SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation

Bianca Scarlini, Tommaso Pasini, Roberto Navigli
Sapienza University of Rome
Department of Computer Science
{scarlini,pasini,navigli}@di.uniroma1.it

Abstract

Contextual representations of words derived by neural language models have proven to effectively encode the subtle distinctions that might occur between different meanings of the same word. However, these representations are not tied to a semantic network, hence they leave the word meanings implicit and thereby neglect the information that can be derived from the knowledge base itself. In this paper, we propose SensEmBERT, a knowledge-based approach that brings together the expressive power of language modelling and the vast amount of knowledge contained in a semantic network to produce high-quality latent semantic representations of word meanings in multiple languages. Our vectors lie in a space comparable with that of contextualized word embeddings, thus allowing a word occurrence to be easily linked to its meaning by applying a simple nearest neighbour approach.

We show that, whilst not relying on manual semantic annotations, SensEmBERT is able to either achieve or surpass state-of-the-art results attained by most of the supervised neural approaches on the English Word Sense Disambiguation task. When scaling to other languages, our representations prove to be equally effective as their English counterpart and outperform the existing state of the art on all the Word Sense Disambiguation multilingual datasets. The embeddings are released in five different languages at http://sensembert.org.

1 Introduction

Word Sense Disambiguation (WSD) is the task of associating the occurrence of a word in a text with its correct meaning from a predefined inventory of senses (Navigli 2009). Over the years, two distinct lines of research have been developed to tackle this problem: supervised and knowledge-based WSD. On the one hand, supervised models rely on semantically-annotated corpora for training (Raganato, Delli Bovi, and Navigli 2017; Kumar et al. 2019; Bevilacqua and Navigli 2019), while, on the other hand, knowledge-based systems employ graph-based algorithms on semantic networks to find the set of meanings that better disambiguate the input words (Moro, Raganato, and Navigli 2014; Agirre, de Lacalle, and Soroa 2014). Even though supervised approaches have proved to achieve better performance, they have difficulty scaling to different languages due to the paucity of multilingual sense-annotated data. Knowledge-based approaches, instead, are more flexible and can be applied to different languages, however at the cost of achieving lower performance than their supervised counterpart.

Recently, language models in different sauces, i.e., ELMo (Peters et al. 2018), BERT (Devlin et al. 2019), XLNET (Yang et al. 2019), etc., have attracted much interest as they have proved to be beneficial to several downstream tasks in NLP (Wang et al. 2018; 2019). In fact, the word representations provided by these models encode several pieces of linguistic information and, differently from static word embeddings (Mikolov et al. 2013; Pennington, Socher, and Manning 2014), they depend directly on the context a word is surrounded by. This has made these vectors especially interesting for the task of WSD, where effective contextual representations can be highly beneficial for solving lexical ambiguity. In fact, thanks to contextual embeddings, simple nearest neighbor algorithms have proved to be more effective and precise than complex supervised and knowledge-based approaches (Loureiro and Jorge 2019). These representations allowed sense-annotated corpora to be exploited in a more efficient way, as demonstrated by Loureiro and Jorge (2019). In fact, they closed the gap between the meanings in a semantic network and their occurrences in a text by producing concept embeddings that lie in a space that is comparable with that of contextual word embeddings. Nevertheless, this approach is still hampered by the need for manual semantic annotations in order to construct the concept vectors, which limits its range of action to texts in English only, as almost no manually-annotated data are available in other languages.

In this paper we present SensEmBERT, a knowledge-based approach for producing sense embeddings in multiple languages. We leverage the lexical-semantic information in a knowledge base, i.e., BabelNet, and an encyclopedic resource like Wikipedia, to relieve the burden of producing manually-tagged corpora. SensEmBERT, whilst not relying on annotated data, achieves state-of-the-art results on the multilingual WSD tasks and remains competitive with the best supervised models on English. Moreover, when providing supervision to our approach, our embeddings set a new state of the art on all the English WSD test sets for nouns.

2 Related Work

At the core of Natural Language Processing lies the problem of Word Sense Disambiguation (WSD), which addresses the ambiguity of words in a given context. WSD is usually tackled by exploiting two sources of knowledge: semantic networks and sense-annotated corpora. Semantic networks encode a more general knowledge that is not tied to a specific task and the information enclosed therein is usually employed for WSD by knowledge-based approaches. Sense-annotated corpora, instead, are tailored to the WSD task and are typically used as training sets for supervised systems.

Knowledge-based systems Knowledge-based approaches (Moro, Raganato, and Navigli 2014; Agirre, de Lacalle, and Soroa 2014) frame WSD as a graph-based problem, where the structure of a semantic network, such as WordNet (Fellbaum 1998) and BabelNet (Navigli and Ponzetto 2012), is used to find, for each input word, its correct meaning according to its context. WordNet is the most widespread lexical knowledge base, but it is limited to the English lexicon only, which restricts its applicability to other vocabularies. BabelNet copes with this problem by merging together lexical-semantic information in multiple languages coming from different resources, hence enabling knowledge-based approaches to scale over all the languages it supports. Despite their ability to scale over different languages, knowledge-based approaches fall behind supervised systems on English in terms of accuracy.

Supervised systems Supervised systems have attained state-of-the-art results across all English datasets by exploiting either SVM models (Iacobacci, Pilehvar, and Navigli 2016), or neural architectures (Melamud, Goldberger, and Dagan 2016; Raganato, Delli Bovi, and Navigli 2017; Vial, Lecouteux, and Schwab 2019). Nevertheless, they suffer from the *knowledge acquisition bottleneck*, which hampers the creation of large manually-curated corpora (Gale, Church, and Yarowsky 1992), and in turn hinders the ability of these approaches to scale over unseen words and new languages. To overcome the aforementioned shortcomings, coarser sense inventories (Lacerra et al. 2020) and automatic data augmentation approaches (Pasini and Navigli 2017; Pasini, Elia, and Navigli 2018; Scarlini, Pasini, and Navigli 2019) have been developed to cover more words, senses and languages. At the same time, dedicated architectures have been built to exploit the definitional information of a knowledge base (Luo et al. 2018; Kumar et al. 2019).

Recently, contextual representations of words (Peters et al. 2018; Devlin et al. 2019) have brought a breeze of change to WSD, where they have been employed for the creation of sense embeddings (Peters et al. 2018; Loureiro and Jorge 2019). These proved to be of high-quality inasmuch as they were able to surpass complex state-of-the-art models on the English WSD tasks when coupled with simple distance-based algorithms, i.e., k-NN. Nevertheless, these approaches rely on sense-annotated corpora to gather contextual information for each sense, and hence are limited to languages for which gold annotations are available, i.e., English.

In this paper, we present SensEmBERT which, in dispensing with the need for human-annotated corpora, unleashes the power of sense embeddings to virtually all BabelNet's languages. By leveraging the mapping between senses and Wikipedia pages, the relations among BabelNet synsets and the expressiveness of contextualized embeddings, we get rid of manual annotations while at the same time providing valuable contexts for the creation of our embeddings for all the nominal senses in BabelNet.

3 Preliminaries

SensEmBERT relies on different resources for building sense vectors: Wikipedia, a multilingual knowledge base, i.e., BabelNet (Navigli and Ponzetto 2012), the NASARI lexical vectors (Camacho-Collados, Pilehvar, and Navigli 2016) and a pre-trained language model for producing contextual representations, i.e., BERT (Devlin et al. 2019).

Wikipedia is the largest electronic encyclopedia freely available on the Web, including approximately 300 separate editions, each written in a different language. The information contained in Wikipedia is organized into articles, which are also referred to as Wikipedia pages. Each page aims at describing either abstract concepts, e.g., Freedom, or real world entities, e.g., Martin Luther King.

BabelNet[1] (Navigli and Ponzetto 2012) is a multilingual semantic network which comprises information coming from heterogeneous resources, such as WordNet, Wikipedia, etc. It is organized into synsets, i.e., sets of synonyms that express a single concept, which, in their turn, are connected to each other by different types of relation. We note that a synset clusters together several senses, each identified by one of the synset's lexicalizations. Moreover, thanks to cross-lingual mappings, the synsets in BabelNet conflate lexicalizations coming from distinct languages. For example, the terms *glass* (English), *verre* (French), *vaso* (Spanish), *bicchiere* (Italian), etc., are grouped together under the same synset expressing the *container* meaning of *glass*.

For our purposes we are especially interested in the following kinds of information contained in BabelNet:

- **Hypernym and hyponym edges**: each concept[2] is connected to other concepts by means of hypernym-hyponym relations. For example, the concept computer^1_n (*A machine for performing calculations automatically*)[3] is connected, inter alia, to the concept machine^1_n (*Any mechanical or electrical device*) via a hypernym relation (i.e., generalization), and to home_computer^1_n (*a computer intended for use in the home*) via a hyponym relation (specialization).

[1] https://babelnet.org

[2] We use synset and concept interchangeably for ease of reading.

[3] We use the notation of Navigli (2009), where l_p^k denotes the k-th meaning of the lemma l with pos p according to WordNet.

- **Semantically-related edges**: BabelNet also comprises edges expressing a general notion of relatedness between concepts. For example, computer^1_n is connected, among others, to mouse^4_n (*a hand-operated electronic device*) and to keyboard^1_n (*device consisting of a set of keys*).
- **Mappings to Wikipedia**: most of the concepts in the knowledge base are linked to one or more Wikipedia pages. For example, the concept for computer^1_n is linked to the Wikipedia page COMPUTER[4].

NASARI lexical vectors[5] (Camacho-Collados, Pilehvar, and Navigli 2016) provide explicit representations of BabelNet concepts by means of sparse lexical vectors. Each dimension is a word scored by its lexical specificity (Lafon 1980) with respect to the concept it is representing. The lexical specificity value is computed from the Wikipedia pages related to the target concept. We note that the words enclosed within each sense vector are those that most characterize it. For example, the vector for the *animal* sense of *mouse* includes, among others, the words *rat*, *rodent*, *animal*, *cat*, etc.

BERT[6] (Devlin et al. 2019) is a Transformer-based language model for learning contextual representations of words in a text. Recently, BERT ushered a new era for NLP. In fact, its contextual embeddings made it possible to achieve high performance in different NLP tasks, such as question answering and sentiment classification. In this work we take advantage of the BERT large and multilingual[7] models for English and the other languages, respectively.

4 SENSEMBERT

In this Section we present SENSEMBERT, a novel knowledge-based approach for creating latent representations of senses in multiple languages. It computes context-aware representations of BabelNet senses by combining the semantic and textual information that can be derived from multilingual resources, i.e., NASARI (Camacho-Collados, Pilehvar, and Navigli 2016) and BabelNet (Navigli and Ponzetto 2012), with the representational power of neural language models, i.e., BERT (Devlin et al. 2019). Our approach can be divided into the following three steps:

- **Context Retrieval**, which collects all the relevant textual information from Wikipedia for a given concept in the semantic network (Section 4.1).
- **Word Embedding**, which, given the contexts retrieved in the previous step, computes the vector representation of each relevant word of the target synset (Section 4.2).
- **Sense Embedding**, which merges the contextual information computed in the previous step and enriches it with additional knowledge from the semantic network, so as to build an embedding for the target sense (Section 4.3).

4.1 Context Retrieval

In this step we aim at retrieving all the contexts from Wikipedia that are suitable for characterizing a given word synset. To this end, similarly to Camacho-Collados, Pilehvar, and Navigli (2016), we exploit the mappings between synsets and Wikipedia pages available in BabelNet, as well as its taxonomic structure, to collect textual information that is relevant to a target synset s.

First, starting from a synset s, we collect the set of its most related concepts, i.e., all the synsets that are connected to s through either a hypernym, hyponym or semantically-related edge (see Section 3). More formally, being s the target synset, we define its set of related synsets R_s as follows:

$$R_s = \{s' \mid (s, s') \in E\}$$

where E is the set of hypernym, hyponym and semantically-related edges in BabelNet.

In order to make this set as reliable as possible and reduce the possible error due to the automatic nature of BabelNet, we further refine the set R_s by retaining just those synsets that are strongly related to the target one, i.e., s. To do so, we employ the information coming from the Wikipedia pages associated with s, i.e., p_s, and each synset s' in R_s, i.e., $p_{s'}$. For each page p_i we compute its lexical vector as described in Camacho-Collados, Pilehvar, and Navigli (2016), where words correspond to dimensions and are scored by their lexical specificity value. These lexical representations are then employed to score the similarity between p_s and $p_{s'}$ for each $s' \in R_s$ by means of the Weighted Overlap (WO) measure (Pilehvar, Jurgens, and Navigli 2013). WO determines the similarity between two input pages p_1 and p_2 as follows:

$$WO(p_1, p_2) = \left(\sum_{w \in O} \frac{1}{r_w^{p_1} + r_w^{p_2}} \right) \left(\sum_{i=1}^{|O|} \frac{1}{2i} \right)^{-1}$$

where O is the set of overlapping dimensions of p_1 and p_2 and $r_w^{p_i}$ is the rank of the word w in the lexical vector of p_i. We preferred the weighted overlap over the more common cosine similarity as it has proven to perform better when comparing sparse vector representations (Pilehvar, Jurgens, and Navigli 2013).

Once we have scored all the $(p_s, p_{s'})$ pairs, we create three partitions of R_s, each comprising all the senses s' connected to s with the same relation r, where r can be one among: hypernymy, hyponymy and semantic relatedness. We then retain from each partition only the top-k scored senses[8] $s'_1, \ldots, s'_k$ according to $WO(p_s, p_{s'_i})$. We further refine each filtered partition by solving the conflicts that might arise when a synset s' is related not only to s but also to another concept s'' that shares a lexicalization with s. This is needed since s and s'' represent two different meanings of the same word and thus require distinct contexts to better characterize and distinguish them. Therefore, we remove s' from the set R_s if $WO(p_s, p_{s'}) < WO(p_{s''}, p_{s'})$, or otherwise from the set $R_{s''}$. Finally, for each synset s, we compute the Bag of Contexts BoC_s comprising all the sentences of the pages associated with a sense in R_s.

[4] https://en.wikipedia.org/wiki/Computer

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

[6] https://github.com/google-research/bert

[7] The training of multilingual BERT was performed on the texts coming from Wikipedia in 104 different languages.

[8] We use $k = 10$ for all our experiments.

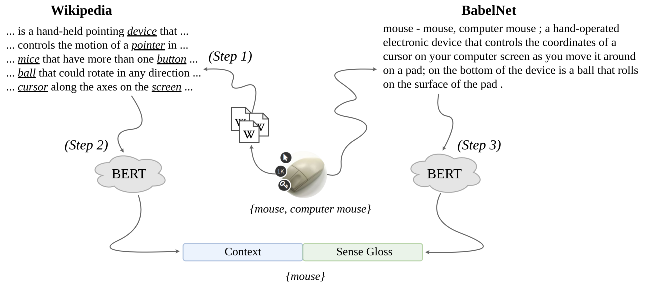
Wikipedia
... is a hand-held pointing device that ...
... controls the motion of a pointer in ...
... mice that have more than one button ...
... ball that could rotate in any direction ...
... cursor along the axes on the screen ...
BabelNet
mouse - mouse, computer mouse ; a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the device is a ball that rolls on the surface of the pad .
(Step 1)
(Step 2)
(Step 3)
BERT
BERT
W
{mouse, computer mouse}
Context
Sense Gloss
{mouse}

Figure 1: Exemplification of the sense embedding's creation for the *device* sense of *mouse*.

4.2 Word Embedding

The aim of the second step is to compute, by means of BERT, the representations of those words in the sentences of BoC_s that best define the target synset s.

First, we define as relevant words for a target synset those that appear in its NASARI lexical vector. Formally, given a synset s and its NASARI vector N_s, we define W_s as the set containing all the non-zero words in N_s. Then, we use W_s to filter out from BoC_s all those sentences which do not contain any of the words therein and compute the embedding of each word $w \in W_s$ by averaging the BERT representations of all its occurrences in BoC_s. That is, given a word $w \in W_s$, we compute its vector v_w as follows:

$$v_w = \frac{\sum_{c \in BoC_s^w} BERT(c, w)}{|BoC_s^w|}$$

where BoC_s^w is the set of contexts in BoC_s where w appears and $BERT(c, w)$ is the vector computed by BERT for the word w in the context c.

At the end of this step, all the words in W_s for a given synset s are associated with their latent representations, which depend on the contexts in BoC_s where they occur.

4.3 Sense Embedding

In this final step, we build a unique representation of each target synset in the knowledge base. Therefore, to prioritize the information that best characterizes a target synset s, we weight each word in W_s, and hence its corresponding embedding, according to its rank in the NASARI vector of s. We then build the final synset representation by combining the word vectors we computed previously. Formally, given a target synset s, the set of its relevant words $w_1, \ldots, w_n \in W_s$ and their corresponding vectors $v_{w_1}, \ldots, v_{w_n}$, we compute the synset embedding of s as follows:

$$v_s = \frac{\sum_{w_i \in W_s} rank(w_i)^{-1} \; v_{w_i}}{\sum_{w_i \in W_s} rank(w_i)^{-1}}$$

where $rank(w_i)$ is the ranking of the word w_i according to its score in the NASARI vector of s. After a vector for each synset s has been computed, we still lack a representation at the sense level, i.e., specific to each lemma of the synset[9]. Therefore, to further enrich and specialize our embeddings at the sense level, we follow Loureiro and Jorge (2019) and leverage the gloss and each of the synset's lemmas. In more detail, i) for each synset s, we enhance its gloss by prepending to it all the lemmas in s; ii) we differentiate the synset gloss for each sense in s by repeating its lemma at the beginning of the gloss; iii) we compute the sense gloss embedding, i.e., the representation of the gloss at the sense level, as the average of the BERT contextual embeddings of the tokens of our enhanced gloss. Lastly, the final representation of the target sense is given by concatenating the embedding of the synset it belongs to, i.e., v_s, with the sense gloss embedding we just computed.

At the end of the three steps outlined in Sections 4.1-4.3, each sense in the knowledge base is associated with a vector which encodes both its contextual and definitional semantics coming from the contexts extracted from Wikipedia and its gloss, respectively. In Figure 1 we exemplify the procedure for creating the embedding of the *device* sense of the lemma *mouse*. As one can see, we compute the representation of the target sense by combining the information we extracted for its corresponding synset, i.e., {*mouse, computer mouse*}, from Wikipedia (left side of the figure) and BabelNet (right side of the figure). As for the first component of the vector, i.e., context, we retrieve all the sentences in the Wikipedia pages collected for the {*mouse, computer mouse*} synset (step 1, Section 4.1). Then we compute, by means of BERT, the embeddings of the relevant words, i.e., device, pointer, etc. (underlined in figure), and average them (step 2, Section 4.2). As for the second component of the vector, i.e., sense gloss, we consider the gloss of the {*mouse, computer mouse*} synset and prepend to it all of its lemmas, i.e., *mouse*, *computer mouse* (upper right side of the figure). Then we specialize it for the *device* sense of *mouse* by adding the lemma *mouse* at the beginning of the text and average the BERT representations of the tokens therein (step 3, Section 4.3). Thus, the final vector for the *device* sense of *mouse* is the concatenation of the context and sense gloss vectors.

5 Experimental Setup

In this Section we report the settings in which we conducted the evaluation of SENSEMBERT when testing it on the English and multilingual WSD tasks. In what follows we introduce the test sets, the system setup along with the reference WSD model, a supervised version of our approach and the comparison systems.

Evaluation Benchmarks As for English, we carried out the evaluation on the test sets in the English WSD framework in Raganato, Camacho-Collados, and Navigli (2017)[10]. This includes five standardized evaluation benchmarks from the past Senseval-SemEval competitions,

[9] We recall from Section 3 that a synset contains several senses, each associated with a lemma.

[10] http://lcl.uniroma1.it/wsdeval/

i.e., Senseval-2 (Edmonds and Cotton 2001), Senseval-3 (Snyder and Palmer 2004), SemEval-07 (Pradhan et al. 2007), SemEval-13 (Navigli, Jurgens, and Vannella 2013), SemEval-15 (Moro and Navigli 2015), together with ALL, the concatenation of the five test sets. As for the multilingual setting, we conducted the experiments on the SemEval-13 (Navigli, Jurgens, and Vannella 2013) and SemEval-15 (Moro and Navigli 2015) multilingual WSD tasks.

In all test sets we considered just nouns, as the NASARI lexical vectors are currently available for nominal synsets only. All performances are reported in terms of F1-measure, i.e., the harmonic mean of precision and recall.

SENSEMBERT Setup We employed two BERT pre-trained models: the English 1024-dimensional and the multilingual 768-dimensional pre-trained cased models for the English and multilingual settings, respectively. Among all the configurations reported by Devlin et al. (2019), we used the sum of the last four hidden layers as contextual embeddings of the words. Moreover, BERT exploits WordPiece tokenization, that is, a token can be further split into several subtokens, e.g., the term "embed" is broken down into two subtokens, namely "em" and "##bed". Thus, the contextual embedding of an input word was computed as the average of its subtoken embeddings.

WSD Model We used a 1-nearest neighbour approach to test SENSEMBERT on the WSD task. For each target word w in the test set we computed its contextual embedding by means of BERT and compared it against the embeddings of SENSEMBERT associated with the senses of w. Hence, we took as prediction for the target word the sense corresponding to its nearest neighbour. We note that the embeddings produced by SENSEMBERT are created by concatenating two BERT representations, i.e., context and sense gloss (see Section 4.3), hence we repeated the BERT embedding of the target instance to match the number of dimensions.

In contrast to most supervised systems, this approach does not rely on the Most Frequent Sense (MFS) backoff strategy, i.e., predicting the most frequent sense of a lemma in WordNet for instances unseen at training time, as SENSEMBERT ensures full coverage for the English nominal senses.

Supervised SENSEMBERT In order to set a level playing field with supervised systems on English, we built a supervised version of SENSEMBERT, i.e., SENSEMBERT$_{sup}$. This version combined the gloss and contextual information (Section 4.3) with the sense-annotated contexts in SemCor (Miller et al. 1993), a corpus of 40K sentences where words have been manually annotated with a WordNet meaning. We leveraged SemCor for building a representation of each sense therein. To this end, we followed Peters et al. (2018) and, given a word-sense pair (w, s), we collected all the sentences $c_1, \ldots, c_n$ where w appears tagged with s. Then, we fed all the retrieved sentences into BERT and extracted the embeddings $BERT(c_1, w), \ldots, BERT(c_n, w)$. The final embedding of s was built by concatenating the average of its context and sense gloss vectors (Figure 1) and its representation coming from SemCor, i.e., the average of $BERT(c_1, w), \ldots, BERT(c_n, w)$. We note that, when a sense did not appear in SemCor, we built its embedding by replacing the SemCor part of the vector with its sense gloss representation.

Comparison Systems We compared SENSEMBERT against the best performing supervised and knowledge-based systems evaluated on the WSD framework for English nouns. Among knowledge-based approaches, we took into account the extension of Lesk comprising word embeddings (Basile, Caputo, and Semeraro 2014, Lesk$_{ext}$+emb),the extended version of UKB with gloss relations (Agirre, de Lacalle, and Soroa 2014, UKB$_{gloss}$) and Babelfy (Moro, Raganato, and Navigli 2014). As for supervised systems we considered an SVM-based classifier integrated with word embeddings (Iacobacci, Pilehvar, and Navigli 2016, IMS+emb), the Bi-LSTM with attention and multi-task objective presented in Raganato, Delli Bovi, and Navigli, Bi-LSTM (2017), and the more recent supervised systems leveraging sense definitions, i.e., HCAN (Luo et al. 2018) and EWISE (Kumar et al. 2019). We also performed a comparison with the two LSTM-based architectures of Yuan et al. (2016, LSTM-LP) and context2vec (Melamud, Goldberger, and Dagan 2016) for learning representations of the annotated sentences in the training corpus. Finally, we applied the 1-NN strategy to two other supervised approaches for creating sense embeddings, namely Peters et al.'s method (2018) using BERT (BERT k-NN) and LMMS (Loureiro and Jorge 2019). All supervised systems, apart from BERT k-NN and LMMS[11], were trained using SemCor and, with the exception of HCAN, EWISE and LMMS, rely on the MFS backoff strategy unless otherwise stated.

As regards the multilingual setting, we took into account OneSeC (Scarlini, Pasini, and Navigli 2019), the best automatically-tagged corpus available for non-English languages. Therefore, we compared the embeddings produced by SENSEMBERT with the state-of-the-art Bi-LSTM-based supervised model trained on OneSeC presented by Scarlini, Pasini, and Navigli (2019, Bi-LSTM$_{OneSeC}$). Moreover, we also created a multilingual version of LMMS by replicating their approach on the data provided by OneSeC (LMMS$_{OneSeC}$) and tested it on all the target languages.

6 Results

In this Section we report the results of the evaluation on the WSD task. We first show the ablation study we carried out to assess the contribution brought by each part of SENSEMBERT's embeddings. We then demonstrate the effectiveness of SENSEMBERT by comparing its results on all standard WSD benchmarks with the existing state of the art.

6.1 Ablation Study

First, we show an ablation study of SENSEMBERT and SENSEMBERT$_{sup}$ on the English ALL test set in Raganato,

[11] We note that, as for SENSEMBERT$_{sup}$, BERT k-NN and LMMS exploit SemCor only to retrieve the sense-annotated contexts that are to be fed to a neural language model.

	Model	Senseval-2	Senseval-3	SemEval-07	SemEval-13	SemEval-15	ALL
KB	MFS	72.1	72.0	65.4	63.0	66.3	67.6
	Lesk$_{ext}$+emb (2014)	74.6	72.7	66.0	66.2	67.8	69.8
	UKB$_{gloss}$ (2014)	70.6	58.4	56.6	59.0	62.3	62.1
	Babelfy (2014)	74.0	66.7	61.0	66.4	69.9	68.6
Sup	IMS+emb (2016)	79.0	74.6	71.1	65.9	72.1	71.9
	Bi-LSTM (2017)	78.6	72.7	71.1	66.4	73.3	71.6
	HCAN (2018)	78.3	73.2	70.9	68.5	73.8	72.6
	EWISE$_{ConvE}$ (2019)	-	-	-	69.4	-	74.0
Sup$_{context}$	context2vec (2016)	78.0	73.1	66.7	65.6	71.6	71.0
	LSTM-LP (2016)	79.6	76.3	71.7	69.5	72.8	-
	BERT k-NN (2019)	71.7	73.0	72.9	65.6	68.4	69.3
	BERT k-NN + MFS (2019)	81.4	76.3	73.6	71.8	74.0	75.5
	LMMS (2019)	81.7	78.7	78.0	75.1	78.2	78.0
Ours	SensEmBERT	80.6	70.3	73.6	74.8	**80.2**	75.9
	SensEmBERT$_{sup}$	**83.7**	**79.7**	**79.9**	**78.7**	**80.2**	**80.4**

Table 1: Comparison in terms of F1 on the nominal instances of the English WSD test sets in Raganato, Camacho-Collados, and Navigli (2017). Approaches are grouped by type: i) knowledge-based systems (*KB*), ii) supervised models for classification (*Sup*), iii) supervised models for learning contextual representations of senses (*Sup$_{context}$*), iv) ours (*Ours*).

Model	F1
Gloss	63.9
SeB$_c$	74.5
SeB$_c$ $\oplus$ Gloss	75.3
SeB$_c$ $\mid$ Gloss	**75.9**

Table 2: Ablation study of SensEmBERT's components on the nouns of the ALL test set in terms of F1.

Model	F1
SemCor	69.3
SemCor $\mid$ Gloss	78.0
SeB$_c$ $\mid$ SemCor	79.2
(SeB$_c$ $\oplus$ SemCor) $\mid$ Gloss	79.6
(SeB$_c$ $\oplus$ Gloss) $\mid$ SemCor	**80.4**

Table 3: Ablation study of SensEmBERT$_{sup}$'s components on the nouns of the ALL test set in terms of F1.

Camacho-Collados, and Navigli (2017) to see how each of their components influences the final results.

As one can see from Table 2, the sense gloss embeddings part alone (Gloss) scores only 63.9 F1 points, meaning that the information encoded therein is not sufficient for providing a clear distinction between senses. However, we show that it is beneficial to SensEmBERT when used to further enrich and specialize its contextual representations. In fact, the sense gloss embeddings and the context-enhanced part of SensEmBERT (SeB$_c$) prove to be complementary by achieving the best F1 score of 75.9 when concatenated together (SeB$_c$ | Gloss)[12], increasing the results over the performance of SeB$_c$ alone by 1.4 points.

Similarly, in Table 3 we show how the contexts extracted with SensEmBERT are also valuable when combined with the vectors extracted from SemCor. In fact, simply concatenating the two representations (SeB$_c$ | SemCor) leads to an increment of 9.9 points over the performance of SemCor alone and 1.2 points over the concatenation of SemCor with the glosses. Moreover, we gain an additional boost in performance when averaging the sense gloss embedding with the context part, by achieving 80.4 F1 in our best configuration, i.e., (SeB$_c$ $\oplus$ Gloss) | SemCor. Therefore, in what follows we report the results of SensEmBERT, i.e., SeB$_c$ | Gloss (see Table 2), and SensEmBERT$_{sup}$, i.e., (SeB$_c$ $\oplus$ Gloss) | SemCor (see Table 3).

6.2 English WSD

We now proceed to testing SensEmBERT on the fine-grained English tasks. In Table 1 we report the results of SensEmBERT and SensEmBERT$_{sup}$ and compare them against the results attained by other knowledge-based and supervised state-of-the-art approaches on all the nominal instances of the test sets in the framework of Raganato, Camacho-Collados, and Navigli (2017).

As one can see, SensEmBERT achieves the best results on ALL when compared to other knowledge-based approaches. These results raise the bar for knowledge-based WSD by improving the existing state of the art by 6.1 F1 points and indicating that SensEmBERT is competitive with supervised models as well. In fact, SensEmBERT ranks second only to LMMS and outperforms all other supervised systems, which, in contrast, rely on sense-annotated data and dedicated WSD architectures. Moreover, we show that we are able to surpass, and hence improve, the existing state of the art by including supervision, i.e., SemCor, in our approach. In fact, SensEmBERT$_{sup}$ proves to be the best system across the board outperforming its competitors on all datasets with an increment of 2.1 points overall compared to LMMS, which also uses SemCor.

6.3 WSD on Rare Words and Senses

To investigate further the benefits brought by SensEmBERT, we carried out the evaluation on only those instances of the test sets which are associated with a rare word or

[12] We use $\oplus$ to represent the average of two vectors and | for their concatenation.

Model	ALL$_{LFS}$	ALL$_{LFW}$
LMMS	66.7	76.3
SensEmBERT	**76.9**	78.1
SensEmBERT$_{sup}$	70.4	**81.1**

Table 4: Comparison in terms of F1 on nouns and nominal senses in the ALL dataset not occurring in SemCor.

sense. Therefore, starting from the ALL dataset we created two additional test sets: i) ALL$_{LFS}$ (Least Frequent Senses), containing the 812 instances in ALL associated with a gold sense not occurring in SemCor; ii) ALL$_{LFW}$ (Least Frequent Words), containing the 528 instances in ALL for a non-monosemous lemma not occurring in SemCor.

In Table 4 we report the performance of SensEmBERT, SensEmBERT$_{sup}$ and LMMS on the two newly-created test sets. SensEmBERT outperforms its direct competitor on both datasets, providing a significant gain of 10.2 F1 points on ALL$_{LFS}$. This implies that our approach is better able to generalize over both words and senses as it can provide diversified contexts, while not being tied to a specific sense-annotated corpus. LMMS, instead, performs more poorly when it comes to predicting rare instances. In fact, even if it provides full coverage of the senses in WordNet, it computes the representations of the senses not in SemCor by averaging the embeddings of the senses therein. Hence, it is biased towards those representations for which sense annotations are provided. In contrast, SensEmBERT$_{sup}$ demonstrates that the contexts extracted from Wikipedia aid better generalization over rare words and senses also when they are coupled with the information from SemCor, thus allowing it to outperform LMMS on the ALL$_{LFS}$ datasets by 3.7 F1 points and to achieve the best result on ALL$_{LFW}$.

6.4 Multilingual WSD

We now test the ability of our approach to build vectors that are also effective for languages other than English. We recall from Section 4 that our method covers all the 104 languages in both BabelNet, BERT and Wikipedia.

In Table 5 we report the results attained by SensEmBERT in the multilingual WSD tasks of SemEval-13 and SemEval-15. We compare our approach with the existing state of the art (Bi-LSTM$_{OneSeC}$) and the embeddings obtained by replicating the LMMS approach on OneSeC's silver data (LMMS$_{OneSeC}$). While our approach also proves its consistency on languages other than English, LMMS loses ground when no manually-curated corpora are available for the target language. In fact, when bootstrapped from a fully-automatic resource such as OneSeC, the performance of LMMS drops heavily on most of the tested languages, since it is not bulletproof when it comes to the unavoidable noise that can be found in silver data. In contrast, not only does SensEmBERT beat its direct competitor (LMMS$_{OneSeC}$) on all test sets by on average 6.6 points, it also sets a new state of the art on all the languages by performing 2.5 points above the best model overall.

Model	SemEval-13				SemEval-15	
	IT	ES	FR	DE	IT	ES
Bi-LSTM$_{OneSeC}$	68.2	72.0	74.8	75.1	62.5	62.8
LMMS$_{OneSeC}$	64.6	67.8	69.0	77.1	62.8	49.4
SensEmBERT	**69.6**	**74.6**	**78.0**	**78.0**	**66.0**	**64.1**

Table 5: Comparison in terms of F1 on the SemEval-2013 and SemEval-2015 multilingual WSD tasks.

The results across different experiments attest the high quality of our embeddings, showing that our approach is robust across languages, and hence enables state-of-the-art multilingual WSD while at the same time relieving the heavy requirement of sense-annotated corpora.

7 Conclusion

In this paper we presented SensEmBERT, a novel approach for creating sense embeddings in multiple languages. SensEmBERT proved to be effective both in the English and multilingual WSD tasks. Indeed, to the best of our knowledge this is the first time, in the neural network era, that a knowledge-based approach, employing a 1-NN strategy has succeeded in surpassing most of its supervised competitors and outperforming the state of the art on rare words and senses. SensEmBERT's generalization ability is further demonstrated by our multilingual experiments, where our approach beats all its alternatives, setting a new state of the art in all the tested languages. Moreover, we show that our context-rich representations are also beneficial when coupled with manually-annotated data, hence enabling the supervised version of SensEmBERT to surpass the bar of 80% accuracy and leaving all the other approaches behind with a gap of more than 2.0 points. We release sense embeddings in five different languages for all the WordNet nominal synsets at http://sensembert.org.

As future work, we plan to extend our approach to cover the other main POS tags, i.e., verbs, adjectives and adverbs, by exploiting other knowledge resources, such as VerbAtlas (Di Fabio, Conia, and Navigli 2019) and SyntagNet (Maru et al. 2019). Moreover, we plan to leverage the sense embeddings provided by SensEmBERT to create high-quality silver data for WSD in multiple languages.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487 under the European Union's Horizon 2020 research and innovation programme.

This work was supported in part by the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of the Sapienza University of Rome.

References

Agirre, E.; de Lacalle, O. L.; and Soroa, A. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics* 40(1):57–84.

Basile, P.; Caputo, A.; and Semeraro, G. 2014. An Enhanced Lesk Word Sense Disambiguation Algorithm through a Distributional Semantic Model. In *Proc. of COLING*, 1591–1600.

Bevilacqua, M., and Navigli, R. 2019. Quasi Bidirectional Encoder Representations from Transformers for Word Sense Disambiguation. In *Proc. of RANLP*, 122–131.

Camacho-Collados, J.; Pilehvar, M. T.; and Navigli, R. 2016. Nasari: Integrating explicit knowledge and corpus statistics for a multilingual representation of concepts and entities. *Artificial Intelligence* 240:36–64.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, 4171–4186.

Di Fabio, A.; Conia, S.; and Navigli, R. 2019. VerbAtlas: a Novel Large-Scale Verbal Semantic Resource and Its Application to Semantic Role Labeling. In *Proc. of EMNLP-IJCNLP*, 627–637.

Edmonds, P., and Cotton, S. 2001. Senseval-2: overview. In *Proc. of SENSEVAL*, 1–5.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Database*. Cambridge, MA: MIT Press.

Gale, W. A.; Church, K.; and Yarowsky, D. 1992. A method for disambiguating word senses in a corpus. *Computers and the Humanities* 26:415–439.

Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proc. of ACL*, volume 1, 897–907.

Kumar, S.; Jat, S.; Saxena, K.; and Talukdar, P. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proc. of ACL*, 5670–5681.

Lacerra, C.; Bevilacqua, M.; Pasini, T.; and Navigli, R. 2020. CSI: A coarse sense inventory for 85% word sense disambiguation. In *Proc. of AAAI*.

Lafon, P. 1980. Sur la variabilité de la fréquence des formes dans un corpus. *Mots. Les langages du politique* 1(1):127–165.

Loureiro, D., and Jorge, A. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proc. of ACL*, 5682–5691.

Luo, F.; Liu, T.; He, Z.; Xia, Q.; Sui, Z.; and Chang, B. 2018. Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proc. of EMNLP*, 1402–1411.

Maru, M.; Scozzafava, F.; Martelli, F.; and Navigli, R. 2019. SyntagNet: Challenging Supervised Word Sense Disambiguation with Lexical-Semantic Combinations. In *Proc. of EMNLP-IJCNLP*, 3525–3531.

Melamud, O.; Goldberger, J.; and Dagan, I. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proc. of CoNLL*, 51–61.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proc. of NIPS*, 3111–3119.

Miller, G. A.; Leacock, C.; Tengi, R.; and Bunker, R. 1993. A semantic concordance. In *Proc. of the Workshop on Human Language Technology*, 303–308.

Moro, A., and Navigli, R. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proc. of SemEval-2015*, 288–297.

Moro, A.; Raganato, A.; and Navigli, R. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL* 2:231–244.

Navigli, R., and Ponzetto, S. P. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193:217–250.

Navigli, R.; Jurgens, D.; and Vannella, D. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proc. of Semeval 2013*, volume 2, 222–231.

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

Pasini, T., and Navigli, R. 2017. Train-o-matic: Large-scale supervised word sense disambiguation in multiple languages without manual training data. In *Proc. of EMNLP*, 78–88.

Pasini, T.; Elia, F.; and Navigli, R. 2018. Huge automatically extracted training-sets for multilingual word sensedisambiguation. In *Proc. of LREC*, 1694 – 1698.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proc. of EMNLP*, 1532–1543. Association for Computational Linguistics.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proc. of NAACL*, 2227–2237.

Pilehvar, M. T.; Jurgens, D.; and Navigli, R. 2013. Align, disambiguate and walk: A unified approach for measuring semantic similarity. In *Proc. of ACL*, 1341–1351.

Pradhan, S. S.; Loper, E.; Dligach, D.; and Palmer, M. 2007. Semeval-2007 task 17: English lexical sample, SRL and all words. In *Proc. of Semeval-2007*, 87–92.

Raganato, A.; Camacho-Collados, J.; and Navigli, R. 2017. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *Proc. of EACL*, 99–110.

Raganato, A.; Delli Bovi, C.; and Navigli, R. 2017. Neural Sequence Learning Models for Word Sense Disambiguation. In *Proc. of EMNLP*, 1156–1167.

Scarlini, B.; Pasini, T.; and Navigli, R. 2019. Just "OneSeC" for Producing Multilingual Sense-Annotated Data. In *Proc. of ACL*, volume 1, 699–709.

Snyder, B., and Palmer, M. 2004. The english all-words task. In *Proc. of Senseval 3*, 41–43.

Vial, L.; Lecouteux, B.; and Schwab, D. 2019. Sense Vocabulary Compression through the Semantic Knowledge of WordNet for Neural Word Sense Disambiguation. In *Proc. of Global Wordnet Conference*.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proc. of the EMNLP Workshop BlackboxNLP*, 353–355.

Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *CoRR*.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*.

Yuan, D.; Richardson, J.; Doherty, R.; Evans, C.; and Altendorf, E. 2016. Semi-supervised word sense disambiguation with neural models. In *Proc. of COLING*, 1374–1385.