_{i1} was the default sense. The default sense is the first sense in Roget and is usually the most common sense when people use this word. Each W_{ij} carried an initial word set $\{w_k | w_k \in V, w_k \text{ is the synonym of } w_{ij}\}$. For example, *bank* has two senses: $bank_1$ and $bank_2$. The initial word set of $bank_1$ is *store* and *treasury* (which refers to *financial institution*). The initial word set of $bank_2$ is *beach* and *coast* (which refers to *ground bounding waters*). Then we attempted to assign a corresponding sense to the words in the initial word set. For each word w_k in the word set of W_{ij}, we first computed all the intersections of W_{ij} with $W_{k1}, W_{k2}, ..., W_{km_k}$, after which we selected the largest intersection according to the cardinalities. If all the cardinalities were zero, then we assigned the default sense to the target word. The procedure for the construction of the ontology for Roget's thesaurus is given in Algorithm 5. The building of the antonym ontology from Roget's thesaurus was similar to Algorithm 5; it differed in that the initial word set was set to $\{w_k | w_k \in V, w_k \text{ is the antonym of } w_{ij}\}$.

Algorithm 5. Construction of ontology given Roget's senses

Input: Pre-fetched sense ontology O
Output: Constructed sense ontology O

foreach $W_{ij} \in O$ **do**
  foreach $w_k \in W_{ij}$ **do**
    $cardinalities \leftarrow \emptyset$
    for $l = 1$ **to** m_k **do**
      $cardinalities\,[l] \leftarrow \text{cardinality}\,(W_{kl} \cap W_{ij})$
    end
    $n \leftarrow \text{find\_largest\_index}\,(cardinalities)$
    if $cardinalities\,[n] = 0$ **then**
      $w_k \leftarrow w_{k1}$
    else
      $w_k \leftarrow w_{kn}$
    end
  end
end
return O

The vocabulary from the pre-trained GloVe word embedding was used to fetch and build the ontology from Roget. In Roget, the synonym relations were provided in three relevance levels; we set the β's to 1.0, 0.6, and 0.3 for the most relevant synonyms to the least. The antonym relations were constructed in the same way. For each sense, β_{ii} was set to the sum of all the relation's specific weights. Unless specified otherwise, in the experiments we set the α's to 1. Although in this study we show only the Roget results, other ontologies (e.g., PPDB Ganitkevitch *et al*. 2013; Pavlick *et al*. 2015 and WordNet Miller 1998) could be incorporated into GenSense as well.

5.2. Semantic relatedness

Measuring semantic relatedness is a common way to evaluate the quality of the proposed embedding models. We downloaded four semantic relatedness benchmark datasets from the web.

5.2.1. MEN

MEN (Bruni, Tran, and Baroni 2014)) contains 3000 word pairs crowdsourced from Amazon Mechanical Turk. Each word pair has a similarity score ranging from 0 to 50. In their

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 2. Word frequency distributions for WS353 and RW

Frequency	WS353	RW
0 (unknown)	0	801
1–100	0	41
101–1000	9	676
1001–10,000	87	719
>10,000	341	714

crowdsourcing procedure, the annotators were asked to select the more related word pair from two candidate word pairs. For example, between the candidates $(wheels, car)$ and $(dog, race)$, the annotators were to select $(wheels, car)$ since every car has wheels, but not every dog is involved in a race. We further labeled the POS in MEN: 81% were nouns, 7% were verbs, and 12% were adjectives. In the MEN dataset, there are two versions of the word pairs: lemma and natural form. We show the natural form in the experimental results, but the performance on the two datasets is quite similar.

5.2.2. MTurk

MTurk (Radinsky *et al*. 2011) contains 287 human-labeled examples of word semantic relatedness. Each word pair has a similarity score ranging from 1 to 5 from 10 subjects. A higher score value indicates higher similarity. In MTurk, we labeled the POS: 61% were nouns, 29% were verbs, and 10% were adjectives.

5.2.3. WordSim353 (WS353)

WordSim-353 (Finkelstein *et al*. 2002) contains 353 noun word pairs. Each pair has a human-rated similarity score ranging from 0 to 10. 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.

5.2.4. Rare words

Rare words (RWs) (Luong, Socher, and Manning 2013) contain 2034 word pairs crowdsourced from Amazon Mechanical Turk. Each word pair has a similarity score ranging from 0 to 10. A higher score value indicates higher similarity. In RW, frequencies of some words are very low. Table 2 shows the word frequency statistics of WS353 and RW based on Wikipedia.

In RW, the number of unknown words is 801; 41 words other appear no more than 100 times in Wikipedia. In WS353, in contrast, all words appear more than 100 times in Wikipedia. As some of the words are challenging even for native English speakers, the annotators were asked if they knew the first and second words. Word pairs unknown to most raters were discarded. We labeled the POS: 47% were nouns, 32% were verbs, 19% were adjectives, and 2% were adverbs.

To measure the semantic relatedness between a word pair (w, w') in the datasets, we adopted the sense evaluation metrics AvgSim and MaxSim (Reisinger and Mooney 2010)

$$AvgSim(w, w') \stackrel{\text{def}}{=\!=} \frac{1}{K_w K_{w'}} \sum_{j=1}^{K_w} \sum_{k=1}^{K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right) \tag{15}$$

$$MaxSim(w, w') \stackrel{\text{def}}{=\!=} \max_{1 \le j \le K_w, 1 \le k \le K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right) \tag{16}$$

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 3. Semantic relatedness benchmark datasets

	MEN	MTurk	RW	WS353
Pair count	3000	287	2034	353
Word count	751	499	2951	437
Roget	707	416	2371	412
Retrofitted word count	685	400	1902	394
Retrofitted word ratio (%)	91.21	80.16	64.45	90.16

where K_w and K'_w denote the number of senses of w and w', respectively. AvgSim can be seen as a *soft* metric as it takes the average of the similarity scores, whereas the MaxSim can be seen as a *hard* metric as it selects only those senses with the maximum similarity score. To measure the performance of the sense embeddings, we computed the Spearman correlation between the human-rated scores and the AvgSim/MaxSim scores. Table 3 shows a summary of the benchmark datasets and their relationships with the ontologies. Row 3 shows the number of words that were listed both in the datasets and the ontology. As some words in Roget were not retrofitted, rows 4 and 5 show the number and ratio of words that were affected by the retrofitting model. The word count for Roget was 63,942.

5.3. Contextual word similarity

Although the semantic relatedness datasets have been used often, one disadvantage is that the words in these word pairs lack contexts. Therefore, we also conducted experiments with Stanford's contextual word similarities (SCWS) dataset (Huang *et al.* 2012), which consists of 2003 word pairs (1713 words in total, as some words are shown in multiple questions) together with human-rated scores. A higher score value indicates higher semantic relatedness. In contrast to the semantic relatedness datasets, SCWS words have contexts and POS tags, that is, the human subjects knew the usage of the word when they rated the similarity. For each word pair, we computed its AvgSimC/MaxSimC scores from the learned sense embedding (Reisinger and Mooney 2010)

$$AvgSimC(w, w') \stackrel{\text{def}}{=\!=} \frac{1}{K^2} \sum_{j=1}^{K} \sum_{k=1}^{K} d_{c,w,k} d_{c',w',j} d\left(\pi_k(w), \pi_j(w')\right) \tag{17}$$

$$MaxSimC(w, w') \stackrel{\text{def}}{=\!=} d\left(\hat{\pi}(w), \hat{\pi}(w')\right) \tag{18}$$

where $d_{c,w,k} \stackrel{\text{def}}{=\!=} d(v(c), \pi_k(w))$ 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 used a window size of 5 for words in the word pairs (i.e., 5 words prior to w/w' and 5 words after w/w'). Stop words were removed from the context. To measure the performance, we computed the Spearman correlation between the human-rated scores and the AvgSimC/MaxSimC scores.

5.4. Semantic difference

This task was to determine if a given word had a closer semantic feature to a concept than another word (Krebs and Paperno 2016). In this dataset, there were 528 concepts, 24,963 word pairs, and 128,515 items. Each word pair came with a feature. For example, in the test $(airplane, helicopter) : wings$, the first word was to be chosen if and only if $\cos(airplane, wings) >$

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 4. Synonym selection benchmark datasets

	ESL-50	RD-300	TOEFL-80
Question count	50	300	80
Word count	224	1464	395

cos $(helicopter, wings)$; otherwise, the second word was chosen. As this dataset did not provide context for disambiguation, we used strategies similar to the semantic relatedness task:

$$AvgSimD(w, w') \stackrel{\text{def}}{=\!=} \frac{1}{K_w K_{w'}} \sum_{j=1}^{K_w} \sum_{k=1}^{K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right) \tag{19}$$

$$MaxSimD(w, w') \stackrel{\text{def}}{=\!=} \max_{1 \le j \le K_w, 1 \le k \le K_{w'}} \cos\left(v_{w_j}, v_{w'_k}\right). \tag{20}$$

In AvgSimD, we chose the first word iff $AvgSimD(w_1, w') > AvgSimD(w_2, w')$. In MaxSimD, we chose the first word iff $MaxSimD(w_1, w') > MaxSimD(w_2, w')$. The performance was determined by computing the accuracy.

5.5. Synonym selection

Finally, we evaluated the proposed GenSense on three benchmark synonym selection datasets: ESL-50 (acronym for *English as a Second Language*) (Turney 2001), RD-300 (acronym for *Reader's Digest Word Power Game*) (Jarmasz and Szpakowicz 2004), and TOEFL-80 (acronym for *Test of English as a Foreign Language*) (Landauer and Dumais 1997). The numbers in each task represent the numbers of questions in the dataset. In each question, there was a question word and a set of answer words. For each sense embedding, the task was to determine which word in the answer set was most similar to the question word. For example, with *brass* as the question word and *metal, wood, stone,* and *plastic* the answer words, the correct answer was *metal*.[d] As with the semantic relatedness task for the synonym selection task, we used AvgSim and MaxSim. We first used AvgSim/MaxSim to compute the scores for the question word and the words in the answer sets and then selected the answer with the maximum score. Performance was determined by computing the accuracy. Table 4 summarizes the synonym selection benchmark datasets.

5.6. Training models

In the experiments, we use GloVe's 50d version unless otherwise specified as the base model (Pennington *et al.* 2014). The pre-trained GloVe word embedding was trained on Wikipedia and Gigaword-5 (6B tokens, 400k vocab, uncased, 50d vectors). We also test GloVe's 300d version and two well-known vector representation models from the literature: Word2Vec's 300d version (trained on part of Google News dataset which contains 100 billion words) (Mikolov *et al.* 2013a) and FastText's 300d version (2 million word vectors trained on Common Crawl which contains 600B tokens) (Bojanowski *et al.* 2017). Since Word2Vec and FastText did not release a 50d version, we extract the first 50 dimensions from the 300d version to explore the impact of dimensionality. We also conduct experiments on four contextualized word embedding models BERT

[d]Note that there are few cases when answer set contains phrase (e.g., *decoration style*). In such situation, we remove those questions. Another few cases are that the question or answer words are not in the word embedding's vocabulary, in that case, we also remove those questions from the dataset.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

(Devlin *et al.* 2019), DistilBERT (Sanh *et al.* 2019), RoBERTa (Liu *et al.* 2019), and Transformer-XL (T-XL) (Dai *et al.* 2019). To control the experiment, we extract the last four layers for all pre-trained transformer models to represent the word. We tried other layer settings and found that the concatenation of the last four layers can generate good results experimentally. We choose the base uncased version for BERT (3072d) and DistilBERT (3072d), the base version for RoBERTa (3072d), and the transfo-xl-wt103 version for T-XL (4096d).

We set the convergence criterion for the sense vectors to $\varepsilon = 0.1$ and the number of iterations to 10. We used three types of generalization: GenSense-syn (only the synonyms and positive contextual neighbors were considered), GenSense-ant (only the antonyms and negative contextual neighbors were considered), and GenSense-all (everything was considered). Specifically, the objective function of the GenSense-syn was

$$\sum_{i=1}^{m}\left[\alpha_1\beta_{ii}\left\|s_i-\hat{q}_{t_i}\right\|^2+\alpha_2\sum_{(i,j)\in E_{r_1}}\beta_{ij}\left\|s_i-s_j\right\|^2+\alpha_3\sum_{(i,j)E_{r_2}}\beta_{ij}\left\|s_i-\hat{q}_j\right\|^2\right] \tag{21}$$

and the objective function of the GenSense-ant was

$$\sum_{i=1}^{m}\left[\alpha_1\beta_{ii}\left\|s_i-\hat{q}_{t_i}\right\|^2+\alpha_4\sum_{(i,j)\in E_{r_3}}\beta_{ij}\left\|s_i+s_j\right\|^2+\alpha_5\sum_{(i,j)\in E_{r_4}}\beta_{ij}\left\|s_i+\hat{q}_j\right\|^2\right]. \tag{22}$$

6. Results and discussion

6.1. Semantic relatedness

Table 5 shows the Spearman correlation ($\rho \times 100$) of AvgSim and MaxSim between the human scores and the sense embedding scores on each benchmark dataset. For each version (except the transformer ones), the first row shows the performance of the vanilla word embedding. Note that the MaxSim and AvgSim scores are equal when there is only one sense for each word (word embedding). The second row shows the performance of the Retro model (Jauhar *et al.* 2015). The third, fourth, and fifth rows show the GenSense performance for three versions: synonym, antonym, and all, respectively. The last row shows the Procrustes method using GenSense-all. The macro-averaged (average over the four benchmark datasets) and the micro-averaged (weighted average, considering the number of word pairs in every benchmark dataset) results are in the rightmost two columns.

From Table 5, we observe that the proposed model outperforms the Retro and GloVe in all datasets (macro and micro). The best overall model is Procrustes-all. All versions of the GenSense model outperform Retro in almost all the tasks. Retro performs poorly on the RW dataset. In RW, GenSense-syn's MaxSim score exceeds Retro by 22.6 (GloVe 300d), 29.3 (FastText 300d), and 29.1 (Word2Vec 300d). We also observe a significant growth in the Spearman correlation between GenSense-syn and GenSense-all. Surprisingly, the model with only synonyms and positive contextual information outperforms Retro and GloVe. After utilizing the antonym knowledge from Roget, its performance is further improved in all but the RW dataset. This suggests that the antonyms in Roget are quite informative and useful. Moreover, GenSense adapts information from synonyms and antonyms to boost its performance. Although the proposed model pulls sense vectors away from their reverse senses with the help of the antonyms and negative contextual information, this shift does not guarantee that the new sense vectors move to a better place every time with only negative relations. As a result, the GenSense-ant does not perform as well as GenSense-syn in general. Procrustes-all performs better than GenSense-all in most tasks, but the improvement is marginal. This is due to the high ratio of retrofitted words (see Table 3). In other words, the Procrustes method is applied only to a small portion of the sense vectors. In both

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 5. $\rho \times 100$ of (MaxSim/AvgSim) on semantic relatedness benchmark datasets

Embedding		MEN	MTurk	RW	WS353	Macro	Micro
BERT 3072d		51.4	43.4	24.2	43.9	40.7	40.8
DistilBERT 3072d		57.4	45.2	33.3	55.5	47.8	48.0
RoBERTa 3072d		22.2	17.6	18.8	33.2	23.0	21.5
T-XL 4096d		59.0	45.3	23.4	48.4	44.0	44.9
GloVe	Vanilla	65.7	61.9	30.3	50.3	52.1	51.9
50d	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
	Procrustes-all	**68.8**/**68.7**	65.0/**64.9**	33.5/33.3	**53.2**/**54.0**	**55.1**/**55.2**	**55.0**/**54.9**
GloVe	Vanilla	74.9	63.3	37.7	60.9	59.2	60.1
300d	Retro	72.3/76.8	61.6/64.0	19.1/33.7	54.9/63.6	52.0/59.5	51.6/59.9
	GenSense-syn	76.4/76.7	66.6/66.3	41.7/**41.7**	60.9/63.1	61.4/62.0	62.5/62.8
	GenSense-ant	74.5/74.1	64.4/64.6	38.6/38.2	59.0/57.6	59.1/58.6	60.2/59.7
	GenSense-all	77.3/77.2	**67.2**/**66.9**	41.6/41.1	**63.0**/**64.1**	**62.3**/62.3	63.1/62.9
	Procrustes-all	**77.4**/**77.3**	**67.2**/**66.9**	**41.8**/41.3	**63.0**/**64.1**	**62.3**/**62.4**	**63.2**/**63.0**
FastText	Vanilla	71.0	52.7	43.2	61.6	57.1	59.6
50d	Retro	39.7/42.9	38.5/42.2	24.8/27.7	23.4/26.9	31.6/34.9	33.3/36.4
	GenSense-syn	72.3/72.1	55.9/54.1	**47.8**/**47.0**	61.2/61.6	**59.3**/**58.7**	**62.0**/**61.6**
	GenSense-ant	71.0/69.1	53.6/51.8	43.5/43.0	60.8/59.3	57.2/55.8	59.6/58.3
	GenSense-all	**72.6**/**72.3**	**56.4**/**54.3**	45.7/45.3	**62.6**/**62.5**	**59.3**/58.6	61.5/61.1
	Procrustes-all	**72.6**/**72.3**	56.2/54.2	45.6/45.3	62.5/**62.5**	59.2/58.6	61.4/61.1
FastText	Vanilla	84.6	72.6	56.2	78.6	73.0	73.4
300d	Retro	36.3/39.0	35.6/38.7	25.5/27.4	21.4/24.3	29.7/32.4	31.5/33.9
	GenSense-syn	82.6/84.6	72.6/74.7	54.8/**58.3**	72.0/77.1	70.5/73.7	71.5/**74.2**
	GenSense-ant	**84.6**/81.8	72.7/69.0	**56.5**/54.9	78.5/74.4	**73.1**/70.0	**73.5**/71.1
	GenSense-all	83.4/84.7	73.9/74.8	56.2/57.7	74.8/78.3	72.1/**73.9**	72.6/**74.2**
	Procrustes-all	83.4/**84.8**	**74.0**/**74.9**	56.1/57.6	74.9/78.4	72.1/**73.9**	72.6/**74.2**
Word2Vec	Vanilla	65.1	52.7	40.2	53.4	52.9	54.8
50d	Retro	58.3/64.3	49.8/54.0	16.6/29.8	45.0/**55.6**	42.4/50.9	42.1/50.9
	GenSense-syn	67.9/**67.5**	54.9/54.7	**41.9**/**42.5**	53.1/54.3	54.4/**54.8**	57.0/**57.1**
	GenSense-ant	64.4/62.8	53.4/51.8	39.8/39.5	52.3/49.9	52.5/51.0	54.3/53.1
	GenSense-all	68.2/67.3	56.6/**55.6**	41.3/41.6	**54.1**/54.7	55.0/**54.8**	**57.1**/56.7
	Procrustes-all	**68.3**/67.3	**56.8**/**55.6**	41.3/41.6	54.0/54.8	**55.1**/**54.8**	**57.1**/56.7

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 5. Continued.

Embedding		MEN	MTurk	RW	WS353	Macro	Micro
Word2Vec	Vanilla	77.1	66.9	50.9	69.1	66.0	66.7
300d	Retro	69.6/75.2	61.5/67.1	22.4/37.8	51.9/65.2	51.3/61.3	51.1/60.7
	GenSense-syn	78.2/**78.4**	69.8/**69.5**	51.5/**53.6**	65.2/69.3	66.2/**67.7**	67.4/**68.5**
	GenSense-ant	77.0/74.7	67.8/65.7	51.2/50.4	68.4/64.9	66.1/63.9	66.8/64.9
	GenSense-all	78.5/78.3	70.6/**69.5**	**52.1**/52.9	66.9/69.7	**67.0**/67.6	**67.9**/68.2
	Procrustes-all	**78.6**/**78.4**	**70.7**/**69.5**	52.0/52.8	66.9/**69.8**	**67.0**/67.6	**67.9**/68.2

of the additional evaluation metrics, the GenSense model outperforms Retro by a large margin. Procrustes-all is the best among the proposed models. These two metrics attest the robustness of our proposed model in comparison to the Retro model. For classic word embedding models, FastText outperforms Word2Vec, and Word2Vec outperforms GloVe, but not in all tasks. There is a clear gap between the 50d and 300d in all the models. Similar trend can be found in other nlp tasks (Yin and Shen 2018).

The performance between transformer models and GenSense models may not be able to compare directly as their training corpus and dimensionality are very different. From the result, only DistilBERT 3072d outperforms the vanilla GloVe 50d model in RW and WS353 and has a performance gap when comparing to GloVe 300d, FastText 300d, and Word2Vec 300d in all the datasets. RoBERTa is the worst among the transformer models. The poor performance of the contextualized word representation model may be due to the fact that they cannot accurately capture the semantic equivalence of contexts (Shi *et al*. 2019). Another possible reason is the best configuration of the transformer models is not explored. Configurations like: How many layer(s) should we select?; How to combine the selected layers (concatenate, average, max pooling, or min pooling)? Although the performance of the transformers is poor here, we will show transformer models outperform GenSense using the same setting in the later experiment that involves context (Section 6.2).

We also conducted an experiment to evaluate the benefits yielded by the relation strength. We ran GenSense-syn over the Roget ontology with a grid of $(\alpha_1, \alpha_2, \alpha_3)$ parameters. Each parameter was tuned from 0.0 to 1.0 with a step size of 0.1. The default setting of GenSense was set to (1.0, 1.0, 1.0). Table 6 shows the MaxSim results and Table 7 shows the AvgSim results. 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 Tables 6 and 7. From Table 6, we observe that the default setting yields relatively good results in comparison to the best case. Another point worth mentioning is that the worst performance occurs under the 0.1/1/0.1 setting, except for the WS353 dataset. Similar results can be found in Table 7's AvgSim metric. Since α_1 is to control the importance of the distance between the original vector and the retrofitted vector, small α_1 leads to poor performance suggests that the original trained word vector should not deviate too far. When observing the best cases, we found α_1, α_2, and α_3 are closed to each other in many tasks. For a deeper analysis on how the parameters affect the performance, Figure 3 shows the histogram of the performance when tuning $\alpha_1/\alpha_2/\alpha_3$ on all the dataset. From Figure 3, we find that in the tasks the distribution is left-skewed except the AvgSim of RW, suggesting the robustness of GenSense-syn.

We also ran GenSense-syn over the Roget ontology with another grid of parameters that contains a higher range. Specifically, the grid parameters were tuned from the parameter set {0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 5.0, 10.0}. The results are shown in Tables 8 and 9. From the results,

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 6. $\rho \times 100$ of MaxSim on semantic relatedness benchmark datasets

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

Table 7. $\rho \times 100$ of AvgSim on semantic relatedness benchmark datasets

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

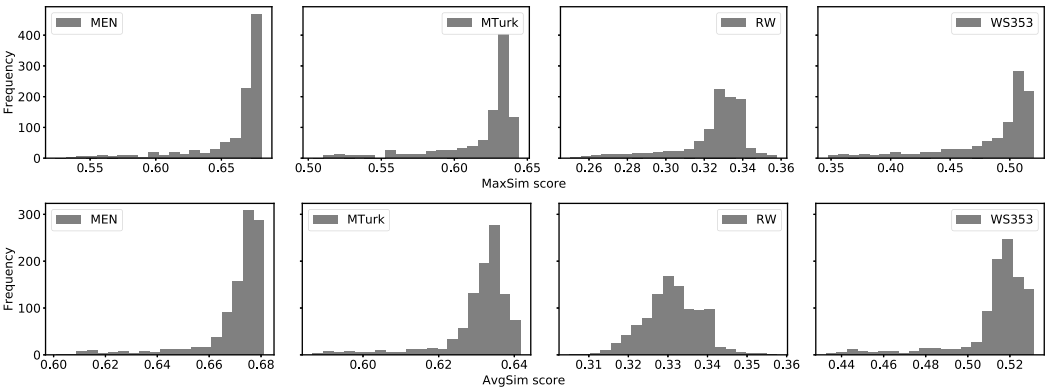

Figure 3. Histogram of all the combinations and the corresponding performance (MaxSim and AvgSim).

we find that the improvements for the best cases are almost the same as those in Tables 6 and 7 (MEN, MTurk, and WS353). Only RW's performance increase is larger. In contrast, the worst case drops considerably for all datasets. For example, MEN's MaxSim drops from 52.4 to 37.0, a 15.4 drop. This shows the importance of carefully selecting parameters in the learning model. Also worth mentioning is that these worst cases happen when α_2 is large, showing the negative effect of too much weight for the synonym neighbors.

In addition to the parameters, it is also worth analyzing the impact of dimensionality. Figure 4 shows the $\rho \times 100$ of MaxSim on the semantic relatedness benchmark datasets as a function of the vector dimension. All GloVe pre-trained models were trained on the 6-billion-token corpus of 50d, 100d, 200d, and 300d. We used the GenSense-all model on the GloVe pre-trained models. In Figure 4, the proposed GenSense-all outperforms GloVe in all the datasets for all the tested dimensions. In GloVe's original paper, they showed GloVe's performance (in terms of accuracy) to be proportional to the dimension between 50d and 300d. In this experiment, we show that both GloVe and GenSense-all's performance is proportional to dimension between 50d and 300d in terms of $\rho \times 100$ of MaxSim. Similar results are found for the AvgSim metric.

Table 10 shows the selected MEN's word pairs and their corresponding GenSense-all, GloVe, and Retro scores for case study. For GenSense-all, GloVe, and Retro, we sorted the MaxSim

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 8. $\rho \times 100$ of MaxSim on semantic relatedness benchmark datasets

GenSense	MEN	$\alpha_1/\alpha_2/\alpha_3$	MTurk	$\alpha_1/\alpha_2/\alpha_3$	RW	$\alpha_1/\alpha_2/\alpha_3$	WS353	$\alpha_1/\alpha_2/\alpha_3$
default	67.6	1./1./1.	64.1	1./1./1.	33.8	1./1./1.	50.5	1./1./1.
worst	37.0	.1/10./.1	41.2	.1/10./.1	21.6	.1/10./.1	27.7	.1/10./.1
best	68.1	1./.5/3.	64.4	2./1.5/2.	36.2	1./.5/10.	52.0	5./3./1.5

Table 9. $\rho \times 100$ of AvgSim on semantic relatedness benchmark datasets

GenSense	MEN	$\alpha_1/\alpha_2/\alpha_3$	MTurk	$\alpha_1/\alpha_2/\alpha_3$	RW	$\alpha_1/\alpha_2/\alpha_3$	WS353	$\alpha_1/\alpha_2/\alpha_3$
default	67.9	1./1./1.	64.0	1./1./1.	33.6	1./1./1.	52.8	1./1./1.
worst	49.1	.1/10./.1	52.8	.1/10./.5	26.0	.1/10./.1	35.3	.1/10./.1
best	68.1	1./1./1.5	64.2	1.5/2.5/2.	37.9	.1/.1/10.	53.0	1.5/2./2.

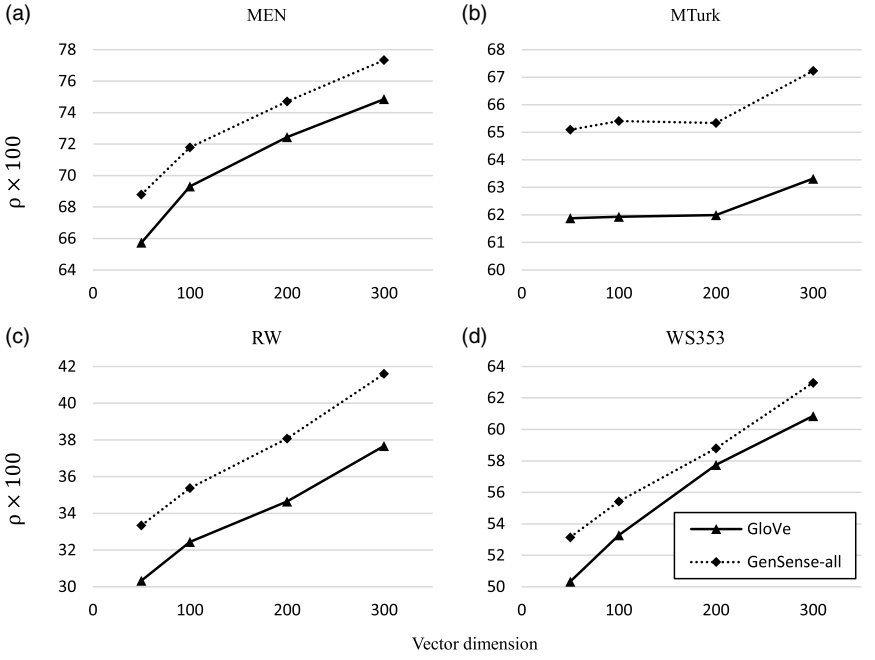

Figure 4. $\rho \times 100$ of MaxSim on semantic relatedness benchmark datasets as a function of vector dimension. Compared with the GloVe model.

scores and re-scaled them to MEN's score distribution. From Table 10, we find that Gensense-all improves the pre-trained word-embedding model (in terms of closeness to MEN's score; smaller is better) in the following situations: (1) both words have few senses $(lizard, reptiles)$, (2) both words have many senses $(stripes, train)$, and (3) one word has many senses and one word has few senses $(rail, railway)$. In other words, GenSense-all handles all possible situations well. In some cases, the Retro model increases the closeness to MEN's score.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 10. Selected MEN's word pairs and their score differences from GenSense-all, GloVe, and Retro models (smaller is better)

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 11. $\rho \times 100$ of (MaxSim/AvgSim) of standardization on dimensions on semantic relatedness benchmark datasets

	MEN	MTurk	RW	WS353	Macro	Micro
GloVe	65.7	61.9	30.3	50.3	52.1	51.9
GloVe-Z	65.2	62.5	26.3	51.4	51.4	50.3
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
GenSense-Z	**69.3**/**69.9**	**66.7**/**67.2**	31.3/31.2	53.8/54.8	55.3/55.8	54.6/55.0
GenSense-Z-last	68.3/68.4	66.1/65.6	34.4/34.8	**54.5**/**55.7**	55.8/56.1	55.2/55.4
Z-GenSense	**69.3**/**69.9**	66.5/67.1	**34.5**/**35.0**	53.8/54.9	**56.0**/**56.7**	**55.7**/**56.3**
Z-first-GenSense	68.6/68.8	65.3/66.2	34.2/34.3	54.1/54.2	55.6/55.9	55.2/55.4

6.1.1. Standardization on dimensions

Table 11 shows results of the vanilla word embedding (GloVe), the standardized vanilla word embedding (GloVe-Z), GenSense-all, and the four standardized GenSense methods (rows 4 to 7). The results show that standardization on the vanilla word embedding (GloVe-Z) improves some datasets but not all. In contrast, standardization on GenSense outperforms both GenSense-all and the GloVe-Z models. This suggests that the vanilla word embedding may not be optimized well. Although there is no model that consistently performs the best of all the standardization models, overall Z-GenSense performs the best in terms of Macro and Micro metrics.

6.1.2. Neighbor expansion from nearest neighbors

Table 12 shows the vanilla word embedding (GloVe), the Retro model, and the neighbor expansion model. The results verify our assumption that nearest neighbors play an important role in the GenSense model. From Table 12, GenSense-NN outperforms GloVe and Retro on all the benchmark datasets.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 12. $\rho \times 100$ of (MaxSim/AvgSim) of neighbor expansion from nearest neighbors on semantic relatedness benchmark datasets

	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-NN	**69.9**/**69.5**	**65.9**/**65.2**	**32.0**/**31.5**	**53.7**/**54.2**	**55.4**/**55.1**	**55.1**/**54.7**

Table 13. $\rho \times 100$ of (MaxSim/AvgSim) of combination of standardization and neighbor expansion on semantic relatedness benchmark datasets

	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-NN-Z	68.7/69.2	64.9/65.7	34.4/**34.7**	53.1/54.8	55.3/56.1	55.2/55.8
GenSense-NN-Z-last	**70.5**/**70.0**	**66.7**/65.0	**34.5**/33.9	**55.5**/**55.9**	**56.8**/**56.2**	**56.5**/**55.9**
GenSense-Z-NN	68.6/69.1	64.7/**65.8**	**34.5**/**34.7**	53.3/55.1	55.3/**56.2**	55.2/55.7
GenSense-Z-NN-first	67.4/67.6	63.3/64.4	33.2/33.4	53.4/53.9	54.3/54.8	54.1/54.3

6.1.3. Combination of standardization and neighbor expansion

Table 13 shows the experimental results of the vanilla word embedding (GloVe), the Retro model, and four combination models (GenSense-NN-Z, GenSense-NN-Z-last, GenSense-Z-NN, and GenSense-Z-NN-first). The overall best model is GenSense-NN-Z-last (in terms of Macro and Micro). GenSense-Z-NN also performs the best in Macro's AvgSim. Again, there is no dominant model (that outperforms all other models) among the combination models, but almost all models outperform the baseline models GloVe and Retro. Tables 12 and 13 suggest that all the benchmark datasets can be further improved.

6.2. Contextual word similarity

Table 14 shows the Spearman correlation $\rho \times 100$ of SCWS dataset. Unlike other models, in the DistilBERT we firstly embed the entire sentence and then extract the embedding of the word. After extracting the embedding of the pair of the words, we compute their cosine similarity. With the sense-level information, both GenSense and Retro outperform the word embedding model GloVe. The GenSense model slightly outperforms Retro. The results suggest that the negative relation information in GenSense-ant may not be helpful. We suspect that the quality of SCWS may not be controlled well. As there are 10 subjects for each question in SCWS, we further analyzed the distribution of the ranges in SCWS. We found that there are many questions with a large range, reflecting the vagueness of the questions. Overall, 35.5% of the questions had a range of 10 (i.e., some subjects assigned the highest score and some assigned the lowest score), and 50.0% had a range of 9 or 10. Unlike the semantic related experiment's result, all the transformer models outperform the GenSense models. The result is not surprising as the contextualized word embedding models are pre-trained to better represent the target word given its context through masked language modeling and next sentence prediction tasks.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 14. $\rho \times 100$ of (MaxSimC/AvgSimC) on SCWS dataset

	SCWS
BERT 3072d	64.5
DistilBERT 3072d	**65.6**
RoBERTa 3072d	57.3
T-XL 4096d	57.9
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 15. (*Accuracy*, *Precision*, *Recall*) $\times 100$ of (MaxSimD/AvgSimD) on the semantic difference dataset

	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

6.3. Semantic difference

Table 15 shows the results of the semantic difference experiment. We observe that GenSense outperforms Retro and GloVe with small margin, and the accuracy of Retro decreases in this experiment. As this task focuses on concepts, we find that synonym and antonym information is not very useful when comparing the results with GloVe. This experiment also suggests that further information about concepts is required to improve performance. Surprisingly, the antonym relation plays an important role when computing the semantic difference, especially in the AvgSimD metric.

6.4. Synonym selection

Table 16 shows the results of the synonym selection experiment: GenSense-all outperforms the baseline models on the RD-300 and TOEFL-80 datasets. In ESL-50, the best model is Retro, showing that improvements are still possible for the default GenSense model. Nevertheless, in ESL-50 GenSense-syn and GenSense-all outperform the vanilla GloVe embedding by a large margin. We also note the relatively poor performance of GenSense-ant in comparison to GenSense-syn and GenSense-all; this shows that the antonym information is relatively unimportant in the synonym selection task.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Table 16. *Accuracy* × 100 of (MaxSim/AvgSim) on the synonym selection datasets ESL-50, RD-300, and TOEFL-80

	ESL-50	RD-300	TOEFL-80	Macro	Micro
GloVe	47.9	56.5	71.8	58.7	58.3
Retro	**70.8**/**64.6**	67.1/64.7	85.9/80.8	**74.6**/**70.0**	71.0/67.7
GenSense-syn	64.6/62.5	69.4/**67.1**	**87.2**/79.5	73.7/69.7	72.2/**68.9**
GenSense-ant	47.9/50.0	56.5/58.8	75.6/68.0	60.0/58.9	59.1/59.5
GenSense-all	66.7/58.3	**70.6**/64.7	84.6/**82.1**	74.0/68.4	**72.8**/67.2

7. Limitations and future directions

This research focuses on generalizing sense retrofitting models and evaluates the models on semantic relatedness, contextual word similarity, semantic difference, and synonym selection datasets. In the semantic relatedness experiment, we compare GenSense and BERT family models and show that GenSense can outperform BERT models. However, the experiment may be unfair to the BERT family models as they are context sensitive language models, while each phrase pair in the dataset is context-free. The slight difference of the dataset and models nature makes comparisons are not as easy to interpret. A possible way to address the issue is to apply the generalized sense representation learnt by the proposed method in downstream natural language processing applications to conduct extrinsic evaluations. If the downstream tasks involve context-sensitive features, the tasks themselves will give advantage to the context sensitive models. Nevertheless, it would be interesting to evaluate the embeddings extrinsically via training neural network models of the same architecture for downstream tasks (e.g., named entity recognition (Santos, Consoli, and Vieira 2020) and comparing different language models. Another research direction that relates to this research is fair NLP. Since word embeddings are largely affected by corpus statistics, many works focus on debiasing word embeddings, especially in gender, human race, and society (Bolukbasi *et al*. 2016; Caliskan, Bryson, and Narayanan 2017; Brunet *et al*. 2019). How to incorporate the techniques in debiasing word embeddings to GenSense model is worth exploring. Finally, social sciences, psychology, and history research fields largely depend on high quality word or sense embedding models (Hamilton, Leskovec, and Jurafsky 2016; Caliskan *et al*. 2017; Jonauskaite *et al*. 2021). Our proposed GenSense embedding model can bring great value to these research fields.

8. Conclusions

In this paper, we present GenSense, a generalized framework for learning sense embeddings. As GenSense belongs to post-processing retrofitting model family, it enjoys all the benefits of retrofitting, such as shorter training times and lower memory requirements. In the generalization, (1) we extend the synonym relation to the positive contextual neighbor relation, the antonym relation, and the negative contextual neighbor relation; (2) we consider the semantic strength of each relation; and (3) we use the relation strength between relations to balance different components. We conduct experiments on four types of tasks: semantic relatedness, contextual word similarity, semantic difference, and synonym selection. Experimental results show that GenSense outperforms previous approaches. In the experiment with grid search to evaluate the benefits yielded by the relation strength, we find a 87.7% performance difference between the worst and the best cases in WS353 dataset. Based on the proposed GenSense, we also propose a standardization process on the dimensions with four settings, a neighbor expansion process from the nearest neighbors, and four different combinations of the two approaches. Finally, we propose a Procrustes analysis

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

approach that inspired from bilingual mapping models for learning representations that outside of the ontology. The experimental results show the advantages of these modifications on the semantic relatedness task. Finally, we have released the source code and the pre-trained model as a resource for the research community.[e,f] Other versions of the sense-retrofitted embeddings can be found on 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.

References

Artetxe M., Labaka G. and Agirre E. (2016). Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, TX, USA. Association for Computational Linguistics, pp. 2289–2294.

Azzini A., da Costa Pereira C., Dragoni M. and Tettamanzi A.G. (2011). A neuro-evolutionary corpus-based method for word sense disambiguation. *IEEE Intelligent Systems* **27**(6), 26–35.

Bengio Y., Delalleau O. and Le Roux N. (2006). Label propagation and quadratic criterion. In *Semi-Supervised Learning*.

Bian J., Gao B. and Liu T.-Y. (2014). Knowledge-powered deep learning for word embedding. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, Nancy, France. Springer, pp. 132–148.

Bojanowski P., Grave E., Joulin A. and Mikolov T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* **5**, 135–146.

Bollegala D., Alsuhaibani M., Maehara T. and Kawarabayashi K.-i. (2016). Joint word representation learning using a corpus and a semantic lexicon. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, Phoenix, AZ, USA. AAAI Press, pp. 2690–2696.

Bolukbasi T., Chang K.-W., Zou J.Y., Saligrama V. and Kalai A.T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS)*, vol. 29, Barcelona, Spain. Curran Associates, Inc.

Brunet M.-E., Alkalay-Houlihan C., Anderson A. and Zemel R. (2019). Understanding the origins of bias in word embeddings. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, Long Beach, CA, USA. PMLR, pp. 803–811.

Bruni E., Tran N.-K. and Baroni M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research* **49**, 1–47.

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

Caliskan A., Bryson J.J. and Narayanan A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science* **356**(6334), 183–186.

Camacho-Collados J. and Pilehvar M.T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research* **63**, 743–788.

Camacho-Collados J., Pilehvar M.T. and Navigli R. (2015). Nasari: A novel approach to a semantically-aware representation of items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Denver, CO, USA. Association for Computational Linguistics, pp. 567–577.

Dai Z., Yang Z., Yang Y., Carbonell J.G., Le Q. and Salakhutdinov R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy. Association for Computational Linguistics, pp. 2978–2988.

Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K. and Harshman R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**(6), 391–407.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (NAACL-HLT)*, Minneapolis, MN, USA. Association for Computational Linguistics, pp. 4171–4186.

Dolan B., Quirk C. and Brockett C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland. COLING, pp. 350–356.

[e] http://nlg.csie.ntu.edu.tw/nlpresource/GenSense.

[f] https://github.com/y95847frank/GenSense.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

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

Ethayarajh K. (2019). How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 55–65.

Ettinger A., **Resnik P. and Carpuat M.** (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 (NAACL-HLT)*, San Diego, CA, USA. Association for Computational Linguistics, pp. 1378–1383.

Faruqui M., **Dodge J.**, **Jauhar S.K.**, **Dyer C.**, **Hovy E. and Smith N.A.** (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 (NAACL-HLT)*, Denver, CO. Association for Computational Linguistics, pp. 1606–1615.

Finkelstein L., **Gabrilovich E.**, **Matias Y.**, **Rivlin E.**, **Solan Z.**, **Wolfman G. and Ruppin E.** (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems* **20**(1), 116–131.

Ganitkevitch J., **Van Durme B. and Callison-Burch C.** (2013). PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Atlanta, GA, USA. Association for Computational Linguistics, pp. 758–764.

Glavaš G. and Vulić I. (2018). Explicit retrofitting of distributional word vectors. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) ACL*, Melbourne, Australia. Association for Computational Linguistics, pp. 34–45.

Hamilton W.L., **Leskovec J. and Jurafsky D.** (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, Berlin, Germany. Association for Computational Linguistics, pp. 1489–1501.

Huang E.H., **Socher R.**, **Manning C.D. and Ng A.Y.** (2012). Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL)*, Jeju Island, Korea. Association for Computational Linguistics, pp. 873–882.

Iacobacci I., **Pilehvar M.T. and Navigli R.** (2015). Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (ACL-IJCNLP)*, Beijing, China. Association for Computational Linguistics, pp. 95–105.

Jarmasz M. and Szpakowicz S. (2004). Roget's thesaurus and semantic similarity. In *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003, 111.

Jauhar S.K., **Dyer C. and Hovy E.** (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 (NAACL-HLT)*, Denver, CO, USA. Association for Computational Linguistics, pp. 683–693.

Jonauskaite D., **Sutton A.**, **Cristianini N. and Mohr C.** (2021). English colour terms carry gender and valence biases: A corpus study using word embeddings. *PLoS ONE* **16**(6), e0251559.

Joulin A., **Bojanowski P.**, **Mikolov T.**, **Jégou H. and Grave E.** (2018). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium. Association for Computational Linguistics, pp. 2979–2984.

Kipfer B.A. (1993). *Roget's 21st Century Thesaurus in Dictionary Form: The Essential Reference for Home, School, or Office*. Laurel.

Krebs A. and Paperno D. (2016). Capturing discriminative attributes in a distributional space: Task proposal. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, Berlin, Germany. Association for Computational Linguistics, pp. 51–54.

Landauer T.K. and Dumais S.T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* **104**(2), 211–240.

Leacock C. and Chodorow M. (1998). Combining local context and WordNet similarity for word sense identification. *WordNet: An Electronic Lexical Database* **49**(2), 265–283.

Lee Y.-Y., **Ke H.**, **Huang H.-H. and Chen H.-H.** (2016). Combining word embedding and lexical database for semantic relatedness measurement. In *Proceedings of the 25th International Conference Companion on World Wide Web (WWW)*, Montréal, Québec, Canada. International World Wide Web Conferences Steering Committee, pp. 73–74.

Lee Y.-Y., **Yen T.-Y.**, **Huang H.-H. and Chen H.-H.** (2017). Structural-fitting word vectors to linguistic ontology for semantic relatedness measurement. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM)*, Singapore, Singapore. Association for Computing Machinery, pp. 2151–2154.

Lee Y.-Y., **Yen T.-Y.**, **Huang H.-H.**, **Shiue Y.-T. and Chen H.-H.** (2018). Gensense: A generalized sense retrofitting model. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, Santa Fe, NM, USA. Association for Computational Linguistics, pp. 1662–1671.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Lengerich B.J., **Maas A.L. and Potts C.** (2017). Retrofitting distributional embeddings to knowledge graphs with functional relations. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, Santa Fe, NM, USA. Association for Computational Linguistics.

Li J. and Jurafsky D. (2015). Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal. Association for Computational Linguistics, pp. 1722–1732.

Lin, **D.** (1998). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML)*, vol. 98, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc., pp. 296–304.

Lin D. and Pantel P. (2001). Dirt – discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, San Francisco, CA, USA. Association for Computing Machinery, pp. 323–328.

Liu X., **Nie J.-Y. and Sordoni A.** (2016). Constraining word embeddings by prior knowledge–application to medical information retrieval. In *Information Retrieval Technology*. Beijing, China: Springer International Publishing, pp. 155–167.

Liu Y., **Ott M.**, **Goyal N.**, **Du J.**, **Joshi M.**, **Chen D.**, **Levy O.**, **Lewis M.**, **Zettlemoyer L. and Stoyanov V.** (2019). Roberta: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Loureiro D. and Jorge A. (2019). Language modelling makes sense: Propagating representations through wordnet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy. Association for Computational Linguistics, pp. 5682–5691.

Luong T., **Socher R. and Manning C.** (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL)*, Sofia, Bulgaria. Association for Computational Linguistics, pp. 104–113.

Mancini M., **Camacho-Collados J.**, **Iacobacci I. and Navigli R.** (2017). Embedding words and senses together via joint knowledge-enhanced training. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL)*, Vancouver, Canada. Association for Computational Linguistics, pp. 100–111.

Maneewongvatana S. and Mount D.M. (1999). It's okay to be skinny, if your friends are fat. In *Center for Geometric Computing 4th Annual Workshop on Computational Geometry*, vol. 2, pp. 1–8.

Mikolov T., **Chen K.**, **Corrado G. and Dean J.** (2013a). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov T., **Le**, **Q.V. and Sutskever I.** (2013b). Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168.

Miller G.A. (1998). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Mrkšic N., **OSéaghdha D.**, **Thomson B.**, **Gašic M.**, **Rojas-Barahona L.**, **Su P.-H.**, **Vandyke D.**, **Wen T.-H. and Young S.** (2016). Counter-fitting word vectors to linguistic constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, San Diego, California. Association for Computational Linguistics, pp. 142–148.

Pavlick E., **Rastogi P.**, **Ganitkevitch J.**, **Van Durme B. and Callison-Burch C.** (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 (Volume 2: Short Papers) (ACL-IJCNLP)*, Beijing, China. Association for Computational Linguistics, pp. 425–430.

Pennington J., **Socher R. and Manning C.** (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics, pp. 1532–1543.

Peters M.E., **Neumann M.**, **Iyyer M.**, **Gardner M.**, **Clark C.**, **Lee K. and Zettlemoyer L.** (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (NAACL)*, New Orleans, Louisiana. Association for Computational Linguistics.

Qiu L., **Tu K. and Yu Y.** (2016). Context-dependent sense embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, Texas. Association for Computational Linguistics, pp. 183–191.

Quirk C., **Brockett C. and Dolan W.B.** (2004). Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain. Association for Computational Linguistics, pp. 142–149.

Radford A., **Wu J.**, **Child R.**, **Luan D.**, **Amodei D. and Sutskever**, **I.** (2019). Language models are unsupervised multitask learners. *OpenAI Blog* **1**(8), 9.

Radinsky K., **Agichtein E.**, **Gabrilovich E. and Markovitch S.** (2011). A word at a time: Computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web (WWW)*. New York, NY, USA: Association for Computing Machinery, pp. 337–346.

Reisinger J. and Mooney R.J. (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 (NAACL-HLT)*, Los Angeles, CA, USA. Association for Computational Linguistics, pp. 109–117.

https://doi.org/10.1017/S1351324922000523 Published online by Cambridge University Press

Remus S. and Biemann C. (2018). Retrofitting word representations for unsupervised sense aware word similarities. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan. European Language Resources Association.

Sanh V., **Debut L.**, **Chaumond J. and Wolf T.** (2019). Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Santos J., **Consoli B. and Vieira R.** (2020). Word embedding evaluation in downstream tasks and semantic analogies. In *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC)*, Marseille, France. European Language Resources Association, pp. 4828–4834.

Scarlini B., **Pasini T. and Navigli R.** (2020). Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* **34**(05), 8758–8765.

Shi W., **Chen M.**, **Zhou P. and Chang K.-W.** (2019). Retrofitting contextualized word embeddings with paraphrases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics.

Smith S.L., **Turban D.H.**, **Hamblin S. and Hammerla N.Y.** (2017). Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations (ICLR)*, Toulon, France. OpenReview.net.

Sun F., **Guo J.**, **Lan Y.**, **Xu J. and Cheng X.** (2016). Inside out: Two jointly predictive models for word representations and phrase representations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 30, Phoenix, AZ, USA. AAAI Press.

Turney P.D. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 491–502.

Wiedemann G., **Remus S.**, **Chawla A. and Biemann C.** (2019). Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS)*, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Wu Z. and Palmer M. (1994). Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Las Cruces, NM, USA. Association for Computational Linguistics, pp. 133–138.

Xing C., **Wang D.**, **Liu C. and Lin Y.** (2015). Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Denver, CO, USA. Association for Computational Linguistics, pp. 1006–1011.

Yen T.-Y., **Lee Y.-Y.**, **Huang H.-H. and Chen H.-H.** (2018). That makes sense: Joint sense retrofitting from contextual and ontological information. In *Companion Proceedings of the Web Conference 2018 (WWW)*, Lyon, France. International World Wide Web Conferences Steering Committee, pp. 15–16.

Yin Z. and Shen Y. (2018). On the dimensionality of word embedding. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS)*, vol. 31, Montréal, Canada. Curran Associates, Inc., pp. 895–906.

Yu M. and Dredze M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (ACL)*, Baltimore, MD, USA. Association for Computational Linguistics, pp. 545–550.

Cite this article: Lee Y-Y, Yen T-Y, Huang H-H, Shiue Y-T and Chen H-H (2023). On generalization of the sense retrofitting model. *Natural Language Engineering* **29**, 1097–1125. https://doi.org/10.1017/S1351324922000523