Word Sense Disambiguation: Towards Interactive Context Exploitation from Both Word and Sense Perspectives

Ming Wang[1] and **Yinglin Wang**[2, *]
School of Information Management and Engineering
Shanghai University of Finance and Economics, Shanghai, China
[1]wangming@163.sufe.edu.cn, [2]wang.yinglin@shufe.edu.cn

Abstract

Lately proposed Word Sense Disambiguation (WSD) systems have approached the estimated upper bound of the task on standard evaluation benchmarks. However, these systems typically implement the disambiguation of words in a document almost independently, underutilizing sense and word dependency in context. In this paper, we convert the nearly isolated decisions into interrelated ones by exposing senses in context when learning sense embeddings in a similarity-based Sense Aware Context Exploitation (SACE) architecture. Meanwhile, we enhance the context embedding learning with selected sentences from the same document, rather than utilizing only the sentence where each ambiguous word appears. Experiments on both English and multilingual WSD datasets have shown the effectiveness of our approach, surpassing previous state-of-the-art by large margins (3.7% and 1.2% respectively), especially on few-shot (14.3%) and zero-shot (35.9%) scenarios.

1 Introduction

Word Sense Disambiguation (WSD) is the task of determining a word's sense given its context. Recently, contextualized representation learning (Devlin et al., 2019; Liu et al., 2019) have accelerated the advancement of WSD, raising the performance on a standard evaluation framework (Raganato et al., 2017a) from slightly higher than 70% (Raganato et al., 2017b; Luo et al., 2018; Kumar et al., 2019) to about 80% (Vial et al., 2019; Blevins and Zettlemoyer, 2020; Bevilacqua and Navigli, 2020). This is an estimated upper bound of the task, which is from the inter-annotator agreement: the percentage of words that are annotated with the same meaning by two or more annotators (Navigli, 2009). There is a clear trend that supervised systems tend to incorporate sense knowledge into their architecture, ranging from sense definition, usage examples to sense relation.

However, the disambiguation of words in a document is almost independent of each other, especially from the perspective of senses in context. The connection of each word's disambiguation is limited to the utilization of a sentence (Loureiro and Jorge, 2019; Huang et al., 2019; Hadiwinoto et al., 2019; Scarlini et al., 2020a) or a small window of text (Bevilacqua and Navigli, 2020) because of computation cost or model restriction. More severely, the interaction of senses in context is barely explored. Similar to word cooccurrence, the appearance of one sense can sometimes dominate the choice of another sense in the same sentence (Agirre et al., 2014; Maru et al., 2019).

In this paper, we introduce SACE, a similarity-based WSD approach. Precisely, we transform the previously almost isolated disambiguation of words in a document into interrelated ones to maximize the contribution of context from both word and sense perspectives. We summarize our contributions as follows:

1. We devise an interactive sense embedding learning technique that takes into account senses in context via a selective attention layer in a neural architecture. It connects senses via their appearance in a piece of text rather than using manually constructed sense relations, being less costly.
2. We introduce a method to better exploit the

* corresponding author

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 5218–5229
August 1–6, 2021.

context sentences of an ambiguous word in the neural architecture by selecting important sentences from the same document according to sentence relatedness.

3. With experiments on corresponding datasets, the proposed architecture is proved to have an overwhelming advantage of few-shot and zero-shot WSD learning ability compared with other strong baselines.
4. We show that the proposed architecture is portable to multilingual scenarios when trained merely on an English dataset with a multilingual pre-trained model, achieving new state-of-the-art on most tested benchmarks and the combined one.

2 Related Work

There are mainly two alternatives for solving WSD, namely knowledge-based and supervised approaches. While the former mainly relies on a sense inventory for disambiguation, the latter is dependent on sense-annotated corpora to train a sense classifier, either for each word or the whole vocabulary. However, many recently proposed systems combine the above two strategies, injecting sense knowledge into their supervised models while somehow inadequately modeling the provided context in a document from both word and sense perspectives.

2.1 Supervised Method

Early supervised approaches model the relational pattern between an ambiguous word's local features and its gold sense from sense-annotated data. IMS (Zhong and Ng, 2010) was one of the most prevalent systems that trained a sense classifier for each lemma in training data. In comparison, Raganato et al. (2017b) unified the disambiguation of words into a single sequence labeling architecture, relieving the efficiency issue. Many following systems improved this architecture by incorporating sense knowledge.

For unseen lemmas, these systems require most frequent sense (MFS) fallback (select the most frequent candidate sense in the training data). To tackle this problem, LMMS (Loureiro and Jorge 2019) implements the disambiguation in a similarity-based manner. It learns a sense embedding for each labeled sense in SemCor (Miller et al., 1994) and maps them to full coverage of WordNet (Miller, 1995) senses using sense relations. BERT (Devlin et al., 2019) is used as a feature-extraction module for both gloss and context encoding. Further, BEM (Blevins and Zettlemoyer, 2020) utilizes two encoders for the above approach in a fine-tuning manner. Although the model is more effective even without exploiting sense knowledge other than glosses, it takes around 2.5 days for training.

The employment of sense relations in previous supervised systems is mostly limited to explicitly defined sense relations including hypernymy and hyponymy relation, severely neglecting how senses in context contribute to the selection of a word's sense.

2.2 Context Exploitation

For supervised WSD approaches, it is typical to use a small fraction of the whole context to carry out disambiguation, such as a sentence, or a sliding window of text. In contrast, knowledge-based WSD approaches tend to more sufficiently exploit a word's context, ranging from a sentence (Lesk, 1986; Wang and Wang, 2020), a few sentences (Agirre et al., 2018, Wang et al., 2020) to even the whole document (Chaplot and Salakhutdinov, 2018). Some studies draw in out-of-dataset context (Ponzetto and Navigli, 2010; Scarlini et al., 2020a) for disambiguation, including Wikipedia documents. Therefore, it is worth exploring whether the disambiguation of words within the same document can benefit from each other in a supervised system.

The utilization of senses in context is far less investigated compared with words in context. UKB (Agirre et al., 2014, a knowledge-based system) is one of the related systems that model sense relations in context. It first connects senses in context via WordNet sense relations and operates personalized PageRank on the constructed sense graph to decide sense importance. For each word, the most important potential sense is considered as the correct sense. SyntagNet (Maru et al., 2019) improves the idea by introducing manually disambiguated sense pairs in context during sense graph construction. Although the system was able to challenge supervised systems at the time, it relied on human labor to obtain sense pairs in context. There was no attempt on integrating the utilization of senses in context into a supervised architecture.

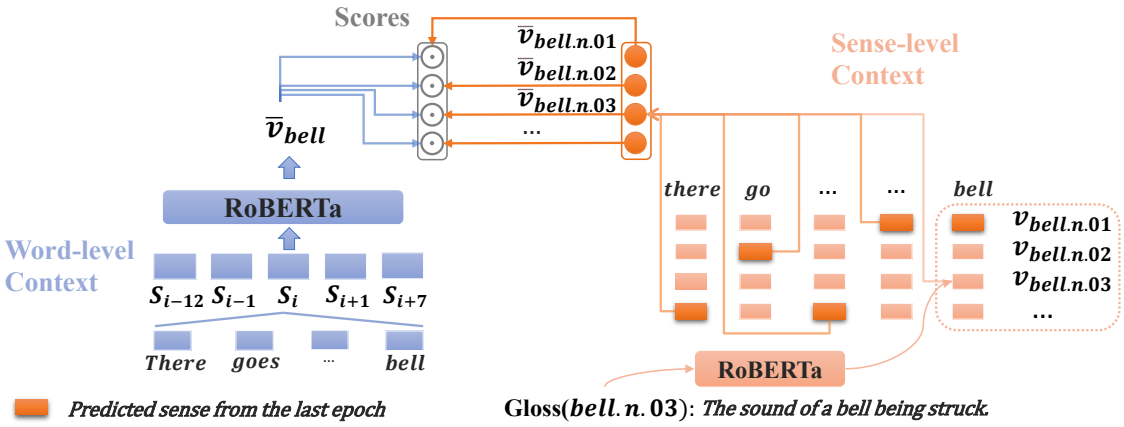

Figure 1: SACE Framework.

3 Preliminary

WSD is to select the correct sense $\tilde{s}_j$ of a word w_{ij} given its context. w_{ij} is the j^{th} word in the i^{th} sentence $S_i = \{w_{i1}, w_{i2}, \dots, w_{ij}, \dots, w_{in}\}$ of a document $D = \{S_1, S_2, \dots, S_i, \dots, S_m\}$. The candidate senses $s(w_{ij}) = \{s_{j1}, s_{j2}, \dots, s_{jk}, \dots, s_{jo}\}$ are from a sense inventory such as WordNet. Here, i, j, and k denote the index of sentence, word, and sense respectively.

In a similarity-based WSD approach, the disambiguation of a word is determined by the similarity between its context representation $v_{w_{ij}}$ and each candidate sense representation $v_{s_{jk}}$. In many cases, both representations are vectors and the similarity is measured by their dot product after normalization. Then, the sense with the highest similarity is selected as the correct sense.

Typically, a word's context representation is learned using the sentence S_i where the word appears (Loureiro and Jorge, 2019; Scarlini et al., 2020a; Scarlini et al., 2020b). The representation of a candidate sense is obtained using its gloss/definition $G_{s_{jk}}$ defined in WordNet (Blevins and Zettlemoyer, 2020). A common approach of encoding these two sequences in recent research is to utilize pre-trained models such as BERT, RoBERTa (Liu et al., 2019), and so on, taking the sum of the outputs of the last four layers as encoded features (Loureiro and Jorge, 2019; Scarlini et al., 2020a), as in (1) and (2). Before feeding S_i and $G_{s_{jk}}$ to the models, a special token [CLS]/[SEP] is added to the beginning/end of the sequence, modifying them into $\bar{S}_i$ and $\bar{G}_{s_{jk}}$, respectively.

$$v_{w_{ij}} = \sum_z^{(-4,-1)} BERT_z^j(\bar{S}_i) \quad (1)$$

For each w_{ij}'s context representation, a normal choice is to utilize the model's output at the position of the word (j), using $\bar{S}_i$ as input, shown in equation (1). If the word is tokenized into several pieces, their mean is taken. In contrast, for each sense representation, when it is fine-tuning a pre-trained model, the sense embedding is the output at the position of [CLS] (Blevins and Zettlemoyer, 2020), with the modified gloss as input, as in (2).

$$v_{s_{jk}}^{[CLS]} = \sum_z^{(-4,-1)} BERT_z^{[CLS]}(\bar{G}_{s_{jk}}) \quad (2)$$

To utilize the supervision from a training corpus, a cross-entropy loss is implemented against the similarity distribution of candidate senses (the SoftMax product without index k in (3)) and the one-hot ground-truth distribution, shown in equation (4). $V_{s(w_{ij})} \in \mathbb{R}^{|s(w_{ij})| \times h}$ is a matrix of concatenated sense embeddings arranged in rows. h is the dimension of the pre-trained model's hidden states (768 or 1024 of BERT). y_{jk} is equal to 1 when s_{jk} (the k^{th} sense of w_{ij}) is the correct sense, otherwise 0, representing each element in the ground-truth one-hot vector. For prediction, the model selects the sense with the largest dot product for each word.

$$sim\left(v_{w_{ij}} \cdot v_{s_{jk}}^{[CLS]}\right) = softmax(V_{s(w_{ij})} v_{w_{ij}})^k \quad (3)$$

$$\mathcal{L}(w_{ij}, s_j) = -\sum_{k=1}^{|s(w_{ij})|} y_{jk} log(sim(v_{w_{ij}} \cdot v_{s_{jk}}^{[CLS]})) \quad (4)$$

In the above approach (from BEM, Blevins and Zettlemoyer, 2020), the embedding learning process of different senses is independent of each other, relying merely on sense gloss. Besides, the

interaction between different words' disambiguation is limited to the utilization of a sentence, leading to inadequate exploitation of the words in context. Therefore, we transform the above almost isolated decisions into interrelated ones by learning the sense and context embeddings interactively.

4 SACE: Sense Aware Context Exploitation in Supervised WSD

4.1 Sense-level Context (SlC)

The interactive sense embedding learning mainly involves a selective attention layer upon the original sense embeddings from the pre-trained model. The goal of this interaction is to assist the learning of one sense's embedding to be aware of the others in the same context. It is supported by the fact that many sense pairs are more commonly used than the others.

In practice, each of the ambiguous words in the document has several candidate senses, which poses questions about which senses should be attended in the selective attention layer. To address this problem, we make use of the iterative characteristic of the model training. In other words, the system's predicted senses of each word within a particular context from the former iteration are attended. For the first iteration, the first sense of each word in context is attended. In such a strategy, the senses of monosemous words (has a single sense) can be exploited at all iterations.

For convenient demonstration, we use the embedding of predicted senses $\hat{s}_p$ of the context words in S_i to enhance that of each sense s_{jk} of word w_{ij}. We note that, S_i can be a larger context. In equation (5), n is the number of words in S_i. In (6), $W \in \mathbb{R}^{h \times h}$ is a learnable weight matrix.

$$\bar{v}_{s_{jk}} = v_{s_{jk}}^{[CLS]} + \sum_{p=1(p \neq j)}^{n} \alpha(s_{jk}, \hat{s}_p)\, v_{\hat{s}_p}^{[CLS]} \quad (5)$$

$$\alpha(s_{jk}, \hat{s}_p) = W v_{s_{jk}}^{[CLS]} \cdot W v_{\hat{s}_p}^{[CLS]} \quad (6)$$

The attention score in (6) only takes into consideration the representation at [CLS] position (sentence level representation) for each gloss, neglecting the relatedness between each gloss word of two senses. To tackle this, we devise a combined attention score by considering both [CLS] and gloss word relevance, in equation (7). g is a predefined gloss length of all senses for normalization. $v_{s_{jk}}^{a} \in \mathbb{R}^{h \times 1}$ is obtained with equation (2) by changing the output position to a. If the length (e.g., l) of a sense gloss is smaller than g, $v_{s_{jk}}^{a}$ is a zero vector where a is larger than l.

$$\alpha(s_{jk}, \hat{s}_p) = W v_{s_{jk}}^{[CLS]} \cdot W v_{\hat{s}_p}^{[CLS]} + \frac{1}{g^2} \sum_{a=1}^{g} \sum_{b=1}^{g} (W v_{s_{jk}}^{a} \cdot W v_{\hat{s}_p}^{b}) \quad (7)$$

4.2 Word-level Context (WlC)

In many previous supervised systems, the disambiguation of one word in a sentence is isolated from the words in the other sentences of the same document. We convert the isolated disambiguation into interactive ones by utilizing several highly related sentences within the same document for context embedding learning.

For each sentence S_i, we select its related sentences under two criteria, with one being the distance to S_i, and the other being the semantic relatedness to S_i. The first criterion can be regarded as local features and the second one is aimed at injecting global features while maintaining a low noise level.

From the perspective of local features, directly surrounding sentences within a window are used as related sentences. For global features, we score context sentences and utilize the top related sentences for context embedding learning. Precisely, in a document D, we regard each sentence as a document d and calculate the TF-IDF score of each word in the vocabulary v of D for all sentences. The intuition behind modeling sentences with TF-IDF is that we find the average length of SemCor sentences is 22, which is reasonably long. This represents the original document as a matrix $V_D \in \mathbb{R}^{m \times |v|}$, where each row and column indicate sentence and word dimension respectively. For instance, $V_D(S_i, w_{ij})$ is the TF-IDF score of w_{ij} in S_i. The score of S_j concerning S_i is shown as follows:

$$score_{S_i}(S_j) = V_D(S_i) \cdot V_D(S_j) \quad (8)$$

After scoring all context sentences for each sentence S_i, we concatenate related sentences with S_i and utilize them as an input to BERT for context embedding learning. As an example, $\{S_{i-1}, S_{i+1}\}$ are related sentences from local features, and if $\{S_{i-12}, S_{i+7}\}$ are top-scored sentences from global features, we use $C_i = \{\bar{S}_{i-12}, \bar{S}_{i-1}, \bar{S}_i, \bar{S}_{i+1}, \bar{S}_{i+7}\}$ as an input to equation (1) and retrieve the enhanced context embedding $\bar{v}_{w_{ij}}$ of each word

w_{ij} in S_i. In such a way, different C_i is retrieved for each sentence in the document. We note that, when the total sequence length is longer than 512, we remove the furthest sentences away from $\bar{S}_i$. For instance, $\bar{S}_{i-12}$, $\bar{S}_{i+7}$ and so on in the above example will be removed in order.

Finally, $v_{w_{ij}}$ and $v_{s_{jk}}^{[CLS]}$ in equation (4) are replaced with $\bar{v}_{w_{ij}}$ and $\bar{v}_{s_{jk}}$ respectively to calculate the loss, with which to update the weights of the pre-trained model and the selective attention layer.

4.3 Try-again Mechanism (TaM)

In a previous similarity-based WSD approach, Wang and Wang (2020) proposed a Try-again Mechanism (TaM) that takes into account not only the similarity of w_{ij}'s context embedding to the sense embedding of s_{jk}, but also to the sense embedding of $s_r \in s_{related}$ during evaluation. Here, s_r and s_{jk} are connected by either WordNet relations or the super-sense relation (i.e., senses that belong to the same super-sense category in WordNet). This mechanism in (9) manages to boost the performance of its knowledge-based system by a relatively large margin.

$$sim(w_{ij}, s_{jk}) = v_{w_{ij}} \cdot v_{s_{jk}} + \max_{s_r \in s_{related}} (v_{w_{ij}} \cdot v_{s_r}) \quad (9)$$

In this subsection, we reconstruct TaM so that it becomes effective in our model. This process helps the disambiguation of words to be even more interactive since it considers an increased number of senses by utilizing sense relation knowledge.

In our implementation, we replace the above relations with only those derived from Coarse Sense Inventory (CSI, Lacerra et al., 2020). Similar to the utilization of super-sense categories, we connect senses that belong to the same label in CSI as related senses. Also, we change the direct sum of the above two similarities into a weighted sum using a hyperparameter β.

$$sim(w_{ij}, s_{jk}) = (1-\beta) * \bar{v}_{w_{ij}} \cdot \bar{v}_{s_{jk}} + \beta * \max_{s_r \in s_{related}} (\bar{v}_{w_{ij}} \cdot \bar{v}_{s_r}) \quad (10)$$

In addition, our approach only learns a sense embedding for the candidate senses whose lemma is annotated in training data. Therefore, in TaM, we save sense embeddings from training for each epoch and use them to implement TaM during evaluation. It is worth mentioning that for senses that do not have a sense embedding in $s_{related}$, we neglect their calculation in equation (10).

5 Experiment Settings

5.1 Datasets

To validate the effectiveness of our approach, we use SemCor and an evaluation framework[†] to train and evaluate our model, SACE$_{base}$, respectively. The evaluation framework contains 5 English all-words WSD benchmarks. We report the experimental results on each dataset including SensEval-2 (SE2, Palmer et al., 2001), SensEval-3 (SE3, Snyder and Palmer, 2004), SemEval-2007 Task-17 (SE07, Pradhan et al., 2007), SemEval-2013 (SE13, Navigli et al., 2013) and SemEval-2015 (SE15, Moro and Navigli, 2015). Also, the results from Part-Of-Speech (POS) perspectives on their combined dataset (ALL) are reported. Following previous works, we train large models, SACE$_{large}$ on SemCor and SACE$_{large+}$ on SemCor, WordNet Gloss Tagged (WNGT), and WordNet examples (WNE) for fair comparisons. Here, WNE is regarded as an extra sense gloss and is concatenated after the original sense gloss for sense embedding learning, which is similar to the implementation in SREF (Wang and Wang, 2020).

For few-shot WSD, we partition ALL according to the gold label of each annotation into ALL$_{WN_1st}$ and ALL$_{WN_others}$. Besides, according to whether senses and lemmas of ALL instances appear in SemCor, we extract two subsets, ALL$_{ZSS}$ and ALL$_{ZSL}$, to evaluate the zero-shot learning ability of our model.

For cross-lingual datasets, we use the WordNet version of the latest evaluation framework[‡] which contains test datasets for Spanish, Italian, French, and German. These datasets are preprocessed data from SemEval-2013 (Navigli et al., 2013) and SemEval-2015 (Moro and Navigli, 2015). The former only disambiguates nouns while the latter covers words in four POS (noun-N, verb-V, adjective-A, adverb-R).

We note that the performance in each table is reported with F1 in percentage.

5.2 Model Design

[†] http://lcl.uniroma1.it/wsdeval/home

[‡] https://github.com/SapienzaNLP/mwsd-datasets

Our base and large model utilize RoBERTa$_{base}$ and RoBERTa$_{large}$ respectively, which perform relatively better than BERT models. For cross-lingual evaluation, we fine-tune XLM-RoBERTa-base (SACE$_{mul}$, Conneau et al., 2020) with the same training data as SACE$_{large+}$, following the setting in EWISER. In each system, two encoders are adopted, with one being a context encoder and the other being a sense gloss encoder. This is identical to the setting in BEM. We note that a major difference is that the pre-trained model adopted in the above papers is BERT.

The hyperparameters of our model are selected using SE07. They include the number of surrounding sentences (2) on both sides of S_i, the number of top related sentences (2) of S_i and β (0.1) in TaM. The learning rate for SACE$_{base}$, SACE$_{large}$, SACE$_{large+}$, and SACE$_{mul}$ is 1e-5, 1e-6, 1e-6, and 5e-6 respectively.

To accelerate the model training, we organize the sentences in a document into batches according to the total number of candidate senses (400 for SACE$_{base}$ and SACE$_{mul}$, 150 for SACE$_{large}$ and SACE$_{large+}$), i.e., if the total number of candidate senses exceeds 400 or 150 when adding a sentence, then the sentence belongs to the next batch. For each batch, the gloss and context encoders are only called once. The context and gloss length is normalized to the maximal sequence length within each batch to reduce unnecessary padding and computation. Also, apex is employed for mixed-precision computing. More details are shown in Appendix A.

5.3 Baselines

We compare the proposed model with previous supervised state-of-the-art from different perspectives. These systems include Sense Vocabulary Compression (SVC, Vial et al., 2019), EWISE (Kumar et al., 2019), LMMS (Loureiro and Jorge, 2019), GLU (Hadiwinoto et al., 2019), GlossBERT (Huang et al., 2019), EWISER (Bevilacqua and Navigli, 2020), BEM (Blevins and Zettlemoyer, 2020), ARES (Scarlini et al., 2020b) and SREF (Wang and Wang, 2020). BEM is our direct baseline, which utilizes two encoders to learn context and sense embedding separately and achieves state-of-the-art with only SemCor.

For cross-lingual evaluation, we compare our results with those reported in SyntagNet, EWISER, ARES, MuLaN (Barba et al., 2020). These systems are all recently proposed systems with state-of-the-art performance.

Ablation Study	ALL	Δ
SACE$_{base}$	**80.9**	0
-w/o WlC	79.4	-1.5
-w/o SlC([CLS]+word)	79.7	-1.2
-w/o TaM	80.3	-0.6
-w/o SlC(word)	80.4	-0.5
-w/o all	78.4	-2.5

Table 1: Ablation study of SACE$_{base}$ on ALL

6 Results

6.1 Ablation Analysis

In this subsection, we demonstrate how each component of our model benefits WSD performance. In table 1, the system's performance on ALL has illustrated that enhancing the interaction between different words' disambiguation in the same document (WlC) can raise the system's performance by the largest margin, 1.5 F1. This promotion is slightly larger than that (1.2 F1) provided by the interactive sense embedding learning (SlC). The gloss word attention in SlC is also proved effective, which helps increase the system's performance by 0.5 F1, similar to the contribution of TaM, 0.6 F1. Most importantly, when all components are removed, the performance on ALL decreases to 78.4 F1. We note that the baseline here is different from BEM since we remove unnecessary padding and utilize RoBERTa. This has dramatically accelerated the training process from 3.5 hours to 0.5 hour per epoch while achieved similar performance. We also note that the experimental results reported in this paper are obtained using the same random seed as BEM. With different random seeds, the performance gap on ALL between SACE$_{base}$ and its baseline (-w/o all) ranges from 1.7 F1 to 2.7 F1.

6.2 All-words WSD

Table 2 demonstrates how our systems and lately proposed baselines perform on different partitions of ALL. When it is trained on SemCor, SACE$_{base}$ has already outperformed all its competitors by at least 1.9 F1, on ALL. This is obtained without utilizing prior sense relation knowledge. It is the first system that surpasses the estimated upper bound (80 F1) of the task using only SemCor.

Except GlossBERT and BEM, the other systems adopt BERT$_{large}$ as their pre-trained model. When

Training data	Systems	Datasets					Concatenation of all Datasets				
		SE2	SE3	SE07	SE13	SE15	ALL	N	V	A	R
SemCor	SVC (GWNC2019)	77.5	77.4	69.5	76.0	78.3	76.7	79.6	65.9	79.5	85.5
	EWISE (ACL2019)	73.8	71.1	67.3*	69.4	74.5	71.8*	74.0	60.2	78.0	82.1
	LMMS (ACL2019)	76.3	75.6	68.1	75.1	77.0	75.4	78.0	64.0	80.5	83.5
	GlossBERT (EMNLP2019)	77.7	75.2	72.5*	76.1	80.4	76.8*	-	-	-	-
	GLU (EMNLP2019)	75.5	73.6	68.1*	71.1	76.2	73.7*	-	-	-	-
	ARES (EMNLP2020)	78.0	77.1	71.0	77.3	83.2	77.9	80.6	68.3	80.5	83.5
	SREF (EMNLP2020)	78.6	76.6	72.1	78.0	80.5	77.8	80.6	66.5	82.6	84.4
	EWISER (ACL2020)	78.9	78.4	71.0	78.9	79.3*	78.3*	81.7	66.3	81.2	85.8
	BEM (ACL2020)	79.4	77.4	74.5*	79.7	81.7	79.0*	81.4	68.5	83.0	87.9
	SACE$_{base}$	80.9	79.1	74.7*	82.4	**84.6**	80.9*	83.2	71.1	85.4	87.9
	SACE$_{large}$	**82.4**	**81.1**	**76.3***	**82.5**	83.7	**81.9***	**84.1**	**72.2**	**86.4**	**89.0**
SemCor +WNGT +WNE	SVC (GWNC2019)	79.7	77.8	73.4	78.7	82.6	79.0	81.4	68.7	83.7	85.5
	EWISER (ACL2020)	80.8	79.0	75.2	80.7	81.8*	80.1*	82.9	69.4	83.6	**87.3**
	SACE$_{large+}$	**83.6**	**81.4**	**77.8**	**82.4**	**87.3***	**82.9***	**85.3**	**74.2**	**85.9**	**87.3**

Table 2: English all-words WSD performance on different partitions of ALL utilizing two sets of training data. Following SREF, those marked with * are (partially) obtained as a validation set. SOTA is in **bold**.

we use RoBERTa$_{large}$, SACE$_{large}$ can further reach 81.9 F1 on ALL, surpassing the previous state-of-the-art by 2.9 (3.7% of 79.0) F1. This is a large margin given that BEM and EWISER are strong baselines. When extra training data and WNE are employed, a similar margin, 2.8 F1, is attained on ALL.

Our systems also obtain state-of-the-art performance on each dataset, with the margin ranging from 0.2 to 2.9 F1 for SACE$_{base}$ and 1.8 to 3.0 F1 for SACE$_{large}$, in the first category. As for SACE$_{large+}$, the margin above the previous best system for each dataset is even larger, varying from 1.7 to 5.5 F1. It is noteworthy that SACE$_{base}$ outperforms SACE$_{large}$ by 0.9 F1 on SE15 and they obtain similar performance on SE13. These two datasets are less ambiguous since each lemma has fewer candidate senses on average. This illustrates the competitive disambiguation capability of SACE$_{base}$ on easier instances. We also note that the development set in two categories is different, with the first being SE07 and the second being SE15. This is because we follow most systems' setting in the first category and follow EWISER's setting in the second category for better comparison.

For the performance on different POS, our systems set new lines for all of them in ALL. The largest advancement comes from the higher disambiguation ability of verbs, making our system the first to reach the line of 70 F1. The systems also obtain unprecedented performance on noun disambiguation, surpassing the previous best system by 1.5, 2.4, and 2.4 for SACE$_{base}$, SACE$_{large}$, and SACE$_{large+}$ respectively. SACE$_{large+}$ is the only system that exceeds 85 F1 on noun disambiguation.

6.3 Rare and Unseen Sense Disambiguation

Rare Sense Disambiguation Table 3 reports different systems' performance on ALL$_{WN_1st}$ and ALL$_{WN_others}$, which has 4278 and 2525 annotations respectively. Compared with previous well-performing systems including LMMS and SREF, our systems achieve much better performance on both datasets, with the major contribution coming from WordNet 1st sense disambiguation. On the contrary, SACE and BEM obtain similar performance on ALL$_{WN_1st}$ while SACE can disambiguate rare senses with higher accuracy. This shows a better few-shot learning ability of SACE in comparison to BEM because the ALL$_{WN_others}$ dataset only contains the words whose correct sense appears infrequently in SemCor.

Here, sense disambiguation is defined as whether a system can select the sense as the correct sense, which is viewed from a sense perspective. In comparison, word or lemma disambiguation is to determine the correct sense of a word or lemma, which is viewed from a word perspective.

Unseen Sense Disambiguation In the second column of table 4, different system's performance on ALL$_{ZSS}$ (691 polysemous instances) is provided. This dataset only contains polysemous words whose gold label is not in SemCor, which evaluate the zero-shot sense disambiguation ability of different systems. It is shown that lately proposed systems have an overwhelming advantage of zero-shot sense disambiguation over ordinary baselines including WordNet S1 and BERT-base, with the margin ranging from about 12 F1 to about 42 F1. Specifically, although BEM outperforms its

Models	ALL_{WN_1st} (n=4728)	ALL_{WN_other} (n=2525)
WordNet 1st	100	0
LMMS	87.6	52.6
SREF	91.0	53.2
BEM	93.6	51.7
$SACE_{base}$	94.2	56.1
$SACE_{large}$	94.1	59.0
$SACE_{large+}$	**94.7**	**60.8**

Table 3: Rare sense disambiguation on ALL

baselines by around 25 F1, our base and large system still beat BEM by almost 12 and 18 F1 respectively.

In the third column, we follow previous works and show how different systems perform on ALL_{ZSS*} (1139 instances including monosemous ones). The aforementioned gaps become narrower since each system can correctly disambiguate monosemous instances.

Unseen Lemma Disambiguation In the last two columns of table 4, the systems' performance on zero-shot lemmas is presented. The difference between these two datasets is whether monosemous lemmas are included. We believe it is more reasonable to focus on ALL_{ZSL} (222 polysemous instances) since monosemous lemmas do not require disambiguation and thus the statistics on ALL_{ZSL*} cannot fully reveal the systems' zero-shot disambiguation ability of words.

Similarly, it shows that lately proposed systems tend to outperform the baselines by large margins, varying from 19 to almost 36 F1. Among them, BEM performs the worst on this dataset, 2.2 F1 lower than a similar system, GlossBERT. In contrast, after incorporating both word and sense level context, our system obtains an unprecedented performance on this dataset, being the first system to reach the line of 90 F1 and beating BEM by almost 16 F1. Also, different from SREF and ARES, our systems do not rely on WordNet or SyntagNet sense relation knowledge.

6.4 Cross-lingual All-words WSD

We utilize two multilingual datasets (including French-FR, German-DE, Italian-IT, and Spanish-ES subsets) to evaluate the multilingual transferability of our method. Table 5 presents the performance of some lately proposed systems and ours. For our system, the baseline is trained with the same training data as $SACE_{large+}$ using XLM-RoBERTa-base, while removing all the proposed

Models	ALL_{ZSS} (n=691)	ALL_{ZSS*} (1139)	ALL_{ZSL} (222)	ALL_{ZSL*} (670)
WordNet 1st	24.0	53.9	54.4	84.9
BERT-base	23.5	53.6	54.4	84.9
LMMS	36.7	61.6	74.8	91.7
GlossBERT	37.4	62.0	75.6	91.9
ARES	42.6	65.2	81.1	93.7
SREF	46.1	67.3	82.4	94.2
BEM	48.7	68.9	73.4	91.2
$SACE_{base}$	60.4	76.0	**90.0**	**96.7**
$SACE_{large}$	**66.2**	**79.5**	**90.0**	**96.7**

Table 4: Zero-shot lemma and sense disambiguation. The datasets marked with * include monosemous instances.

components including SlC, WlC, and TaM. For the systems under comparison, all but UKB_{+Syn} utilizes English training data. Also, EWISER and MuLaN further employ SemCor and WNGT as their training data, being the same as $SACE_{mul}$.

It shows that $SACE_{mul}$ has obtained a new state-of-the-art on both the combined dataset and most individual datasets, surpassing its direct baseline by 2.4 F1. In detail, the largest margin, about 5.5 F1 on its Spanish and Italian subset, above the previous best system is acquired on SE15, which covers instances in all POS. This has revealed the overwhelming advantage of $SACE_{mul}$ on disambiguating instances of other POS. In contrast, $SACE_{mul}$ performs 6.5 F1 lower than MuLaN on the Spanish subset of SE13, which only covers noun instances. In a word, $SACE_{mul}$ is more compatible with real cross-lingual scenarios since it has a strong disambiguation ability of words in different POS.

6.5 Analysis

Error Analysis By comparing the disambiguation results of $SACE_{base}$ and its baseline (all factors removed), it is revealed that both systems have correctly disambiguated 5346 instances in ALL while 525 and 339 instances are only correctly disambiguated by $SACE_{base}$ and its baseline respectively. In other words, $SACE_{base}$ has falsely

	SE13				SE15		
	DE	ES	FR	IT	ES	IT	Average
UKB_{+Syn}	76.4	74.1	70.3	72.1	63.4	69.0	71.1
EWISER	80.9	78.8	**83.6**	77.7	69.5	71.8	77.5
MuLaN	82.3	**81.1**	81.6	77.9	69.4	71.8	77.8
ARES	79.6	75.3	81.2	77.0	70.1	71.4	76.2
Baseline	80.5	74.9	80.7	73.6	72.7	74.9	76.3
$SACE_{mul}$	**82.6**	74.6	83.0	**78.1**	**75.6**	**77.3**	**78.7**

Table 5: Multilingual all-words WSD

ID	Score	Sentence
10	/	They belong to a group of ringers who drive every Sunday from ***church*** to ***church*** in a sometimes-exhausting effort to keep the bells sounding in the many belfries of East Anglia.
47	0.969	"The *sound* of *bells* is a net to draw people into the *church*," he says.
19	0.807	Proper English *bells* are started off in "*rounds*, " from the highest-pitched *bell* to the lowest -- a simple descending scale using, in larger *churches*, as many as 12 *bells*.
1	/	Immigration policy under Nicolas Sarkozy was criticized from various aspects a congestion of police, legal and administrative services subjected to a policy of numbers and the compatibility of that policy with the self-proclaimed status of the **country** as the **country** of French human rights.
0	0.384	Is immigration a burden or an opportunity for the economy?
13	0.476	Restraining immigration leads to anaemic growth and harms employment.

Table 6: Two examples of WlC

Synset-1	Synset-2
family.n.01	member.n.01
pilot_burner.n.01	burner.n.01
cruise.v.01	travel.v.01
republican.a.01	democratic.a.02
time.n.02	take.v.02
sport.n.05	player.n.01

Table 7: Related synsets by SlC

predicted 339 examples that are correctly predicted by its baseline. This indicates the proposed methods might have injected excessive noise for the disambiguation of these instances. Therefore, selective exploitation of context for different instances might be beneficial.

The bottom half of table 6 shows an example (*country*) that SACE$_{base}$ falsely predicted. It is shown that the WlC does not manage to retrieve valuable information for disambiguating the word while injecting some irrelevant context.

Case Study Table 6 gives an example of top related sentences (#47 and #19) of a particular sentence (#10) under disambiguation. Here, *church* is falsely predicted when WlC is disabled. It shows that WlC has detected similar sentences in the same document and incorporated valuable context for context embedding learning.

Table 7 provides some examples regarding synsets that are connected by the selective attention layer, indicating its ability of detecting some syntagmatic sense relations and senses of close meaning. The connection is established by using the largest attention score $\alpha(s_{jk}, \hat{s}_p)$ in a batch after filtering self-connection.

7 Conclusion

In this paper, we propose an interactive context exploitation method from both word and sense perspectives in a supervised similarity-based WSD architecture. Experiments on English and cross-lingual all-words WSD datasets verify the effectiveness of our approach, surpassing previous state-of-the-art by large margins. It also shows that the proposed method has an overwhelming advantage of learning few-shot and zero-shot WSD ability. For future work, we intend to utilize reinforcement learning to enhance current interactive WSD by customizing the context exploitation for different instances. The source code is available at: https://github.com/lwmlyy/SACE.

8 Ethics Impact Statement

This paper does not involve the presentation of a new dataset, an NLP application and the utilization of demographic or identity characteristics in formation. For compute time/power, the proposed system requires less GPU amount (1 versus 2 GPUs) and time (10 versus about 70 hours) for training compared with its direct baseline (Blevins and Zettlemoyer, 2020).

Acknowledgments

We thank the anonymous reviewers and Jianzhang Zhang for their insightful comments. This work was supported by the National Natural Science Foundation of China (under Project No. 61375053) and the graduate innovation fund of Shanghai University of Finance and Economics (under Project No. CXJJ-2019-395).

References

Eneko Agirre, Oier López de Lacalle and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. Computational Linguistics,

40(1): 57-84.

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2018. The risk of sub-optimal use of Open Source NLP Software: UKB is inadvert-ently state-of-the-art in knowledge-based WSD. *In Proceedings of Workshop for NLP Open Source Software*, pages 29-33, Melbourne, Australia: Association for Computational Linguistics.

Edoardo Barba, Luigi Procopio, Niccolò Campolungo, Tommaso Pasini and Roberto Navigli. 2020. MuLaN: multilingual label propagation for word sense disambiguation. *In IJCAI-2020*, pages 3837-3844.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. *In COLING 2014*, pages 1591-1600, Dublin, Ireland.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking Through the 80% Glass Ceiling: Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information. *In ACL 2020*, pages 2854-2864. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2020. Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. *In ACL 2020*, pages 1006-1017. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *In ACL 2020*, pages 8440-8451. Association for Computational Linguistics.

Devendra Singh Chaplot and Ruslan Salakhutdinov. 2018. Knowledge-based word sense disambiguation using topic models. *In AAAI 2018*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *In NAACL 2019*, pages 4171-4186, Minneapolis, Minnesota.

Christian Hadiwinoto, Hwee Tou Ng and Wee Chung Gan. Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations. *In EMNLP-IJCNLP 2019*, pages 3507-3512, Hong Kong, China.

Luyao Huang, Chi Sun, Xipeng Qiu and Xuanjing Huang. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. *In EMNLP-IJCNLP 2019*, pages 3507-3512, Hong Kong, China.

Sawan Kumar, Sharmistha Jat, Karan Saxena and Partha Talukdar. 2019. Zero-shot Word Sense Disambiguation using Sense Definition Embeddings. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670-5681, Florence, Italy. Association for Computational Linguistics.

Caterina Lacerra, Michele Bevilacqua, Tommaso Pasini, Roberto Navigli. 2020. CSI: a coarse sense inventory for 85%word sense disambiguation. *In Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence*, Pages 8123-8130. Association for the Advancement of Artificial Intelligence.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. *In SIGDOC '86*, pages 24-26, New York, NY, USA. ACM.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. RoBERTa: a robustly optimized BERT pretraining approach. *In arXiv:1907.11692*.

Daniel Loureiro and Alípio Mário Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full coverage word sense disambiguation. *In ACL 2019*, pages 5682-5691, Florence, Italy.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. *In ACL 2018*, pages 2473-2482, Melbourne, Australia. Association for Computational Linguistics.

Marco Maru, Federico Scozzafava, Federico Martelli, and Roberto Navigli. 2019. SyntagNet: challenging supervised word sense disambiguation with lexical-semantic combinations. *In Proc. Of EMNLP*, pages 3525-3531. Association for Computational Linguistics.

George A. Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G. Thomas. 1994. Using a semantic concordance for sense identification. *In HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro*, New Jersey, March 8-11, 1994.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 41(2): 39-41.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. *In SemEval 2015*,

pages 288-297, Denver, Colorado.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. *In SemEval 2013 *SEM*, pages 222-231, Atlanta, Georgia, USA.

Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. 2007. Semeval-2007 task 07: Coarse grained English all-words task. *In Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 30–35, Prague, Czech Republic.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1-10:69.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. *In Proceedings of SENSEVAL-2*, pages 21-24, Toulouse, France.

Simone Paolo Ponzetto and Roberto Navigli. 2010. Knowledge-rich Word Sense Disambiguation Rivaling Supervised Systems. *In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1522-1531. Association for Computational Linguistics.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. *In EACL 2017*, pages 99-110, Valencia, Spain.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156-1167, Copenhagen, Denmark. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, Roberto Navigli. 2020a. SENSEMBERT: Context-enhanced sense embeddings for multilingual word sense disambiguation. *In AAAI 2020*.

Bianca Scarlini, Tommaso Pasini and Roberto Navigli. 2020b. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. *In the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. *In Senseval-3*, pages 41-43, Barcelona, Spain.

Loïc Vial, Benjamin Lecouteux and Didier Schwab. Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. *In proceedings of the 10th Global WordNet Conference*.

Ming Wang and Yinglin Wang. 2020. A synset relation-enhanced framework with a try-again mechanism for word sense disambiguation. *In the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yinglin Wang, Ming Wang and Hamido Fujita. 2020. Word Sense Disambiguation: A Comprehensive Knowledge Exploitation Framework. *Knowledge-Based Systems*, 10530.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. *In ACL 2010 System Demonstrations*, pages 78-83, Uppsala, Sweden.

Appendix

A Experimental Setting

Computing Infrastructure We use Pytorch deep learning infrastructure along with Transformers and Apex to implement our model. Other required packages can be found in readme.md file in the source code.

Runtime The average training time for SACE_{base}, SACE_{large}, SACE_{large+} and SACE_{mul} is 10 hours, 20 hours, 59 hours and 17 hours, respectively.

Parameters The parameters include those from the pre-trained models such as RoBERTa-base, RoBERTa-large and XLM-RoBERTa-base, and those from the selective attention layer (6 heads * 768/1024 * 768/1024).

Evaluation Metrics We use F1-measure to report the evaluation results. For systems that can provide sense predictions for each lemma, F1-measure is equal to accuracy, which is the number of instances that are correctly predicted by the model. See Navigli, 2009 for details.

β in TaM	0.1, 0.2, 0.3, 0.4, 0.5
WiC local sentences	1, 2, 3, 4, 5
WiC global sentences	1, 2, 3, 4, 5
lr	1e-5, 5e-5, 1e-6, 5e-6
gloss_batch-size	150, 200, 250, 300, 350, 400

Table 1: Hyperparameter bounds and optimal setting

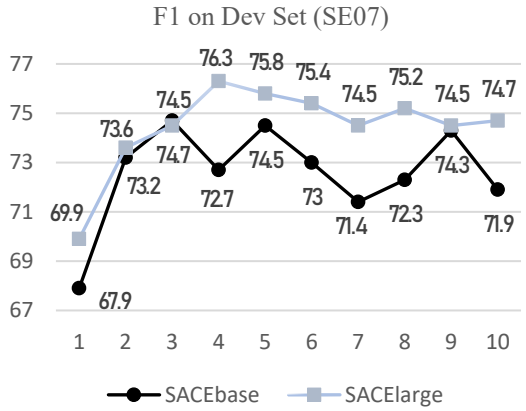

Figure 1: F1 on SE07 of SACE$_{base}$ and SACE$_{large}$

Hyperparameter Search The bounds for each hyperparameter are listed in table 1, with configurations for best performing models underlined. We use the F1-measure on SE07 to select the values. All the details are shown in the source code. For those that have two underlined numbers, they are the best setting for base and large models.

B Experimental Results

In figure 1, we show how SACE$_{base}$ and SACE$_{large}$ perform on SE07 at each epoch during training. It is shown that both systems reach their optimal performance on SE07 at early epoch, 3rd or 4th epoch. This indicates if we utilize the method of early stopping during training, its time efficiency can further be enlarged.