SENSEMBED: Learning Sense Embeddings for Word and Relational Similarity

Ignacio Iacobacci, **Mohammad Taher Pilehvar** and **Roberto Navigli**
Department of Computer Science
Sapienza University of Rome
{iacobacci,pilehvar,navigli}@di.uniroma1.it

Abstract

Word embeddings have recently gained considerable popularity for modeling words in different Natural Language Processing (NLP) tasks including semantic similarity measurement. However, notwithstanding their success, word embeddings are by their very nature unable to capture polysemy, as different meanings of a word are conflated into a single representation. In addition, their learning process usually relies on massive corpora only, preventing them from taking advantage of structured knowledge. We address both issues by proposing a multi-faceted approach that transforms word embeddings to the sense level and leverages knowledge from a large semantic network for effective semantic similarity measurement. We evaluate our approach on word similarity and relational similarity frameworks, reporting state-of-the-art performance on multiple datasets.

1 Introduction

The much celebrated word embeddings represent a new branch of corpus-based distributional semantic model which leverages neural networks to model the context in which a word is expected to appear. Thanks to their high coverage and their ability to capture both syntactic and semantic information, word embeddings have been successfully applied to a variety of NLP tasks, such as Word Sense Disambiguation (Chen et al., 2014), Machine Translation (Mikolov et al., 2013b), Relational Similarity (Mikolov et al., 2013c), Semantic Relatedness (Baroni et al., 2014) and Knowledge Representation (Bordes et al., 2013).

However, word embeddings inherit two important limitations from their antecedent corpus-based distributional models: (1) they are unable to model distinct meanings of a word as they conflate the contextual evidence of different meanings of a word into a single vector; and (2) they base their representations solely on the distributional statistics obtained from corpora, ignoring the wealth of information provided by existing semantic resources.

Several research works have tried to address these problems. For instance, basing their work on the original sense discrimination approach of Reisinger and Mooney (2010), Huang et al. (2012) applied K-means clustering to decompose word embeddings into multiple prototypes, each denoting a distinct meaning of the target word. However, the sense representations obtained are not linked to any sense inventory, a mapping that consequently has to be carried out either manually, or with the help of sense-annotated data. Another line of research investigates the possibility of taking advantage of existing semantic resources in word embeddings. A good example is the Relation Constrained Model (Yu and Dredze, 2014). When computing word embeddings, this model replaces the original co-occurrence clues from text corpora with the relationship information derived from the Paraphrase Database[1] (Ganitkevitch et al., 2013, PPDB), an automatically extracted dataset of paraphrase pairs.

However, none of these techniques have simultaneously solved both above-mentioned issues, i.e., inability to model polysemy and reliance on text corpora as the only source of knowledge. We propose a novel approach, called SENSEMBED, which addresses both drawbacks by exploiting semantic knowledge for modeling arbitrary word senses in a large sense inventory. We evaluate our representation on multiple datasets in two standard tasks: word-level semantic similarity and relational similarity. Experimental results show that moving from words to senses, while making use

[1] http://paraphrase.org/#/download

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 95–105,
Beijing, China, July 26-31, 2015.

of lexical-semantic knowledge bases, makes embeddings significantly more powerful, resulting in consistent performance improvement across tasks.

Our contributions are twofold: (1) we propose a knowledge-based approach for obtaining continuous representations for individual word senses; and (2) by leveraging these representations and lexical-semantic knowledge, we put forward a semantic similarity measure with state-of-the-art performance on multiple datasets.

2 Sense Embeddings

Word embeddings are vector space models (VSM) that represent words as real-valued vectors in a low-dimensional (relative to the size of the vocabulary) semantic space, usually referred to as the continuous space language model. The conventional way to obtain such representations is to compute a term-document occurrence matrix on large corpora and then reduce the dimensionality of the matrix using techniques such as singular value decomposition (Deerwester et al., 1990; Bullinaria and Levy, 2012, SVD). Recent predictive techniques (Bengio et al., 2003; Collobert and Weston, 2008; Mnih and Hinton, 2007; Turian et al., 2010; Mikolov et al., 2013a) replace the conventional two-phase approach with a single supervised process, usually based on neural networks.

In contrast to word embeddings, which obtain a single model for potentially ambiguous words, sense embeddings are continuous representations of individual word senses. In order to be able to apply word embeddings techniques to obtain representations for individual word senses, large sense-annotated corpora have to be available. However, manual sense annotation is a difficult and time-consuming process, i.e., the so-called knowledge acquisition bottleneck. In fact, the largest existing manually sense annotated dataset is the SemCor corpus (Miller et al., 1993), whose creation dates back to more than two decades ago. In order to alleviate this issue, we leveraged a state-of-the-art Word Sense Disambiguation (WSD) algorithm to automatically generate large amounts of sense-annotated corpora.

In the rest of Section 2, first, in Section 2.1, we describe the sense inventory used for SENSEMBED. Section 2.2 introduces the corpus and the disambiguation procedure used to sense annotate this corpus. Finally in Section 2.3 we discuss how we leverage the automatically sense-tagged dataset for the training of sense embeddings.

2.1 Underlying sense inventory

We selected BabelNet[2] (Navigli and Ponzetto, 2012) as our underlying sense inventory. The resource is a merger of WordNet with multiple other lexical resources, the most prominent of which is Wikipedia. As a result, the manually-curated information in WordNet is augmented with the complementary knowledge from collaboratively-constructed resources, providing a high coverage of domain-specific terms and named entities and a rich set of relations. The usage of BabelNet as our underlying sense inventory provides us with the advantage of having our sense embeddings readily applicable to multiple sense inventories.

2.2 Generating a sense-annotated corpus

As our corpus we used the September-2014 dump of the English Wikipedia.[3] This corpus comprises texts from various domains and topics and provides a suitable word coverage. The unprocessed text of the corpus includes approximately three billion tokens and more than three million unique words. We only consider tokens with at least five occurrences.

As our WSD system, we opted for Babelfy[4] (Moro et al., 2014), a state-of-the-art WSD and Entity Linking algorithm based on BabelNet's semantic network. Babelfy first models each concept in the network through its corresponding "semantic signature" by leveraging a graph random walk algorithm. Given an input text, the algorithm uses the generated semantic signatures to construct a subgraph of the semantic network representing the input text. Babelfy then searches this subgraph for the intended sense of each content word using an iterative process and a dense subgraph heuristic. Thanks to its use of BabelNet, Babelfy inherently features multilinguality; hence, our representation approach is equally applicable to languages other than English. In order to guarantee high accuracy and to avoid bias towards more frequent senses, we do not consider those judgements made by Babelfy while backing off to the most frequent sense, a case that happens when a certain confidence threshold is not met by the algorithm. The disambiguated items with high confidence correspond to more than 50% of all the

[2] http://www.babelnet.org/

[3] http://dumps.wikimedia.org/enwiki/

[4] http://www.babelfy.org/

bank$^n_1$ (geographical)	***bank***$^n_2$ (financial)	***number***$^n_4$ (phone)	***number***$^n_3$ (acting)	***hood***$^n_1$ (gang)	***hood***$^n_{12}$ (convertible car)
upstream$^r_1$	commercial_bank$^n_1$	calls$^n_1$	appearing$^v_6$	tortures$^n_5$	taillights$^n_1$
downstream$^r_1$	financial_institution$^n_1$	dialled$^v_1$	minor_roles$^n_1$	vengeance$^n_1$	grille$^n_2$
runs$^v_6$	national_bank$^n_1$	operator$^n_{20}$	stage_production$^n_1$	badguy$^n_1$	bumper$^n_2$
confluence$^n_1$	trust_company$^n_1$	telephone_network$^n_1$	supporting_roles$^n_1$	brutal$^a_1$	fascia$^n_2$
river$^n_1$	savings_bank$^n_1$	telephony$^n_1$	leading_roles$^n_1$	execution$^n_1$	rear_window$^n_1$
stream$^n_1$	banking$^n_1$	subscriber$^n_2$	stage_shows$^n_1$	murders$^n_1$	headlights$^n_1$

Table 1: Closest senses to two senses of three ambiguous nouns: *bank*, *number*, and *hood*

content words. As a result of the disambiguation step, we obtain sense-annotated data comprising around one billion tagged words with at least five occurrences and 2.5 million unique word senses.

2.3 Learning sense embeddings

The disambiguated text is processed with the Word2vec (Mikolov et al., 2013a) toolkit[5]. We applied Word2vec to produce continuous representations of word senses based on the distributional information obtained from the annotated corpus. For each target word sense, a representation is computed by maximizing the log likelihood of the word sense with respect to its context. We opted for the Continuous Bag of Words (CBOW) architecture, the objective of which is to predict a single word (word sense in our case) given its context. The context is defined by a window, typically with the size of five words on each side with the paragraph ending barrier. We used hierarchical softmax as our training algorithm. The dimensionality of the vectors were set to 400 and the subsampling of frequent words to 10^{-3}.

As a result of the learning process, we obtain vector-based semantic representations for each of the word senses in the automatically-annotated corpus. We show in Table 1 some of the closest senses to six sample word senses: the geographical and financial senses of *river*, the performance and phone number senses of *number*, and the gang and car senses of *hood*.[6] As can be seen, sense embeddings can capture effectively the clear distinctions between different senses of a word. Additionally, the closest senses are not necessarily constrained to the same part of speech. For instance, the river sense of *bank* has the adverbs *upstream* and *downstream* and the "move along, of liquid" sense of the verb *run* among its closest senses.

[5] `http://code.google.com/p/word2vec/`

[6] We follow Navigli (2009) and show the n^{th} sense of the *word* with part of speech x as *word*^x_n.

	Synset Description	**Synonymous senses**
hood$^n_1$	rough or violent youth	hoodlum$^n_1$, goon$^n_2$, thug$^n_1$
hood$^n_4$	photography equipment	lens_hood$^n_1$
hood$^n_9$	automotive body parts	bonnet$^n_2$, cowl$^n_1$, cowling$^n_1$
hood$^n_{12}$	car with retractable top	convertible$^n_1$

Table 2: Sample initial senses of the noun *hood* (leftmost column) and their synonym expansion (rightmost column).

3 Similarity Measurement

This Section describes how we leverage the generated sense embeddings for the computation of word similarity and relational similarity. We start the Section by explaining how we associate a word with its set of corresponding senses and how we compare pairs of senses in Sections 3.1 and 3.2, respectively. We then illustrate our approach for measuring word similarity, together with its knowledge-based enhancement, in Section 3.3, and relational similarity in Section 3.4. Hereafter, we refer to our similarity measurement approach as SENSEMBED.

3.1 Associating senses with words

In order to be able to utilize our sense embeddings for a word-level task such as word similarity measurement, we need to associate each word with its set of relevant senses, each modeled by its corresponding vector. Let $\mathcal{S}_w$ be the set of senses associated with the word w. Our objective is to cover as many senses as can be associated with the word w. To this end we first initialize the set $\mathcal{S}_w$ by the word senses of the word w and all its synonymous word senses, as defined in the BabelNet sense inventory. We show in Table 2 some of the senses of the noun *hood* and the synonym expansion for these senses. We further expand the set $\mathcal{S}_w$ by repeating the same process for the lemma of word w (if not already in lemma form).

3.2 Vector comparison

For comparing vectors, we use the *Tanimoto* distance. The measure is a generalization of Jaccard similarity for real-valued vectors in [-1, 1]:

$$\mathcal{T}(\vec{w_1}, \vec{w_2}) = \frac{\vec{w_1} \cdot \vec{w_2}}{\|\vec{w_1}\|^2 + \|\vec{w_2}\|^2 - \vec{w_1} \cdot \vec{w_2}} \quad (1)$$

where $\vec{w_1} \cdot \vec{w_2}$ is the dot product of the vectors $\vec{w_1}$ and $\vec{w_2}$ and $\|\vec{w_1}\|$ is the Euclidean norm of $\vec{w_1}$. Rink and Harabagiu (2013) reported consistent improvements when using vector space metrics, in particular the Tanimoto distance, on the SemEval-2012 task on relational similarity (Jurgens et al., 2012) in comparison to several other measures that are designed for probability distributions, such as Jensen-Shannon divergence and Hellinger distance.

3.3 Word similarity

We show in Algorithm 1 our procedure for measuring the semantic similarity of a pair of input words w_1 and w_2. The algorithm also takes as its inputs the similarity strategy and the *weighted* similarity parameter α (Section 3.3.1) along with a *graph vicinity factor* flag (Section 3.3.2).

3.3.1 Similarity measurement strategy

We take two strategies for calculating the similarity of the given words w_1 and w_2. Let $\mathcal{S}_{w_1}$ and $\mathcal{S}_{w_2}$ be the sets of senses associated with the two respective input words w_1 and w_2, and let $\vec{s_i}$ be the sense embedding vector of the sense s_i. In the first strategy, which we refer to as *closest*, we follow the conventional approach (Budanitsky and Hirst, 2006) and measure the similarity of the two words as the similarity of their closest senses, i.e.:

$$Sim_{closest}(w_1, w_2) = \max_{\substack{s_1 \in \mathcal{S}_{w_1} \\ s_2 \in \mathcal{S}_{w_2}}} \mathcal{T}(\vec{s_1}, \vec{s_2}) \quad (2)$$

However, taking the similarity of the closest senses of two words as their overall similarity ignores the fact that the other senses can also contribute to the process of similarity judgement. In fact, psychological studies suggest that humans, while judging semantic similarity of a pair of words, consider different meanings of the two words and not only the closest ones (Tversky, 1977; Markman and Gentner, 1993). For instance, the WordSim-353 dataset (Finkelstein et al., 2002) contains the word pair *brother-monk*. Despite having the religious devotee sense in common, the two words are assigned the similarity judgement of 6.27, which is slightly above the middle point in the similarity scale [0,10] of the dataset. This clearly indicates that other non-synonymous, yet still related, senses of the two words have also played a role in the similarity judgement. Additionally, the relatively low score reflects the fact that the religious devotee sense is not a dominant meaning of the word *brother*.

Algorithm 1 Word Similarity

Input: Two words w_1 and w_2
Str, the similarity strategy
Vic, the *graph vicinity factor* flag
α parameter for the *weighted* strategy
Output: The similarity between w_1 and w_2

$\mathcal{S}_{w_1} \leftarrow$ *getSenses*(w_1), $\mathcal{S}_{w_2} \leftarrow$ *getSenses*(w_2)
if *Str* **is** *closest* **then**
    sim $\leftarrow$ -1
else
    sim $\leftarrow$ 0
end if
for each $s_1 \in \mathcal{S}_{w_1}$ **and** $s_2 \in \mathcal{S}_{w_2}$ **do**
    if *Vic* **is** *true* **then**
        tmp $\leftarrow \mathcal{T}^*(\vec{s_1},\vec{s_2})$
    else
        tmp $\leftarrow \mathcal{T}(\vec{s_1},\vec{s_2})$
    end if
    if *Str* **is** *closest* **then**
        sim $\leftarrow \max$ (*sim*, *tmp*)
    else
        sim $\leftarrow$ *sim* + *tmp*$^{\alpha} \times d(s_1) \times d(s_2)$
    end if
end for

We therefore put forward another similarity measurement strategy, called *weighted*, in which different senses of the two words contribute to their similarity computation, but the contributions are scaled according to their relative importance. To this end, we first leverage sense occurrence frequencies in order to estimate the dominance of each specific word sense. For each word w, we first compute the dominance of its sense $s \in \mathcal{S}_w$ by dividing the frequency of s by the overall frequency of all senses associated with w, i.e., $\mathcal{S}_w$:

$$d(s) = \frac{freq(s)}{\sum_{s' \in \mathcal{S}_w} freq(s')} \quad (3)$$

We further recognize that the importance of a specific sense of a word can also be triggered by

the word it is being compared with. We model this by biasing the similarity computation towards closer senses, by increasing the contribution of closer senses through a power function with parameter α. The similarity of a pair of words w_1 and w_2 according to the *weighted* strategy is computed as:

$$Sim_{weighted}(w_1, w_2) = \sum_{s_1 \in \mathcal{S}_{w_1}} \sum_{s_2 \in \mathcal{S}_{w_2}} d(s_1)\, d(s_2)\, \mathcal{T}(\vec{s_1}, \vec{s_2})^{\alpha} \quad (4)$$

where the α parameter is a real-valued constant greater than one. We show in Section 4.1.3 how we tune the value of this parameter.

3.3.2 Enhancing similarity accuracy

Our similarity measurement approach takes advantage of lexical knowledge at two different levels. First, as we described in Sections 2.2 and 2.3, we use a knowledge-based disambiguation approach, i.e., Babelfy, which exploits BabelNet's semantic network. Second, we put forward a methodology that leverages the relations in BabelNet's graph for enhancing the accuracy of similarity judgements, to be discussed next.

As a distributional vector representation technique, our sense embeddings can potentially suffer from inaccurate modeling of less frequent word senses. In contrast, our underlying sense inventory provides a full coverage of all its concepts, with relations that are taken from WordNet and Wikipedia. In order to make use of the complementary information provided by our lexical knowledge base and to obtain more accurate similarity judgements, we introduce a *graph vicinity factor*, that combines the structural knowledge from BabelNet's semantic network and the distributional representation of sense embeddings. To this end, for a given sense pair, we scale the similarity judgement obtained by comparing their corresponding sense embeddings, based on their placement in the network. Let E be the set of all sense-to-sense relations provided by BabelNet's semantic network, i.e., $E = \{(s_i, s_j) : s_i - s_j\}$. Then, the similarity of a pair of words with the *graph vicinity factor* in formulas 2 and 4 is computed by replacing $\mathcal{T}$ with $\mathcal{T}^*$, defined as:

$$\mathcal{T}^*(\vec{s_1}, \vec{s_2}) = \begin{cases} \mathcal{T}(\vec{s_1}, \vec{s_2}) \times \beta, & \text{if } (s_1, s_2) \in E \\ \mathcal{T}(\vec{s_1}, \vec{s_2}) \times \beta^{-1}, & \text{otherwise} \end{cases} \quad (5)$$

We show in Section 4.1.3 how we tune the parameter β. This procedure is particularly helpful for the case of less frequent word senses that do not have enough contextual information to allow an effective representation. For instance, the SimLex-999 dataset (Hill et al., 2014), which we use as our tuning dataset (see Section 4.1.3), contains the highly-related pair *orthodontist-dentist*. We observed that the intended sense of the noun *orthodontist* occurs only 70 times in our annotated corpus. As a result, the obtained representation was not accurate, resulting in a low similarity score for the pair. The two respective senses are, however, directly connected in the BabelNet graph. Hence, the *graph vicinity factor* scales up the computed similarity value for the word pair.

3.4 Relational similarity

Relational similarity evaluates the correspondence between relations (Medin et al., 1990). The task can be viewed as an analogy problem in which, given two pairs of words (w_a, w_b) and (w_c, w_d), the goal is to compute the extent to which the relations of w_a to w_b and w_c to w_d are similar. Sense embeddings are suitable candidates for measuring this type of similarity, as they represent relations between senses as linear transformations. Given this property, the relation between a pair of words can be obtained by subtracting their corresponding normalized embeddings. Following Zhila et al. (2013), the relational similarity between two pairs of word (w_a, w_b) and (w_c, w_d) is accordingly calculated as:

$$\text{ANALOGY}(\vec{w_a}, \vec{w_b}, \vec{w_c}, \vec{w_d}) = \mathcal{T}(\vec{w_b} - \vec{w_a}, \vec{w_d} - \vec{w_c}) \quad (6)$$

We show the procedure for measuring the relational similarity in Algorithm 2. The algorithm first finds the closest senses across the two word pairs: s_a^* and s_b^* for the first pair and s_c^* and s_d^* for the second. The analogy vector representations are accordingly computed as the difference between the sense embeddings of the corresponding closest senses. Finally, the relational similarity is computed as the similarity of the analogy vectors of the two pairs.

4 Experiments

We evaluate our sense-enhanced semantic representation on multiple word similarity and relatedness datasets (Section 4.1), as well as the relational similarity framework (Section 4.2).

Algorithm 2 Relational Similarity

Input: Two pairs of words w_a, w_b and w_c, w_d

Output: The degree of analogy between the two pairs

1: $\mathcal{S}_{w_a} \leftarrow$ *getSenses*(w_a), $\mathcal{S}_{w_b} \leftarrow$ *getSenses*(w_b)

2: $(s_a^*, s_b^*) \leftarrow \underset{\substack{s_a \in \mathcal{S}_{w_a} \\ s_b \in \mathcal{S}_{w_b}}}{argmax} \mathcal{T}(\vec{s_a}, \vec{s_b})$

3: $\mathcal{S}_{w_c} \leftarrow$ *getSenses*(w_c), $\mathcal{S}_{w_d} \leftarrow$ *getSenses*(w_d)

4: $(s_c^*, s_d^*) \leftarrow \underset{\substack{s_c \in \mathcal{S}_{w_c} \\ s_d \in \mathcal{S}_{w_d}}}{argmax} \mathcal{T}(\vec{s_c}, \vec{s_d})$

5: **return:** $\mathcal{T}(\vec{s_b}^* - \vec{s_a}^*, \vec{s_d}^* - \vec{s_c}^*)$

4.1 Word similarity experiment

Word similarity measurement is one of the most popular evaluation methods in lexical semantics, and semantic similarity in particular, with numerous evaluation benchmarks and datasets. Given a set of word pairs, a system's task is to provide similarity judgments for each pair, and these judgements should ideally be as close as possible to those given by humans.

4.1.1 Datasets

We evaluate SENSEMBED on standard word similarity and relatedness datasets: the RG-65 (Rubenstein and Goodenough, 1965) and the WordSim-353 (Finkelstein et al., 2002, WS-353) datasets. Agirre et al. (2009) suggested that the original WS-353 dataset conflates similarity and relatedness and divided the dataset into two subsets, each containing pairs for just one type of association measure: similarity (the WS-Sim dataset) and relatedness (the WS-Rel dataset).

We also evaluate our approach on the YP-130 dataset, which was created by Yang and Powers (2005) specifically for measuring verb similarity, and also on the Stanford's Contextual Word Similarities (SCWS), a dataset for measuring word-in-context similarity (Huang et al., 2012). In the SCWS dataset each word is provided with the sentence containing it, which helps in pointing out the intended sense of the corresponding target word.

Finally, we also report results on the MEN dataset which was recently introduced by Bruni et al. (2014). MEN contains two sets of English word pairs, together with human-assigned similarity judgments, obtained by crowdsourcing using Amazon Mechanical Turk.

4.1.2 Comparison systems

We compare the performance of our similarity measure against twelve other approaches. As regards traditional distributional models, we report the best results computed by Baroni et al. (2014) for PMI-SVD, a system based on Pointwise Mutual Information (PMI) and SVD-based dimensionality reduction. For word embeddings, we report the results of Pennington et al. (2014, GloVe) and Collobert and Weston (2008). GloVe is an alternative way for learning embeddings, in which vector dimensions are made explicit, as opposed to the opaque meaning of the vector dimensions in Word2vec. The approach of Collobert and Weston (2008) is an embeddings model with a deeper architecture, designed to preserve more complex knowledge as distant relations. We also show results for the word embeddings trained by Baroni et al. (2014). The authors first constructed a massive corpus by combining several large corpora. Then, they trained dozens of different Word2vec models by varying the system's training parameters and reported the best performance obtained on each dataset.

As representatives for graph-based similarity techniques, we report results for the state-of-the-art approach of Pilehvar et al. (2013) which is based on random walks on WordNet's semantic network. Moreover, we present results for the graph-based approach of Zesch et al. (2008), which compares a pair of words based on the path lengths on Wiktionary's semantic network.

We also compare our word similarity measure against the multi-prototype models of Reisinger and Mooney (2010) and Huang et al. (2012), and against the approaches of Yu and Dredze (2014) and Chen et al. (2014), which enhance word embeddings with semantic knowledge derived from PPDB and WordNet, respectively. Finally, we report results for word embeddings, as our baseline, obtained using the Word2vec toolkit on the same corpus that was annotated and used for learning our sense embeddings (cf. Section 2.3).

4.1.3 Parameter tuning

Recall from Sections 3.3.1 and 3.3.2 that our algorithm has two parameters: the α parameter for the *weighted* strategy and the β parameter for the *graph vicinity factor*. We tuned these two parameters on the SimLex-999 dataset (Hill et al., 2014). We picked SimLex-999 since there are not many comparison systems in the literature that report re-

Measure	Dataset					
	RG-65	WS-Sim	WS-Rel	YP-130	MEN	Average
Pilehvar et al. (2013)	0.868	0.677	0.457	0.710	0.690	0.677
Zesch et al. (2008)	0.820	—	—	0.710	—	—
Collobert and Weston (2008)	0.480	0.610	0.380	—	0.570	—
Word2vec (Baroni et al., 2014)	0.840	0.800	0.700	—	0.800	—
GloVe	0.769	0.666	0.559	0.577	0.763	0.737
ESA	0.749	—	—	—	—	—
PMI-SVD	0.738	0.659	0.523	0.337	0.726	0.695
Word2vec	0.732	0.707	0.476	0.343	0.665	0.644
SENSEMBED$_{closest}$	**0.894**	0.756	0.645	**0.734**	0.779	0.769
SENSEMBED$_{weighted}$	0.871	**0.812**	**0.703**	0.639	**0.805**	**0.794**

Table 3: Spearman correlation performance on five word similarity and relatedness datasets.

sults on the dataset. We found the optimal values for α and β to be 8 and 1.6, respectively.

4.1.4 Results

Table 3 shows the experimental results on five different word similarity and relatedness datasets. We report the Spearman correlation performance for the two strategies of our approach as well as eight other comparison systems. SENSEMBED proves to be highly reliable on both similarity and relatedness measurement tasks, obtaining the best performance on most datasets. In addition, our approach shows itself to be equally suitable for verb similarity, as indicated by the results on YP-130.

The rightmost column in the Table shows the average performance weighted by dataset size. Between the two similarity measurement strategies, *weighted* proves to be the more suitable, achieving the best overall performance on three datasets and the best mean performance of 0.794 across the two strategies. This indicates that our assumption of considering all senses of a word in similarity computation was beneficial.

We report in Table 4 the Spearman correlation performance of four approaches that are similar to SENSEMBED: the multi-prototype models of Reisinger and Mooney (2010) and Huang et al. (2012), and the semantically enhanced models of Yu and Dredze (2014) and Chen et al. (2014). We provide results only on WS-353 and SCWS, since the above-mentioned approaches do not report their performance on other datasets. As we can see from the Table, SENSEMBED outperforms the other approaches on the WS-353 dataset. However, our approach lags behind on SCWS, highlighting the negative impact of taking the closest senses as the intended meanings. In fact, on this dataset, SENSEMBED$_{weighted}$ provides better performance owing to its taking into account other senses as well. The better performance of the multi-prototype systems can be attributed to their coarse-grained sense inventories which are automatically constructed by means of Word Sense Induction.

Measure	WS-353	SCWS
Huang et al. (2012)	0.713	0.628
Reisinger and Mooney (2010)	0.770	–
Chen et al. (2014)	–	**0.662**
Yu and Dredze (2014)	0.537	–
Word2vec	0.694	0.642
SENSEMBED$_{closest}$	0.714	0.589
SENSEMBED$_{weighted}$	**0.779**	0.624

Table 4: Spearman correlation performance of the multi-prototype and semantically-enhanced approaches on the WordSim-353 and the Stanford's Contextual Word Similarities datasets.

4.2 Relational similarity experiment

Dataset and evaluation. We take as our benchmark the SemEval-2012 task on Measuring Degrees of Relational Similarity (Jurgens et al., 2012). The task provides a dataset comprising 79 graded word relations, 10 of which are used for training and the rest for test. The task evaluated the participating systems in terms of the Spearman correlation and the MaxDiff score (Louviere, 1991).

Model	Setting			Dataset					
	Strategy	Vicinity	Expansion	RG-65	WS-Sim	WS-Rel	YP-130	MEN	Average
Word2vec	–	–		0.732	0.707	0.476	0.343	0.665	0.644
Word2vec$_{exp}$	–	–	✓	0.700	0.665	0.326	0.621	0.655	0.632
SENSEMBED	*closest*			0.825	0.693	0.488	0.492	0.712	0.690
			✓	0.844	0.714	0.562	0.681	0.743	0.728
		✓	✓	**0.894**	0.756	0.645	**0.734**	0.779	0.769
	weighted			0.877	0.776	0.639	0.446	0.783	0.762
			✓	0.864	0.783	0.665	0.591	0.773	0.761
		✓	✓	0.871	**0.812**	**0.703**	0.639	**0.805**	**0.794**

Table 6: Spearman correlation performance of word embeddings (Word2vec) and SENSEMBED on different semantic similarity and relatedness datasets.

Measure	MaxDiff	Spearman
Com	45.2	0.353
PairDirection	45.2	—
RNN-1600	41.8	0.275
UTD-LDA	—	0.334
UTD-NB	39.4	0.229
UTD-SVM	34.7	0.116
PMI baseline	33.9	0.112
Word2vec	43.2	0.288
SENSEMBED$_{closest}$	**45.9**	**0.358**

Table 5: Spearman correlation performance of different systems on the SemEval-2012 Task on Relational Similarity.

Comparison systems. We compare our results against six other systems and the PMI baseline provided by the task organizers. As for systems that use word embeddings for measuring relational similarity, we report results for RNN-1600 (Mikolov et al., 2013c) and PairDirection (Levy and Goldberg, 2014). We also report results for UTD-NB and UTD-SVM (Rink and Harabagiu, 2012), which rely on lexical pattern classification based on Naïve Bayes and Support Vector Machine classifiers, respectively. UTD-LDA (Rink and Harabagiu, 2013) is another system presented by the same authors that casts the task as a selectional preferences one. Finally, we show the performance of Com (Zhila et al., 2013), a system that combines Word2vec, lexical patterns, and knowledge base information. Similarly to the word similarity experiments, we also report a baseline based on word embeddings (Word2vec) trained on the same corpus and with the same settings as SENSEMBED.

Results. Table 5 shows the performance of different systems in the task of relational similarity in terms of the Spearman correlation and MaxDiff score. A comparison of the results for Word2vec and SENSEMBED shows the advantage gained by moving from the word to the sense level. Among the comparison systems, Com attains the closest performance. However, we note that the system is a combination of several methods, whereas SENSEMBED is based on a single approach.

4.3 Analysis

In order to analyze the impact of the different components of our similarity measure, we carried out a series of experiments on our word similarity datasets. We show in Table 6 the experimental results in terms of Spearman correlation. Performance is reported for the two similarity measurement strategies, i.e., *closest* and *weighted*, and for different system settings with and without the expansion procedure (cf. Section 3.1) and *graph vicinity factor* (cf. Section 3.3.2). As our comparison baseline, we also report results for word embeddings, obtained using the Word2vec toolkit on the same corpus and with the same configuration (cf. Section 2.3) used for learning the sense embeddings (Word2vec in the Table). The rightmost column in the Table reports the mean performance weighted by dataset size. Word2vec$_{exp}$ is the word embeddings system in which the similarity of the two words is determined in terms of the closest word embeddings among all the corresponding synonyms obtained with the expansion procedure (cf. Section 3.1).

A comparison of word and sense embeddings in the vanilla setting (with neither the expansion procedure nor *graph vicinity factor*) indicates the consistent advantage gained by moving from word

to sense level, irrespective of the dataset and the similarity measurement strategy. The consistent improvement shows that the semantic information provided more than compensates for the inherently imperfect disambiguation. Moreover, the results indicate the consistent benefit gained by introducing the *graph vicinity factor*, highlighting the fact that our combination of the complementary knowledge from sense embeddings and information derived from a semantic network is beneficial. Finally, note that the expansion procedure leads to performance improvement in most cases for sense embeddings. In direct contrast, the step proves harmful in the case of word embeddings, mainly due to their inability to distinguish individual word senses.

5 Related Work

Word embeddings were first introduced by Bengio et al. (2003) with the goal of statistical language modeling, i.e., learning the joint probability function of a sequence of words. The initial model was a Multilayer Perceptron (MLP) with two hidden layers: a shared non-linear and a regular hidden hyperbolic tangent one. Collobert and Weston (2008) deepened the original neural model by adding a convolutional layer and an extra layer for modeling long-distance dependencies. A significant contribution was later made by Mikolov et al. (2013a), who simplified the original model by removing the hyperbolic tangent layer and hence significantly speeding up the training process. Other related work includes GloVe (Pennington et al., 2014), which is an effort to make the vector dimensions in word embeddings explicit, and the approach of Bordes et al. (2013), which trains word embeddings on the basis of relationship information derived from WordNet.

Several techniques have been proposed for transforming word embeddings to the sense level. Chen et al. (2014) leveraged word embeddings in Word Sense Disambiguation and investigated the possibility of retrofitting embeddings with the resulting disambiguated words. Guo et al. (2014) exploited parallel data to automatically generate sense-annotated data, based on the fact that different senses of a word are usually translated to different words in another language (Chan and Ng, 2005). The automatically-generated sense-annotated data was later used for training sense-specific word embeddings. Huang et al. (2012) adopted a similar strategy by decomposing each word's single-prototype representation into multiple prototypes, denoting different senses of that word. To this end, they first gathered the context for all occurrences of a word and then used spherical K-means to cluster the contexts. Each cluster was taken as the context for a specific meaning of the word and hence used to train embeddings for that specific meaning (i.e., word sense). However, these techniques either suffer from low coverage as they can only model word senses that occur in the parallel data, or require manual intervention for linking the obtained representations to an existing sense inventory. In contrast, our approach enables high coverage and is readily applicable for the representation of word senses in widely-used lexical resources, such as WordNet, Wikipedia and Wiktionary, without needing to resort to additional manual effort.

6 Conclusions and Future Work

We proposed an approach for obtaining continuous representations of individual word senses, referred to as sense embeddings. Based on the proposed sense embeddings and the knowledge obtained from a large-scale lexical resource, i.e., BabelNet, we put forward an effective technique, called SENSEMBED, for measuring semantic similarity. We evaluated our approach on multiple datasets in the tasks of word and relational similarity. Two conclusions can be drawn on the basis of the experimental results: (1) moving from word to sense embeddings can significantly improve the effectiveness and accuracy of the representations; and (2) a meaningful combination of sense embeddings and knowledge from a semantic network can further enhance the similarity judgements. As future work, we intend to utilize our sense embeddings to perform WSD, as was proposed in Chen et al. (2014), in order to speed up the process and train sense embeddings on larger amounts of sense-annotated data.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant MultiJEDI No. 259234.

References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Boulder, Colorado.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 238–247, Baltimore, Maryland.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3:1137–1155.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *Advances in Neural Information Processing Systems*, volume 26, pages 2787–2795.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting Semantic Representations from Word Co-occurrence Statistics: Stop-lists, Stemming and SVD. *Behavior Research Methods*, 44:890–907.

Yee Seng Chan and Hwee Tou Ng. 2005. Scaling Up Word Sense Disambiguation via Parallel Texts. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, pages 1037–1042, Pittsburgh, Pennsylvania.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167, Helsinki, Finland.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of American Society for Information Science*, 41(6):391–407.

Lev Finkelstein, Gabrilovich Evgeniy, Matias Yossi, Rivlin Ehud, Solan Zach, Wolfman Gadi, and Ruppin Eytan. 2002. Placing Search in Context: The Concept Revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 758–764, Atlanta, Georgia.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning Sense-specific Word Embeddings By Exploiting Bilingual Resources. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 497–507, Dublin, Ireland.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving Word Representations Via Global Context And Multiple Word Prototypes. In *Proceedings of 50th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 873–882, Jeju Island, South Korea.

David A. Jurgens, Peter D. Turney, Saif M. Mohammad, and Keith J. Holyoak. 2012. Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364, Montreal, Canada.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180, Ann Arbor, Michigan.

Jordan Louviere. 1991. Best-Worst Scaling: A Model for the Largest Difference Judgments. Working paper, University of Alberta.

Arthur B. Markman and Dedre Gentner. 1993. Structural alignment during similarity comparisons. *Cognitive Psychology*, 25(4):431 – 467.

Douglas L. Medin, Robert L. Goldstone, and Dedre Gentner. 1990. Similarity involving attributes and relations: Judgments of similarity and difference are not inverses. *Psychological Science*, 1(1):64–69.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013b. Exploiting Similarities among Languages for Machine Translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 746–751, Atlanta, Georgia.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A Semantic Concordance. In *Proceedings of the Workshop on Human Language Technology*, pages 303–308, Princeton, New Jersey.

Andriy Mnih and Geoffrey Hinton. 2007. Three New Graphical Models for Statistical Language Modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648, Corvallis, Oregon.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, volume 12, pages 1532–1543, Doha, Qatar.

Mohammad Taher Pilehvar, David A. Jurgens, and Roberto Navigli. 2013. Align, Disambiguate and Walk: a Unified Approach for Measuring Semantic Similarity. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1341–1351, Sofia, Bulgaria.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-Prototype Vector-Space Models of Word Meaning. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, Los Angeles, California.

Bryan Rink and Sanda Harabagiu. 2012. UTD: Determining relational similarity using lexical patterns. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 413–418, Montreal, Canada.

Bryan Rink and Sanda Harabagiu. 2013. The Impact of Selectional Preference Agreement on Semantic Relational Similarity. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS) – Long Papers*, pages 204–215, Potsdam, Germany.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Communications of the ACM*, 8(10):627–633.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden.

Amos Tversky. 1977. Features of similarity. *Psychological Review*, 84:327–352.

Dongqiang Yang and David M. W. Powers. 2005. Measuring semantic similarity in the taxonomy of wordnet. In *Proceedings of the Twenty-eighth Australasian Conference on Computer Science*, volume 38, pages 315–322, Darlinghurst, Australia.

Mo Yu and Mark Dredze. 2014. Improving Lexical Embeddings with Semantic Knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 545–550, Baltimore, Maryland.

Torsten Zesch, Christof Müller, and Iryna Gurevych. 2008. Using Wiktionary for Computing Semantic Relatedness. In *Proceedings of the 23rd National Conference on Artificial Intelligence*, volume 2, pages 861–866, Chicago, Illinois.

Alisa Zhila, Wen-tau Yih, Christopher Meek, Geoffrey Zweig, and Tomas Mikolov. 2013. Combining Heterogeneous Models for Measuring Relational Similarity. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1000–1009, Atlanta, Georgia.