Embedding Words and Senses Together via Joint Knowledge-Enhanced Training

Massimiliano Mancini*, **Jose Camacho-Collados***, **Ignacio Iacobacci** and **Roberto Navigli**
Department of Computer Science
Sapienza University of Rome
mancini@dis.uniroma1.it
{collados,iacobacci,navigli}@di.uniroma1.it

Abstract

Word embeddings are widely used in Natural Language Processing, mainly due to their success in capturing semantic information from massive corpora. However, their creation process does not allow the different meanings of a word to be automatically separated, as it conflates them into a single vector. We address this issue by proposing a new model which learns word and sense embeddings jointly. Our model exploits large corpora and knowledge from semantic networks in order to produce a unified vector space of word and sense embeddings. We evaluate the main features of our approach both qualitatively and quantitatively in a variety of tasks, highlighting the advantages of the proposed method in comparison to state-of-the-art word- and sense-based models.

1 Introduction

Recently, approaches based on neural networks which embed words into low-dimensional vector spaces from text corpora (i.e. word embeddings) have become increasingly popular (Mikolov et al., 2013; Pennington et al., 2014). Word embeddings have proved to be beneficial in many Natural Language Processing tasks, such as Machine Translation (Zou et al., 2013), syntactic parsing (Weiss et al., 2015), and Question Answering (Bordes et al., 2014), to name a few. Despite their success in capturing semantic properties of words, these representations are generally hampered by an important limitation: the inability to discriminate among different meanings of the same word.

Previous works have addressed this limitation by automatically inducing word senses from monolingual corpora (Schütze, 1998; Reisinger and Mooney, 2010; Huang et al., 2012; Di Marco and Navigli, 2013; Neelakantan et al., 2014; Tian et al., 2014; Li and Jurafsky, 2015; Vu and Parker, 2016; Qiu et al., 2016), or bilingual parallel data (Guo et al., 2014; Ettinger et al., 2016; Šuster et al., 2016). However, these approaches learn solely on the basis of statistics extracted from text corpora and do not exploit knowledge from semantic networks. Additionally, their induced senses are neither readily interpretable (Panchenko et al., 2017) nor easily mappable to lexical resources, which limits their application. Recent approaches have utilized semantic networks to inject knowledge into existing word representations (Yu and Dredze, 2014; Faruqui et al., 2015; Goikoetxea et al., 2015; Speer and Lowry-Duda, 2017; Mrksic et al., 2017), but without solving the meaning conflation issue. In order to obtain a representation for each sense of a word, a number of approaches have leveraged lexical resources to learn sense embeddings as a result of post-processing conventional word embeddings (Chen et al., 2014; Johansson and Pina, 2015; Jauhar et al., 2015; Rothe and Schütze, 2015; Pilehvar and Collier, 2016; Camacho-Collados et al., 2016).

Instead, we propose SW2V (*Senses and Words to Vectors*), a neural model that exploits knowledge from both text corpora and semantic networks in order to simultaneously learn embeddings for both words and senses. Moreover, our model provides three additional key features: (1) both word and sense embeddings are represented in the same vector space, (2) it is flexible, as it can be applied to different predictive models, and (3) it is scalable for very large semantic networks and text corpora.

Authors marked with an asterisk (*) contributed equally.

2 Related work

Embedding words from large corpora into a low-dimensional vector space has been a popular task since the appearance of the probabilistic feed-forward neural network language model (Bengio et al., 2003) and later developments such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). However, little research has focused on exploiting lexical resources to overcome the inherent ambiguity of word embeddings.

Iacobacci et al. (2015) overcame this limitation by applying an off-the-shelf disambiguation system (i.e. Babelfy (Moro et al., 2014)) to a corpus and then using word2vec to learn sense embeddings over the pre-disambiguated text. However, in their approach words are replaced by their intended senses, consequently producing as output sense representations only. The representation of words and senses in the same vector space proves essential for applying these knowledge-based sense embeddings in downstream applications, particularly for their integration into neural architectures (Pilehvar et al., 2017). In the literature, various different methods have attempted to overcome this limitation. Chen et al. (2014) proposed a model for obtaining both word and sense representations based on a first training step of conventional word embeddings, a second disambiguation step based on sense definitions, and a final training phase which uses the disambiguated text as input. Likewise, Rothe and Schütze (2015) aimed at building a shared space of word and sense embeddings based on two steps: a first training step of only word embeddings and a second training step to produce sense and synset embeddings. These two approaches require multiple steps of training and make use of a relatively small resource like WordNet, which limits their coverage and applicability. Camacho-Collados et al. (2016) increased the coverage of these WordNet-based approaches by exploiting the complementary knowledge of WordNet and Wikipedia along with pre-trained word embeddings. Finally, Wang et al. (2014) and Fang et al. (2016) proposed a model to align vector spaces of words and entities from knowledge bases. However, these approaches are restricted to nominal instances only (i.e. Wikipedia pages or entities).

In contrast, we propose a model which learns both words and sense embeddings from a single joint training phase, producing a common vector space of words and senses as an emerging feature.

3 Connecting words and senses in context

In order to jointly produce embeddings for words and senses, SW2V needs as input a corpus where words are connected to senses[1] in each given context. One option for obtaining such connections could be to take a sense-annotated corpus as input. However, manually annotating large amounts of data is extremely expensive and therefore impractical in normal settings. Obtaining sense-annotated data from current off-the-shelf disambiguation and entity linking systems is possible, but generally suffers from two major problems. First, supervised systems are hampered by the very same problem of needing large amounts of sense-annotated data. Second, the relatively slow speed of current disambiguation systems, such as graph-based approaches (Hoffart et al., 2012; Agirre et al., 2014; Moro et al., 2014), or word-expert supervised systems (Zhong and Ng, 2010; Iacobacci et al., 2016; Melamud et al., 2016), could become an obstacle when applied to large corpora.

This is the reason why we propose a simple yet effective unsupervised *shallow word-sense connectivity* algorithm, which can be applied to virtually any given semantic network and is linear on the corpus size. The main idea of the algorithm is to exploit the connections of a semantic network by associating words with the senses that are most connected within the sentence, according to the underlying network.

Shallow word-sense connectivity algorithm. Formally, a corpus and a semantic network are taken as input and a set of connected words and senses is produced as output. We define a semantic network as a graph (S, E) where the set S contains synsets (nodes) and E represents a set of semantically connected synset pairs (edges). Algorithm 1 describes how to connect words and senses in a given text (sentence or paragraph) T. First, we gather in a set S_T all candidate synsets of the words (including multiwords up to trigrams) in T (lines 1 to 3). Second, for each candidate synset s we calculate the number of synsets which are connected with s in the semantic network and are included in S_T, excluding connections of synsets which only appear as candidates of the

[1] In this paper we focus on senses but other items connected to words may be used (e.g. supersenses or images).

Algorithm 1 Shallow word-sense connectivity

Input: Semantic network (S, E) and text T represented as a bag of words
Output: Set of connected words and senses $T^* \subset T \times S$

1: Set of synsets $S_T \leftarrow \emptyset$
2: **for each** word $w \in T$
3: $S_T \leftarrow S_T \cup S_w$ (S_w: set of candidate synsets of w)
4: Minimum connections threshold $\theta \leftarrow \frac{|S_T|+|T|}{2\,\delta}$
5: Output set of connections $T^* \leftarrow \emptyset$
6: **for each** $w \in T$
7: Relative maximum connections $max = 0$
8: Set of senses associated with w, $C_w \leftarrow \emptyset$
9: **for each** candidate synset $s \in S_w$
10: Number of edges $n = |s' \in S_T : (s, s') \in E \;\; \& \;\; \exists w' \in T : w' \neq w \;\; \& \;\; s' \in S_{w'}|$
11: **if** $n \geq max \;\; \& \;\; n \geq \theta$ **then**
12: **if** $n > max$ **then**
13: $C_w \leftarrow \{(w, s)\}$
14: $max \leftarrow n$
15: **else**
16: $C_w \leftarrow C_w \cup \{(w, s)\}$
17: $T^* \leftarrow T^* \cup C_w$
18: **return** Output set of connected words and senses T^*

same word (lines 5 to 10). Finally, each word is associated with its top candidate synset(s) according to its/their number of connections in context, provided that its/their number of connections exceeds a threshold $\theta = \frac{|S_T|+|T|}{2\,\delta}$ (lines 11 to 17).[2] This parameter aims to retain relevant connectivity across senses, as only senses above the threshold will be connected to words in the output corpus. θ is proportional to the reciprocal of a parameter δ,[3] and directly proportional to the average text length and number of candidate synsets within the text.

The complexity of the proposed algorithm is $N + (N \times \alpha)$, where N is the number of words of the training corpus and α is the average polysemy degree of a word in the corpus according to the input semantic network. Considering that non-content words are not taken into account (i.e. polysemy degree 0) and that the average polysemy degree of words in current lexical resources (e.g. WordNet or BabelNet) does not exceed a small constant (3) in any language, we can safely assume that the algorithm is linear in the size of the training corpus. Hence, the training time is not significantly increased in comparison to training words only, irrespective of the corpus size. This enables a fast training on large amounts of text corpora, in contrast to current unsupervised disambiguation algorithms. Additionally, as we will show in Section 5.2, this algorithm does not only speed up significantly the training phase, but also leads to more accurate results.

Note that with our algorithm a word is allowed to have more than one sense associated. In fact, current lexical resources like WordNet (Miller, 1995) or BabelNet (Navigli and Ponzetto, 2012) are hampered by the high granularity of their sense inventories (Hovy et al., 2013). In Section 6.2 we show how our sense embeddings are particularly suited to deal with this issue.

4 Joint training of words and senses

The goal of our approach is to obtain a shared vector space of words and senses. To this end, our model extends conventional word embedding models by integrating explicit knowledge into its architecture. While we will focus on the Continuous Bag Of Words (CBOW) architecture of word2vec (Mikolov et al., 2013), our extension can easily be applied similarly to Skip-Gram, or to other predictive approaches based on neural networks. The CBOW architecture is based on the feedforward neural network language model (Bengio et al., 2003) and aims at predicting the current word using its surrounding context. The architecture consists of input, hidden and output layers. The input layer has the size of the word vocabulary and encodes the context as a combination of one-hot vector representations of surrounding words of a given target word. The output layer has the same size as the input layer and contains a one-hot vector of the target word during the training phase.

Our model extends the input and output layers of the neural network with word senses[4] by exploiting the intrinsic relationship between words and senses. The leading principle is that, since a word is the surface form of an underlying sense, updating the embedding of the word should produce a consequent update to the embedding representing that particular sense, and vice-versa. As a consequence of the algorithm described in the previous section, each word in the corpus may be connected with zero, one or more senses. We re-

[2] As mentioned above, all unigrams, bigrams and trigrams present in the semantic network are considered. In the case of overlapping instances, the selection of the final instance is performed in this order: mention whose synset is more connected (i.e. n is higher), longer mention and from left to right.

[3] Higher values of δ lead to higher recall, while lower values of δ increase precision but lower the recall. We set the value of δ to 100, as it was shown to produce a fine balance between precision and recall. This parameter may also be tuned on downstream tasks.

[4] Our model can also produce a space of words and synset embeddings as output: the only difference is that all synonym senses would be considered to be the same item, i.e. a synset.

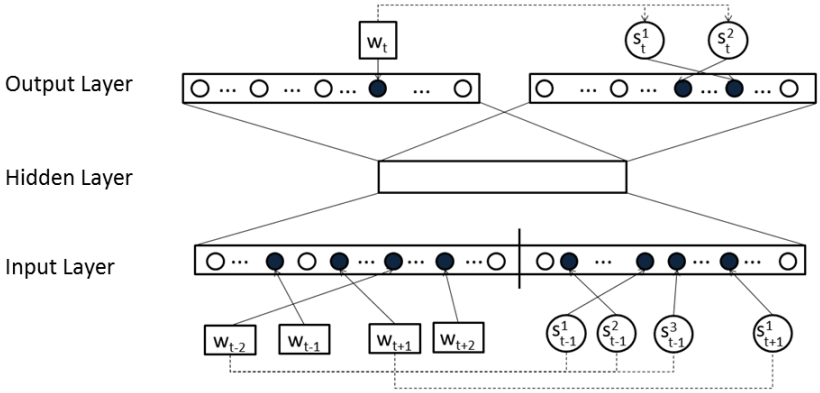

Figure 1: The SW2V architecture on a sample training instance using four context words. Dotted lines represent the virtual link between words and associated senses in context. In this example, the input layer consists of a context of two previous words (w_{t-2}, w_{t-1}) and two subsequent words (w_{t+1}, w_{t+2}) with respect to the target word w_t. Two words (w_{t-1}, w_{t+2}) do not have senses associated in context, while w_{t-2}, w_{t+1} have three senses (s^1_{t-1}, s^2_{t-1}, s^3_{t-1}) and one sense associated (s^1_{t+1}) in context, respectively. The output layer consists of the target word w_t, which has two senses associated (s^1_t, s^2_t) in context.

fer to the set of senses connected to a given word within the specific context as its *associated senses*.

Formally, we define a training instance as a sequence of words $W = w_{t-n}, ..., w_t, ..., w_{t+n}$ (being w_t the target word) and $S = S_{t-n}, ..., S_t, ..., S_{t+n}$, where $S_i = s^1_i, ..., s^{k_i}_i$ is the sequence of all associated senses in context of $w_i \in W$. Note that S_i might be empty if the word w_i does not have any associated sense. In our model each target word takes as context both its surrounding words and all the senses associated with them. In contrast to the original CBOW architecture, where the training criterion is to correctly classify w_t, our approach aims to predict the word w_t and its set S_t of associated senses. This is equivalent to minimizing the following loss function:

$$E = -\log(p(w_t|W^t, S^t)) - \sum_{s \in S_t} \log(p(s|W^t, S^t))$$

where $W^t = w_{t-n}, ..., w_{t-1}, w_{t+1}, ..., w_{t+n}$ and $S^t = S_{t-n}, ..., S_{t-1}, S_{t+1}, ..., S_{t+n}$. Figure 1 shows the organization of the input and the output layers on a sample training instance. In what follows we present a set of variants of the model on the output and the input layers.

4.1 Output layer alternatives

Both words and senses. This is the default case explained above. If a word has one or more associated senses, these senses are also used as target on a separate output layer.

Only words. In this case we exclude senses as target. There is a single output layer with the size of the word vocabulary as in the original CBOW model.

Only senses. In contrast, this alternative excludes words, using only senses as target. In this case, if a word does not have any associated sense, it is not used as target instance.

4.2 Input layer alternatives

Both words and senses. Words and their associated senses are included in the input layer and contribute to the hidden state. Both words and senses are updated as a consequence of the backpropagation algorithm.

Only words. In this alternative only the surrounding words contribute to the hidden state, i.e. the target word/sense (depending on the alternative of the output layer) is predicted only from word features. The update of an input word is propagated to the embeddings of its associated senses, if any. In other words, despite not being included in the input layer, senses still receive the same gradient of the associated input word, through a virtual connection. This configuration, coupled with the only-words output layer configuration, corresponds exactly to the default CBOW architecture of word2vec with the only addition of the update step for senses.

Only senses. Words are excluded from the input layer and the target is predicted only from the senses associated with the surrounding words. The weights of the words are updated through the updates of the associated senses, in contrast to the only-words alternative.

5 Analysis of Model Components

In this section we analyze the different components of SW2V, including the nine model configurations (Section 5.1) and the algorithm which generates the connections between words and senses in context (Section 5.2). In what follows we describe the common analysis setting:

- **Training model and hyperparameters.** For evaluation purposes, we use the CBOW model of word2vec with standard hyperparameters: the dimensionality of the vectors is set to 300 and the window size to 8, and hierarchical softmax is used for normalization. These hyperparameter values are set across all experiments.

- **Corpus and semantic network.** We use a 300M-words corpus from the UMBC project (Han et al., 2013), which contains English paragraphs extracted from the web.[5] As semantic network we use BabelNet 3.0[6], a large multilingual semantic network with over 350 million semantic connections, integrating resources such as Wikipedia and WordNet. We chose BabelNet owing to its wide coverage of named entities and lexicographic knowledge.

- **Benchmark.** Word similarity has been one of the most popular benchmarks for *in-vitro* evaluation of vector space models (Pennington et al., 2014; Levy et al., 2015). For the analysis we use two word similarity datasets: the similarity portion (Agirre et al., 2009, WS-Sim) of the WordSim-353 dataset (Finkelstein et al., 2002) and RG-65 (Rubenstein and Goodenough, 1965). In order to compute the similarity of two words using our sense embeddings, we apply the standard closest senses strategy (Resnik, 1995; Budanitsky and Hirst, 2006; Camacho-Collados et al., 2015), using cosine similarity (cos) as comparison measure between senses:

$$sim(w_1, w_2) = \max_{s \in S_{w_1}, s' \in S_{w_2}} \cos(\vec{s}_1, \vec{s}_2) \quad (1)$$

where S_{w_i} represents the set of all candidate senses of w_i and $\vec{s}_i$ refers to the sense vector representation of the sense s_i.

5.1 Model configurations

In this section we analyze the different configurations of our model in respect of the input and the output layer on a word similarity experiment. Recall from Section 4 that our model could have words, senses or both in either the input and output layers. Table 1 shows the results of all nine configurations on the WS-Sim and RG-65 datasets.

As shown in Table 1, the best configuration according to both Spearman and Pearson correlation measures is the configuration which has only senses in the input layer and both words and senses in the output layer.[7] In fact, taking only senses as input seems to be consistently the best alternative for the input layer. Our hunch is that the knowledge learned from both the co-occurrence information and the semantic network is more balanced with this input setting. For instance, in the case of including both words and senses in the input layer, the co-occurrence information learned by the network would be duplicated for both words and senses.

5.2 Disambiguation / Shallow word-sense connectivity algorithm

In this section we evaluate the impact of our *shallow word-sense connectivity algorithm* (Section 3) by testing our model directly taking a pre-disambiguated text as input. In this case the network exploits the connections between each word and its disambiguated sense in context. For this comparison we used Babelfy[8] (Moro et al., 2014), a state-of-the-art graph-based disambiguation and entity linking system based on BabelNet. We compare to both the default Babelfy system which

[5] http://ebiquity.umbc.edu/blogger/2013/05/01/umbc-webbase-corpus-of-3b-english-words/

[6] http://babelnet.org

[7] In this analysis we used the word similarity task for optimizing the sense embeddings, without caring about the performance of word embeddings or their interconnectivity. Therefore, this configuration may not be optimal for word embeddings and may be further tuned on specific applications. More information about different configurations in the documentation of the source code.

[8] http://babelfy.org

		Output											
		Words				Senses				Both			
		WS-Sim		RG-65		WS-Sim		RG-65		WS-Sim		RG-65	
		r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
Input	**Words**	0.49	0.48	0.65	0.66	0.56	0.56	0.67	0.67	0.54	0.53	0.66	0.65
	Senses	0.69	0.69	0.70	0.71	0.69	0.70	0.70	**0.74**	**0.72**	**0.71**	**0.71**	**0.74**
	Both	0.60	0.65	0.67	0.70	0.62	0.65	0.66	0.67	0.65	**0.71**	0.68	0.70

Table 1: Pearson (r) and Spearman (ρ) correlation performance of the nine configurations of SW2V

	WS-Sim		RG-65	
	r	ρ	r	ρ
Shallow	**0.72**	**0.71**	**0.71**	**0.74**
Babelfy	0.65	0.63	0.69	0.70
Babelfy*	0.63	0.61	0.65	0.64

Table 2: Pearson (r) and Spearman (ρ) correlation performance of SW2V integrating our *shallow* word-sense connectivity algorithm (default), Babelfy, or Babelfy*.

uses the *Most Common Sense* (MCS) heuristic as a back-off strategy and, following (Iacobacci et al., 2015), we also include a version in which only instances above the Babelfy default confidence threshold are disambiguated (i.e. the MCS back-off strategy is disabled). We will refer to this latter version as Babelfy* and report the best configuration of each strategy according to our analysis.

Table 2 shows the results of our model using the three different strategies on RG-65 and WS-Sim. Our shallow word-sense connectivity algorithm achieves the best overall results. We believe that these results are due to the semantic connectivity ensured by our algorithm and to the possibility of associating words with more than one sense, which seems beneficial for training, making it more robust to possible disambiguation errors and to the sense granularity issue (Erk et al., 2013). The results are especially significant considering that our algorithm took a tenth of the time needed by Babelfy to process the corpus.

6 Evaluation

We perform a qualitative and quantitative evaluation of important features of SW2V in three different tasks. First, in order to compare our model against standard word-based approaches, we evaluate our system in the word similarity task (Section 6.1). Second, we measure the quality of our sense embeddings in a sense-specific application: sense clustering (Section 6.2). Finally, we evaluate the coherence of our unified vector space by measuring the interconnectivity of word and sense embeddings (Section 6.3).

Experimental setting. Throughout all the experiments we use the same standard hyperparameters mentioned in Section 5 for both the original word2vec implementation and our proposed model SW2V. For SW2V we use the same optimal configuration according to the analysis of the previous section (only senses as input, and both words and senses as output) for all tasks. As training corpus we take the full 3B-words UMBC webbase corpus and the Wikipedia (Wikipedia dump of November 2014), used by three of the comparison systems. We use BabelNet 3.0 ($SW2V_{BN}$) and WordNet 3.0 ($SW2V_{WN}$) as semantic networks.

Comparison systems. We compare with the publicly available pre-trained sense embeddings of four state-of-the-art models: Chen et al. (2014)[9] and AutoExtend[10] (Rothe and Schütze, 2015) based on WordNet, and SensEmbed[11] (Iacobacci et al., 2015) and NASARI[12] (Camacho-Collados et al., 2016) based on BabelNet.

6.1 Word Similarity

In this section we evaluate our sense representations on the standard SimLex-999 (Hill et al., 2015) and MEN (Bruni et al., 2014) word similarity datasets[13]. SimLex and MEN contain 999 and 3000 word pairs, respectively, which constitute, to our knowledge, the two largest similar-

[9] http://pan.baidu.com/s/1eQcPK8i

[10] We used the AutoExtend code (http://cistern.cis.lmu.de/~sascha/AutoExtend/) to obtain sense vectors using W2V embeddings trained on UMBC (GoogleNews corpus used in their pre-trained models is not publicly available). We also tried the code to include BabelNet as lexical resource, but it was not easily scalable (BabelNet is two orders of magnitude larger than WordNet).

[11] http://lcl.uniroma1.it/sensembed/

[12] http://lcl.uniroma1.it/nasari/

[13] To enable a fair comparison we did not perform experiments on the small datasets used in Section 5 for validation.

	System	Corpus	SimLex-999		MEN	
			r	ρ	r	ρ
Senses	$SW2V_{BN}$	UMBC	**0.49**	**0.47**	0.75	0.75
	$SW2V_{WN}$	UMBC	0.46	0.45	**0.76**	**0.76**
	AutoExtend	UMBC	0.47	0.45	0.74	0.75
	AutoExtend	Google-News	0.46	0.46	0.68	0.70
	$SW2V_{BN}$	Wikipedia	0.47	0.43	0.71	0.73
	$SW2V_{WN}$	Wikipedia	0.47	0.43	0.71	0.72
	SensEmbed	Wikipedia	0.43	0.39	0.65	0.70
	Chen et al. (2014)	Wikipedia	0.46	0.43	0.62	0.62
Words	Word2vec	UMBC	0.39	0.39	0.75	0.75
	$Retrofitting_{BN}$	UMBC	0.47	0.46	0.75	**0.76**
	$Retrofitting_{WN}$	UMBC	0.47	0.46	**0.76**	**0.76**
	Word2vec	Wikipedia	0.39	0.38	0.71	0.72
	$Retrofitting_{BN}$	Wikipedia	0.35	0.32	0.66	0.66
	$Retrofitting_{WN}$	Wikipedia	0.47	0.44	0.73	0.73

Table 3: Pearson (r) and Spearman (ρ) correlation performance on the SimLex-999 and MEN word similarity datasets.

ity datasets comprising a balanced set of noun, verb and adjective instances. As explained in Section 5, we use the closest sense strategy for the word similarity measurement of our model and all sense-based comparison systems. As regards the word embedding models, words are directly compared by using cosine similarity. We also include a *retrofitted* version of the original word2vec word vectors (Faruqui et al., 2015, Retrofitting[14]) using WordNet ($Retrofitting_{WN}$) and BabelNet ($Retrofitting_{BN}$) as lexical resources.

Table 3 shows the results of SW2V and all comparison models in SimLex and MEN. SW2V consistently outperforms all sense-based comparison systems using the same corpus, and clearly performs better than the original word2vec trained on the same corpus. Retrofitting decreases the performance of the original word2vec on the Wikipedia corpus using BabelNet as lexical resource, but significantly improves the original word vectors on the UMBC corpus, obtaining comparable results to our approach. However, while our approach provides a shared space of words and senses, Retrofitting still conflates different meanings of a word into the same vector.

Additionally, we noticed that most of the score divergences between our system and the gold standard scores in SimLex-999 were produced on antonym pairs, which are over-represented in this dataset: 38 word pairs hold a clear antonymy relation (e.g. *encourage-discourage* or *long-short*), while 41 additional pairs hold some degree of antonymy (e.g. *new-ancient* or *man-woman*).[15] In contrast to the consistently low gold similarity scores given to antonym pairs, our system varies its similarity scores depending on the specific nature of the pair[16]. Recent works have managed to obtain significant improvements by tweaking usual word embedding approaches into providing low similarity scores for antonym pairs (Pham et al., 2015; Schwartz et al., 2015; Nguyen et al., 2016; Mrksic et al., 2017), but this is outside the scope of this paper.

6.2 Sense Clustering

Current lexical resources tend to suffer from the high granularity of their sense inventories (Palmer et al., 2007). In fact, a meaningful clustering of their senses may lead to improvements on downstream tasks (Hovy et al., 2013; Flekova and Gurevych, 2016; Pilehvar et al., 2017). In this section we evaluate our synset representations on the Wikipedia sense clustering task. For a fair comparison with respect to the BabelNet-based com-

[14] https://github.com/mfaruqui/retrofitting

[15] Two annotators decided the degree of antonymy between word pairs: *clear antonyms*, *weak antonyms* or *neither*.

[16] For instance, the pairs *sunset-sunrise* and *day-night* are given, respectively, 1.88 and 2.47 gold scores in the 0-10 scale, while our model gives them a higher similarity score. In fact, both pairs appear as coordinate synsets in WordNet.

	Accuracy	F-Measure
SW2V	**87.8**	**63.9**
SensEmbed	82.7	40.3
NASARI	87.0	62.5
Multi-SVM	85.5	-
Mono-SVM	83.5	-
Baseline	17.5	29.8

Table 4: Accuracy and F-Measure percentages of different systems on the SemEval Wikipedia sense clustering dataset.

parison systems that use the Wikipedia corpus for training, in this experiment we report the results of our model trained on the Wikipedia corpus and using BabelNet as lexical resource only. For the evaluation we consider the two Wikipedia sense clustering datasets (500-pair and SemEval) created by Dandala et al. (2013). In these datasets sense clustering is viewed as a binary classification task in which, given a pair of Wikipedia pages, the system has to decide whether to cluster them into a single instance or not. To this end, we use our synset embeddings and cluster Wikipedia pages[17] together if their similarity exceeds a threshold γ. In order to set the optimal value of γ, we follow Dandala et al. (2013) and use the first 500-pairs sense clustering dataset for tuning. We set the threshold γ to 0.35, which is the value leading to the highest F-Measure among all values from 0 to 1 with a 0.05 step size on the 500-pair dataset. Likewise, we set a threshold for NASARI (0.7) and SensEmbed (0.3) comparison systems.

Finally, we evaluate our approach on the SemEval sense clustering test set. This test set consists of 925 pairs which were obtained from a set of highly ambiguous words gathered from past SemEval tasks. For comparison, we also include the supervised approach of Dandala et al. (2013) based on a multi-feature Support Vector Machine classifier trained on an automatically-labeled dataset of the English Wikipedia (Mono-SVM) and Wikipedia in four different languages (Multi-SVM). As naive baseline we include the system which would cluster all given pairs.

Table 4 shows the F-Measure and accuracy results on the SemEval sense clustering dataset. SW2V outperforms all comparison systems according to both measures, including the sense representations of NASARI and SensEmbed using the same setup and the same underlying lexical resource. This confirms the capability of our system to accurately capture the semantics of word senses on this sense-specific task.

[17] Since Wikipedia is a resource included in BabelNet, our synset representations are expandable to Wikipedia pages.

6.3 Word and sense interconnectivity

In the previous experiments we evaluated the effectiveness of the sense embeddings. In contrast, this experiment aims at testing the interconnectivity between word and sense embeddings in the vector space. As explained in Section 2, there have been previous approaches building a shared space of word and sense embeddings, but to date little research has focused on testing the semantic coherence of the vector space. To this end, we evaluate our model on a Word Sense Disambiguation (WSD) task, using our shared vector space of words and senses to obtain a *Most Common Sense* (MCS) baseline. The insight behind this experiment is that a semantically coherent shared space of words and senses should be able to build a relatively strong baseline for the task, as the MCS of a given word should be closer to the word vector than any other sense. The MCS baseline is generally integrated into the pipeline of state-of-the-art WSD and Entity Linking systems as a back-off strategy (Navigli, 2009; Jin et al., 2009; Zhong and Ng, 2010; Moro et al., 2014; Raganato et al., 2017) and is used in various NLP applications (Bennett et al., 2016). Therefore, a system which automatically identifies the MCS of words from non-annotated text may be quite valuable, especially for resource-poor languages or large knowledge resources for which obtaining sense-annotated corpora is extremely expensive. Moreover, even in a resource like WordNet for which sense-annotated data is available (Miller et al., 1993, SemCor), 61% of its polysemous lemmas have no sense annotations (Bennett et al., 2016).

Given an input word w, we compute the cosine similarity between w and all its candidate senses, picking the sense leading to the highest similarity:

$$MCS(w) = \underset{s \in S_w}{\operatorname{argmax}} \cos(\vec{w}, \vec{s}) \qquad (2)$$

where $\cos(\vec{w}, \vec{s})$ refers to the cosine similarity between the embeddings of w and s. In order to assess the reliability of SW2V against previous models using WordNet as sense inventory, we test our model on the all-words SemEval-2007 (task 17) (Pradhan et al., 2007) and SemEval-2013 (task

	SemEval-07	SemEval-13
SW2V	**39.9**	**54.0**
AutoExtend	17.6	31.0
Baseline	24.8	34.9

Table 5: F-Measure percentage of different MCS strategies on the SemEval-2007 and SemEval-2013 WSD datasets.

12) (Navigli et al., 2013) WSD datasets. Note that our model using BabelNet as semantic network has a far larger coverage than just WordNet and may additionally be used for Wikification (Mihalcea and Csomai, 2007) and Entity Linking tasks. Since the versions of WordNet vary across datasets and comparison systems, we decided to evaluate the systems on the portion of the datasets covered by all comparison systems[18] (less than 10% of instances were removed from each dataset).

Table 5 shows the results of our system and AutoExtend on the SemEval-2007 and SemEval-2013 WSD datasets. SW2V provides the best MCS results in both datasets. In general, AutoExtend does not accurately capture the predominant sense of a word and performs worse than a baseline that selects the intended sense randomly from the set of all possible senses of the target word.

In fact, AutoExtend tends to create clusters which include a word and all its possible senses. As an example, Table 6 shows the closest word and sense[19] embeddings of our SW2V model and AutoExtend to the *military* and *fish* senses of, respectively, *company* and *school*. AutoExtend creates clusters with all the senses of *company* and *school* and their related instances, even if they belong to different domains (e.g., firm^2_n or business^1_n clearly concern the *business* sense of *company)*. Instead, SW2V creates a semantic cluster of word and sense embeddings which are semantically close to the corresponding $company^2_n$ and $school^7_n$ senses.

7 Conclusion and Future Work

In this paper we proposed SW2V (*Senses and Words to Vectors*), a neural model which learns vector representations for words and senses in a joint training phase by exploiting both text corpora and knowledge from semantic networks. Data (including the preprocessed corpora and pre-trained embeddings used in the evaluation) and source code to apply our extension of the word2vec architecture to learn word and sense embeddings from any preprocessed corpus are freely available at `http://lcl.uniroma1.it/sw2v`. Unlike previous sense-based models which require post-processing steps and use WordNet as sense inventory, our model achieves a semantically coherent vector space of both words and senses as an emerging feature of a single training phase and is easily scalable to larger semantic networks like BabelNet. Finally, we showed, both quantitatively and qualitatively, some of the advantages of using our approach as against previous state-of-the-art word- and sense-based models in various tasks, and highlighted interesting semantic properties of the resulting unified vector space of word and sense embeddings.

As future work we plan to integrate a WSD and Entity Linking system for applying our model on downstream NLP applications, along the lines of Pilehvar et al. (2017). We are also planning to apply our model to languages other than English and to study its potential on multilingual and cross-lingual applications.

$company^2_n$ (military unit)		*$school^7_n$ (group of fish)*	
AutoExtend	**SW2V**	**AutoExtend**	**SW2V**
$company^9_n$	$battalion^1_n$	school	$schools^7_n$
company	battalion	$school^4_n$	$sharks^1_n$
$company^8_n$	$regiment^1_n$	$school^6_n$	sharks
$company^6_n$	$detachment^4_n$	$school^1_v$	$shoals^3_n$
$company^7_n$	$platoon^1_n$	$school^3_n$	$fish^1_n$
$company^1_v$	$brigade^1_n$	elementary	$dolphins^1_n$
firm	regiment	schools	$pods^3_n$
$business^1_n$	$corps^1_n$	$elementary^3_a$	eels
$firm^2_n$	brigade	$school^5_n$	dolphins
$company^1_n$	platoon	$elementary^1_a$	$whales^2_n$

Table 6: Ten closest word and sense embeddings to the senses $company^2_n$ (military unit) and $school^7_n$ (group of fish).

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487.

Jose Camacho-Collados is supported by a Google Doctoral Fellowship in Natural Language Processing. We would also like to thank Jim McManus for his comments on the manuscript.

[18] We were unable to obtain the word embeddings of Chen et al. (2014) for comparison even after contacting the authors.

[19] Following Navigli (2009), $word^p_n$ is the n^{th} sense of *word* with part of speech p (using WordNet 3.0).

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 NAACL*. pages 19–27.

Eneko Agirre, Oier Lopez de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* 40(1):57–84.

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

Andrew Bennett, Timothy Baldwin, Jey Han Lau, Diana McCarthy, and Francis Bond. 2016. Lexsemtm: A semantic dataset based on all-words unsupervised sense distribution learning. In *Proceedings of ACL*. pages 1513–1524.

Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In *Proceedings of EMNLP*. pages 615–620.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)* 49(1-47).

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

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. A Unified Multilingual Semantic Representation of Concepts. In *Proceedings of ACL*. Beijing, China, pages 741–751.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence* 240:36–64.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of EMNLP*. Doha, Qatar, pages 1025–1035.

Bharath Dandala, Chris Hokamp, Rada Mihalcea, and Razvan C. Bunescu. 2013. Sense clustering using Wikipedia. In *Proc. of RANLP*. Hissar, Bulgaria, pages 164–171.

Antonio Di Marco and Roberto Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics* 39(3):709–754.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. 2013. Measuring word meaning in context. *Computational Linguistics* 39(3):511–554.

Allyson Ettinger, Philip Resnik, and Marine Carpuat. 2016. Retrofitting Sense-Specific Word Vectors Using Parallel Text. In *Proceedings of NAACL-HLT*. pages 1378–1383.

Wei Fang, Jianwen Zhang, Dilin Wang, Zheng Chen, and Ming Li. 2016. Entity disambiguation by knowledge and text jointly embedding. In *Proceedings of CoNLL*. pages 260–269.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*. pages 1606–1615.

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.

Lucie Flekova and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of ACL*. pages 2029–2041.

Josu Goikoetxea, Aitor Soroa, Eneko Agirre, and Basque Country Donostia. 2015. Random walks and neural network language models on knowledge bases. In *Proceedings of NAACL*. pages 1434–1439.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning sense-specific word embeddings by exploiting bilingual resources. In *Proceedings of COLING*. pages 497–507.

Lushan Han, Abhay Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC EBIQUITY-CORE: Semantic textual similarity systems. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*. volume 1, pages 44–52.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics* .

Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2012. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of CIKM*. pages 545–554.

Eduard H. Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence* 194:2–27.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proc. of ACL*. Jeju Island, Korea, pages 873–882.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of ACL*. Beijing, China, pages 95–105.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *Proceedings of ACL*. pages 897–907.

Sujay Kumar Jauhar, Chris Dyer, and Eduard Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *Proceedings of NAACL*.

Peng Jin, Diana McCarthy, Rob Koeling, and John Carroll. 2009. Estimating and exploiting the entropy of sense distributions. In *Proceedings of NAACL (2)*. pages 233–236.

Richard Johansson and Luis Nieto Pina. 2015. Embedding a semantic network in a word space. In *Proceedings of NAACL*. pages 1428–1433.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *TACL* 3:211–225.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of EMNLP*. Lisbon, Portugal.

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proc. of CONLL*. pages 51–61.

Rada Mihalcea and Andras Csomai. 2007. Wikify! Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge management*. Lisbon, Portugal, pages 233–242.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A semantic concordance. In *Proceedings of the 3rd DARPA Workshop on Human Language Technology*. Plainsboro, N.J., pages 303–308.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL* 2:231–244.

Nikola Mrksic, Ivan Vulić, Diarmuid Ó Séaghdha, Ira Leviant, Roi Reichart, Milica Gai, Anna Korhonen, and Steve Young. 2017. Semantic Specialisation of Distributional Word Vector Spaces using Monolingual and Cross-Lingual Constraints. *TACL* .

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

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Proceedings of SemEval 2013*. pages 222–231.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *AIJ* 193:217–250.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of EMNLP*. Doha, Qatar, pages 1059–1069.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of ACL*. pages 454–459.

Martha Palmer, Hoa Dang, and Christiane Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Natural Language Engineering* 13(2):137–163.

Alexander Panchenko, Eugen Ruppert, Stefano Faralli, Simone Paolo Ponzetto, and Chris Biemann. 2017. Unsupervised does not mean uninterpretable: The case for word sense induction and disambiguation. In *Proceedings of EACL*. pages 86–98.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of EMNLP*. pages 1532–1543.

Nghia The Pham, Angeliki Lazaridou, and Marco Baroni. 2015. A multitask objective to inject lexical contrast into distributional semantics. In *Proceedings of ACL*. pages 21–26.

Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a Seamless Integration of Word Senses into Downstream NLP Applications. In *Proceedings of ACL*. Vancouver, Canada.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of EMNLP*. Austin, TX.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of SemEval*. pages 87–92.

Lin Qiu, Kewei Tu, and Yong Yu. 2016. Context-dependent sense embedding. In *Proceedings of EMNLP*. Austin, Texas, pages 183–191.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proceedings of EACL*. pages 99–110.

Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of ACL*. pages 109–117.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI*. pages 448–453.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending Word Embeddings to Embeddings for Synsets and Lexemes. In *Proceedings of ACL*. Beijing, China, pages 1793–1803.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Commun. ACM* 8(10):627–633.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational linguistics* 24(1):97–123.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2015. Symmetric pattern based word embeddings for improved word similarity prediction. In *Proceedings of CoNLL*. pages 258–267.

Robert Speer and Joanna Lowry-Duda. 2017. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. pages 76–80.

Simon Šuster, Ivan Titov, and Gertjan van Noord. 2016. Bilingual learning of multi-sense embeddings with discrete autoencoders. In *Proceedings of NAACL-HLT*. pages 1346–1356.

Fei Tian, Hanjun Dai, Jiang Bian, Bin Gao, Rui Zhang, Enhong Chen, and Tie-Yan Liu. 2014. A probabilistic model for learning multi-prototype word embeddings. In *Proceedings of COLING*. pages 151–160.

Thuy Vu and D Stott Parker. 2016. K-embeddings: Learning conceptual embeddings for words using context. In *Proceedings of NAACL-HLT*. pages 1262–1267.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*. pages 1591–1601.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of ACL*. Beijing, China, pages 323–333.

Mo Yu and Mark Dredze. 2014. Improving lexical embeddings with semantic knowledge. In *Proceedings of ACL (2)*. pages 545–550.

Zhi Zhong and Hwee Tou Ng. 2010. It Makes Sense: A wide-coverage Word Sense Disambiguation system for free text. In *Proc. of ACL System Demonstrations*. pages 78–83.

Will Y. Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of EMNLP*. pages 1393–1398.