Word Sense Disambiguation with Knowledge-Enhanced and Local Self-Attention-based Extractive Sense Comprehension

Guobiao Zhang[1] , Wenpeng Lu[1,*], Xueping Peng[2] , Shoujin Wang[3] , Baoshuo Kan[1] , Rui Yu[1]
[1]School of Computer, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2]Australian Artificial Intelligence Institute, University of Technology Sydney, Sydney, Australia
[3]Data Science Institute, University of Technology Sydney, Sydney, Australia
guobiao.zhang@foxmail.com, lwp@qlu.edu.cn
{xueping.peng,shoujin.wang}@uts.edu.au
10431200583@stu.qlu.edu.cn, rui.yu1996@foxmail.com

Abstract

Word sense disambiguation (WSD), identifying the most suitable meaning of ambiguous words in the given contexts according to a predefined sense inventory, is one of the most classical and challenging tasks in natural language processing. Reformulating WSD as a text span extraction task is an effective approach, which accepts a sentence context of an ambiguous word together with all definitions of its candidate senses simultaneously, and requires to extract the text span corresponding with the right sense. However, the approach merely depends on a short definition to learn sense representation, which neglects abundant semantic knowledge from related senses and leads to data-inefficient learning and suboptimal WSD performance. To address the limitations, we propose a novel WSD method with **K**nowledge-**E**nhanced and **L**ocal self-attention-based **E**xtractive **S**ense **C**omprehension (KELESC). Specifically, a knowledge-enhanced method is proposed to enrich semantic representation by incorporating additional examples and definitions of the related senses in WordNet. Then, in order to avoid the huge computing complexity induced by the additional information, a local self-attention mechanism is utilized to constrain attention to be local, which allows longer input texts without large-scale computing burdens. Extensive experimental results demonstrate that KELESC achieves better performance than baseline models on public benchmark datasets.[1]

1 Introduction

Word sense disambiguation (WSD) is to identify a proper sense with an ambiguous word in a given context according to a predefined sense inventory, which is one of the most typical and challenging tasks in natural language processing (NLP) and play a critical role for human language understanding (Conia and Navigli, 2021). For instance, the noun word *plant* conveys different senses in *industrial plant* and *plant seeds*. WSD has been able to determine accurate meanings of ambiguous words, which is beneficial to a variety of downstream NLP applications, such as machine translation, information extraction and retrieval (Song et al., 2021; Pasini and Navigli, 2020).

In recent years, with the rapid development of deep learning, the performance of WSD with neural networks-based methods has great improvement. The early neural networks-based models have cast WSD as a multi-label classification task, which disambiguated all polysemous words with a unified classier (Kågebäck and Salomonsson, 2016; Raganato et al., 2017a). However, these models have focused on modeling contexts containing ambiguous words from sense-labeled training data, which ignored the rich semantic knowledge in lexical resources, such as WordNet and BabelNet (Navigli et al., 2021), and resulted in their inability to outperform the traditional *word expert* supervised methods (Song et al., 2021).

Due to the semantic knowledge in a lexical dictionary including sense definitions (glosses), examples, relations, etc., defined by professional lexicographers, which is essential and valuable for WSD, some works (Banerjee and Pedersen, 2002; Basile et al., 2014) have attempted to integrate gloss information into neural WSD models in order to leverage the lexical knowledge. GAS (Luo et al., 2018) has incorporated glosses into WSD, which jointly encoded glosses and contexts, and captured their relations with a memory network. GlossBERT (Huang et al., 2019) has utilized glosses in WordNet together with the annotated data to construct *context-gloss* pairs, reformulated WSD as a text matching task. BEM (Blevins and Zettlemoyer, 2020) has been a bi-encoder method that encodes the target word and candidate glosses inde-

*Corresponding author
[1]The source code of this paper can be obtained from https://github.com/Stubborn-z/KELESC

pendently and optimizes the encoders in the same representation space. Although these works considered the gloss information in WordNet, they neglected to explore the knowledge contained in semantic relations such as hypernyms. Therefore, EWISER (Bevilacqua and Navigli, 2020) has been proposed to enhance WSD by integrated synset embeddings and semantic relations including hypernyms and hyponyms. ESR (Song et al., 2021) has further enhanced sense representations by incorporating synonyms, example sentences and sense glosses of hypernyms.

Although the methods mentioned above have achieved great successes, they have treated WSD as multi-label classification or text matching tasks, which focused on modeling the relations between a context and each specified candidate sense. None of them considers all candidate senses of an ambiguous word simultaneously, which is not consistent with human behaviors, as humans always justify the right sense by comparing all possible senses with the context. In order to simulate the cognitive process of human, ESC (Barba et al., 2021) has reformulated WSD as a text span extraction task, called extractive sense comprehension, which accepted a context of an ambiguous word together with all definitions of its candidate senses. Although ESC demonstrated the superiority over the competitive methods, it merely relied on a short definition (gloss) to represent a sense, which was insufficient to learn an ideal sense representation and inevitably hinder the improvement of WSD performance.

To address the above-mentioned limitations, we propose a novel WSD method with **K**nowledge-**E**nhanced and **L**ocal self-attention-based **E**xtractive **S**ense **C**omprehension (KELESC), inspired by ESR (Song et al., 2021). Specifically, a knowledge-enhanced method is proposed to enrich semantic representation by incorporating additional examples and definitions of the related senses in WordNet. Then, in order to avoid the huge computing complexity induced by the additional information, a local self-attention mechanism is utilized to constrain attentions to be local, which allows longer input texts without large-scale computing burdens (Beltagy et al., 2020; Manakul and Gales, 2021). Extensive experiments have been conducted to verify the effectiveness of the proposed model on public benchmark datasets. In summary, this paper makes the following contributions:

- We propose a novel end-to-end WSD model with Knowledge-Enhanced and Local self-attention-based Extractive Sense Comprehension (KELESC). The model reformulates WSD as a text extraction task, fully utilizes lexical knowledge to enhance sense representation, and considers all candidate senses simultaneously instead of one by one to identify the right sense.

- We devise a knowledge enhancement method to enrich semantic representation by incorporating additional sense information of related senses in WordNet. Besides, we exploit a local self-attention mechanism to reduce the computing burden of training the model.

- Extensive experiments are conducted on public datasets to demonstrate the superiority of our proposed model on all-words English WSD tasks by making comparisons with the baseline models.

2 Related Work

The existing works on WSD can be categorized into three groups: knowledge-based, supervised and neural-based methods.

2.1 Knowledge-based WSD methods

These methods focus on leveraging semantic knowledge contained in lexical resources to identify the right sense (Luo et al., 2018). They mainly exploit two kinds of knowledge: sense definitions and structure of semantic network. For sense definitions' knowledge, Lesk algorithm and its variants are the typical works, which select the right sense according to the overlap of contexts and sense definitions (Lesk, 1986). For structure knowledge of semantic network, Personalized PageRank (Agirre et al., 2014; Scozzafava et al., 2020), BabelNet (Navigli and Ponzetto, 2012) and structural semantic interconnections (Navigli and Velardi, 2005) are the representative methods, which construct semantic graphs with senses and their relations, and utilize graph-based algorithms to choose the most important sense as the right one. With the support of lexical knowledge, knowledge-based methods achieve satisfied WSD coverage while their accuracy usually is worse than the others.

2.2 Traditional supervised WSD methods

These methods utilize manually feature engineering to train a special classifier for each polysemous word, i.e., *word expert*. IMS system first proposes instance and feature extractors to extract instances and their features, then trains an independent classifier for each word type on sense-annotated SemCor corpus (Zhong and Ng, 2010). Iacobacci et al. (2016) investigate how word embeddings has been utilized in WSD, which combines word embeddings and traditional manual features to enhance the original IMS system. Although these traditional supervised methods show better performance on WSD, they are confused by the manually engineered features and sense-annotated training dataset. Besides, they train a dedicated classifier for each word, which are hard to be applied on all-word WSD tasks.

2.3 Neural-based WSD methods

These approaches usually train a unified classifier based on neural networks to disambiguate all of the polysemous words. The early neural-based models mainly focus on modeling the relations of sentence context and sense labels contained in training datasets. For example, (Kågebäck and Salomonsson, 2016) and (Raganato et al., 2017a) employ bidirectional LSTM and encoder-decoder architecture to train unified models for all-word WSD tasks. However, they neglect to utilize the valuable semantic knowledge contained in lexical resources such as WordNet. Thus, The GAS model attempts to incorporate gloss information into an end-to-end WSD model (Luo et al., 2018). GlossBERT (Huang et al., 2019) also leverages glosses in WordNet to construct *context-gloss* pairs, reformulates WSD as a text matching task to model the matching relations of sense glosses and the contexts of ambiguous words. BEM (Blevins and Zettlemoyer, 2020) proposes a jointly optimized bi-encoder (the context encoder and the gloss encoder), which encode the context and sense glosses, and choose the nearest sense with the context according to gloss and context embeddings. However, these methods merely utilize the gloss information in WordNet, which still ignore the semantic relation knowledge such as hypernyms. Therefore, EWISER (Bevilacqua and Navigli, 2020) is proposed to integrate sense embeddings together with hypernyms and hyponyms relations to enhance WSD performance. And, ESR (Song et al., 2021) further incorporates synonyms, example sentences and sense glosses of hypernyms to enhance sense representations. All methods mentioned above focus on modeling relations between a context and each specified candidate sense individually, while human usually determines the sense by comparing all possible senses with a context simultaneously. This means that there are still some room to improve the neural-based WSD methods. In order to simulate the cognitive progress of human, that is, to comparing all candidate senses simultaneously, ESC (Barba et al., 2021) reformulates WSD as a text span extraction task, which accepts a context of an ambiguous word together with the definitions of all possible senses, and choose the text span of the right sense by comparing all sense definitions at once. ESC has shown the superiority on WSD task, however, it merely utilizes a short definition to learn a sense, which is insufficient.

3 Methodology

In this section, we first give the task definition. Then, we detail our proposed model, **K**nowledge-**E**nhanced and **L**ocal self-attention-based **E**xtractive **S**ense **C**omprehension (KELESC) for WSD.

3.1 Task Definition

Given the context with target word with glosses, example sentences and hypernym glosses of all candidate senses, the task of the paper is to identify the text span that indicates the right sense. Specifically, we represent the context of target word $\hat{w}$ as $C = \{w_1^c, \cdots, w_m^c\}$, where m is the number of words in the context. For the k-th candidate sense of the target word $\hat{w}$, its gloss, example sentence and hypernym gloss are represented as $G^k = \{w_1^{gk}, \cdots, w_{|gk|}^{gk}\}$, $ES^k = \{w_1^{ek}, \cdots, w_{|ek|}^{ek}\}$, and $HG^k = \{w_1^{hk}, \cdots, w_{|hk|}^{hk}\}$. $|gk|$, $|ek|$ and $|hk|$ indicate their lengths. Given the concatenation of the context C and the information G, ES, HG of all candidate senses, our model will identify the interval $[i_{cor}, j_{cor}]$, which indicates the start and end positions of the text span corresponding with the gloss, example sentence and hypernym gloss of the right sense of $\hat{w}$.

3.2 Model Architecture

The overall structure of KELESC model is shown in Figure 1. KELESC model consists of three core modules: (1) a knowledge enhancement module,

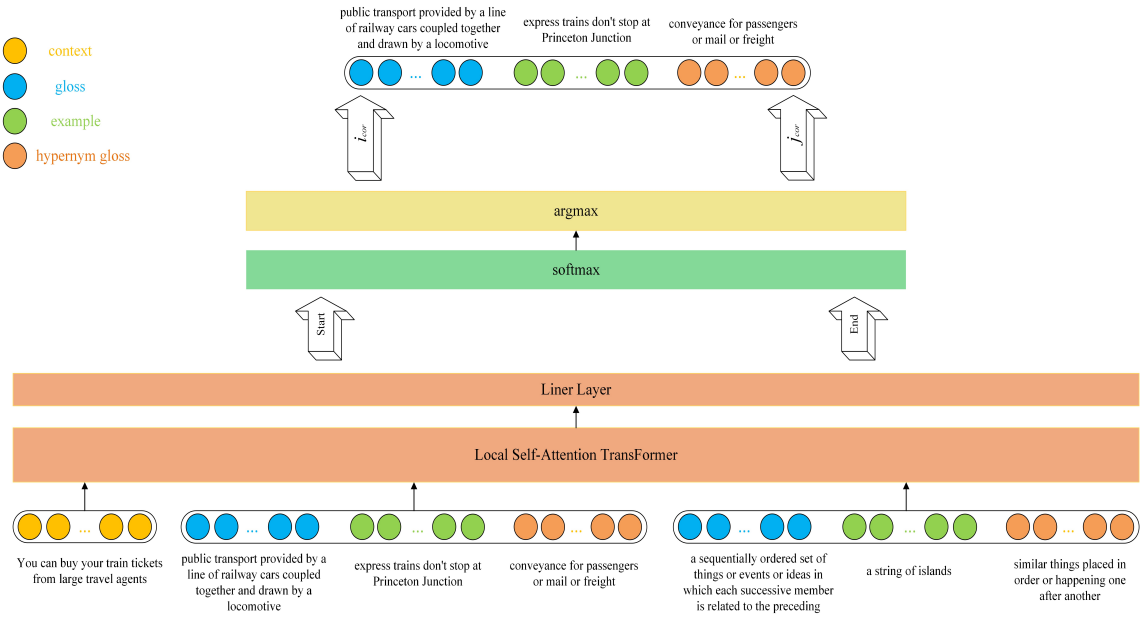

Figure 1: Overview structure of KELESC model. Concatenate the context of target word together with the gloss, example sentence, hypernym gloss of each candidate sense as the input of our model. $Start$ and End represent the logits for each word, which indicates whether it is the start or end of the text span of the right sense of the target word, respectively. i_{cor} and j_{cor} is the start and end indices of the correct sense, respectively.

Context Sentence		You can buy your *train* tickets from large travel agents.
Sense#1	Gloss	public transport provided by a line of railway cars coupled together and drawn by a locomotive.
	Example sentence	express trains don't stop at Princeton Junction.
	Hypernym gloss	conveyance for passengers or mail or freight.
Sense#2	Gloss	a sequentially ordered set of things or events or ideas in which each successive member is related to the preceding.
	Example sentence	train of mourners.
	Hypernym gloss	similar things placed in order or happening one after another.
Context and enhanced knowledge		You can buy your train tickets from large travel agents. public transport provided by a line of railway cars coupled together and drawn by a locomotive. express trains don't stop at Princeton Junction. a sequentially ordered set of things or events or ideas in which each successive member is related to the preceding. a string of islands. similar things placed in order or happening one after another.

Table 1: An example of knowledge enhancement of the target word *train*.

which utilizes gloss, example sentence and hypernym gloss to enhance the representation of each candidate sense, (2) a local self-attention transformer, which encodes the entire input texts with local self-attention transformer, (3) a span prediction module, which extracts the text span with the highest probability of expressing the correct sense of the target word.

3.2.1 Knowledge Enhancement

Recent studies have shown that lexical knowledge in WordNet is essential and valuable for accurate sense representation learning (Song et al., 2021). To this end, KELESC model devises a knowledge enhancement module to explore and integrate the richer lexical knowledge. As shown in Table 1, for each candidate sense of a target word, the module collects its gloss, example sentence and hypernym glosses together to enrich sense representation. Specifically, the gloss is a short definition of the current sense in WordNet. The example sentence is a sentence instance that conveys the corresponding sense. The hypernym gloss refers to the sense definition of the hypernym synsets of the current sense, which describes high-level semantic information. The original context and all candidate senses of the the target word with its glosses, example sentences and hypernym glosses are concatenated together, which is fed into our model.

3.2.2 Local Self-Attention Transformer

In order to effectively encode the input text, we adopt the pre-trained transformer-based model, i.e., BART$_{large}$, as it works well on long sequence modeling and comprehension tasks (Lewis et al., 2020; Beltagy et al., 2020). As shown in Figure 1 and Table 1, we concatenate the gloss, example sentence and hypernym gloss of each candidate sense together for a target word, marked as $A = \{G^1, ES^1, HG^1, \cdots, G^{|s|}, ES^{|s|}, HG^{|s|}\}$, where $|s|$ is the number of candidate senses of the target word. The context sentence and enhanced knowledge are fed into the transformer as the input. There could be some exceptions: if there is no example sentence in WordNet, we ignore it; if there are multiple example sentences, we only select the first one.

Specifically, we use the tags $< s >$ and $< /s >$ to surround the entire input sequence. The context sentence C and the enhanced lexical knowledge A are segmented by the special symbol $< /d >$ and the target word $\hat{w}$ is surrounded by $< t >$ and $< /t >$. The entire input of the transformer is as follows:

$$
\begin{aligned}
input = & < s > w_1^c \cdots < t > \hat{w} < /t > \cdots w_m^c \\
& < /d > w_1^{g1} \cdots w_{|g1|}^{g1} w_1^{e1} \cdots w_{|e1|}^{e1} w_1^{h1} \cdots w_{|h1|}^{h1} \cdots \\
& w_1^{gn} \cdots w_{|gn|}^{gn} w_1^{en} \cdots w_{|en|}^{en} w_1^{hn} \cdots w_{|hn|}^{hn} < /s >
\end{aligned}
$$

where $input$ is tokenized as $T = \{t_1, t_2, \cdots, t_n\}$, and n denotes the length of input token sequence.

The sense representation is enriched by our proposed knowledge enhancement module. However, it inevitably results in the longer sequence of input text, which the memory requirement and computing complexity of the transformer-based model could be quadratic with the length of input sequence. It increases the huge burden during the model training. To alleviate this problem, we introduce a local self-attention mechanism proposed by (Manakul and Gales, 2021). It is noteworthy that the mechanism in KELESC focuses on the encoder part. Our local self-attention transformer-based module adopts a fixed window $\widehat{w}$ around each token which only focuses on the ones lying in the window on each side. As shown in Figure 2, we set a window size to 1 as a toy example to show the local self-attention mechanism in an encoding layer. The outputs of encoding layer in local self-attention transformer are calculated as bellow:

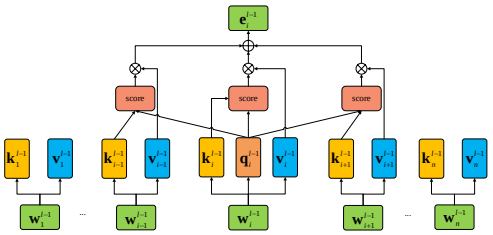

Figure 2: Local self-attention with a window size $\widehat{w}$ of 1, $\mathbf{w}_i^{l-1}$ represents the embedding of the i-th token w_i generated by the previous encoding layer $(l-1)$. $\mathbf{k}_i^{l-1}$, $\mathbf{v}_i^{l-1}$ and $\mathbf{q}_i^{l-1}$ represent the vector of key, value and query, respectively. $\mathbf{w}_i^l$ is the embedding of w_i obtained with local self-attention mechanism in the current layer (l).

$$
\mathbf{w}_i^l = \text{sum}\left(\text{softmax}\left(\frac{\mathbf{Q}_i^{l-1}{\mathbf{K}_i^{l-1}}^\top}{\sqrt{\mathbf{d}_k}}\right)\mathbf{v}_i^{l-1}\right)
$$

where $\mathbf{Q}_i^{l-1}=[\mathbf{q}_i^{l-1}]_{2*\widehat{w}+1}$ is the local query matrix, $\mathbf{K}_i^{l-1}=[\mathbf{k}_{i-\widehat{w}}^{l-1},\cdots,\mathbf{k}_{i-1}^{l-1},\mathbf{k}_i^{l-1},\mathbf{k}_{i+1}^{l-1},\cdots,\mathbf{k}_{i+\widehat{w}}^{l-1}]$ is the local key matrix, and $\mathbf{v}_i^{l-1}=[\mathbf{v}_{i-\widehat{w}}^{l-1},\cdots,\mathbf{v}_{i-1}^{l-1},\mathbf{v}_i^{l-1},\mathbf{v}_{i+1}^{l-1},\cdots,\mathbf{v}_{i+\widehat{w}}^{l-1}]$ is the local value matrix. $\mathbf{d}_k$ is the dimension of the embedding vector. By stacking multiple layers of this local self-attention transformer, a large receptive range will be obtained, in which the top layer can access all input positions and has the ability to build a representation containing the whole input information. In this way, the memory requirement and computational complexity of the model increases linearly with the length of the input sequence. This reduce the training burden greatly.

After passing through the last layer of the local self-attention transformer, we obtain the hidden states representation of the final layer:

$$
\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n = \text{Transformer}(T),
$$

where $\mathbf{h} \in \mathbb{R}^d$, d represents the dimension of each hidden state. All these representations of hidden units form the final matrix $\mathbf{H}$, i.e., $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n] \in \mathbb{R}^{d \times n}$, which is further transferred to a liner layer:

$$
\mathbf{Z} = \mathbf{W}^\top \mathbf{H} + \mathbf{b},
$$

where $\mathbf{W} \in \mathbb{R}^{d \times 2}$ and $\mathbf{b} \in \mathbb{R}^2$ are trainable parameters.

3.2.3 Loss Function

For the target word $\hat{w}$, the correct start and end positions are represented as:

$$
\begin{aligned}
\textbf{Start} &= [\mathbf{Z}_{11} \ \cdots \ \mathbf{Z}_{1n}], \\
\textbf{End} &= [\mathbf{Z}_{21} \ \cdots \ \mathbf{Z}_{2n}],
\end{aligned}
$$

where **Start** and **End** indicate the logits for each token, denoting whether it is the start or the end of the text span corresponding with the correct sense of the target word $\hat{w}$, respectively.

We add the two cross-entropy loss functions for the start and end positions to train the model:

$$\mathcal{L}_s = -\textbf{Start}_{i_{cor}} + \log \sum_{v=1}^{l} \exp(\textbf{Start}_v),$$

$$\mathcal{L}_e = -\textbf{End}_{j_{cor}} + \log \sum_{v=1}^{l} \exp(\textbf{End}_v),$$

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_e.$$

where $\textbf{Start}_{i_{cor}}$ and $\textbf{End}_{j_{cor}}$ are the scores correspond to the correct start and end positions.

3.2.4 Prediction

Following the work of Barba et al. (2021), our model outputs a pair of (i_{cor}, j_{cor}), which indicate the start and end positions of the right sense in the input text. To assure the pair is exactly matched with the text span of any sense, the model selects its output by comparing their probability. First, the logits **Start** and **End** are fed into softmax to obtain the probability distribution. Then, we perform a product operation on the probability distributions of the start and end positions to generate the probability of pair (i_{cor}, j_{cor}) that starts at i and ends at j :

$$\mathbf{P}(i_{cor}) = \text{softmax}(\textbf{Start}),$$

$$\mathbf{P}(j_{cor}) = \text{softmax}(\textbf{End}),$$

$$\mathbf{P}(i_{cor}, j_{cor}) = \mathbf{P}(i_{cor}) \times \mathbf{P}(j_{cor}),$$

where $\mathbf{P}(i_{cor})$ and $\mathbf{P}(j_{cor})$ indicates the probability that i_{cor} is the correct start position or the j_{cor} is the correct end position, respectively. $\mathbf{P}(i_{cor}, j_{cor})$ represents the probability of span that starts at i_{cor} and ends at j_{cor} across all the other spans in the input T.

Finally, the model outputs the pair with max probability, as follows:

$$output = \text{argmax}\ \mathbf{P}(i_{cor}, j_{cor}).$$

4 Experiment

4.1 Datasets

Following the existing works, we evaluate our proposed model on English all-words WSD task through a public unified evaluation framework (Raganato et al., 2017b). SemCor is selected as our training corpus (Miller et al., 1994), the smallest SemEval-2007 dataset (**SE07**) (Pradhan et al., 2007) is chosen as development set, and the rest are used as test datasets, including Senseval-2 (**SE2**) (Edmonds and Cotton, 2001), Senseval-3 (**SE3**) (Snyder and Palmer, 2004), SemEval-2013 (**SE13**) (Navigli et al., 2013), SemEval-2015 (**SE15**) (Moro and Navigli, 2015). The four test datasets are concatenated together marked as **ALL**. F1 score is used as the evaluation measure to report the performance.

4.2 Baselines

According to the exploitation of lexical knowledge, we categorize the baselines into three groups.

The first group includes the methods without any lexical knowledge, which merely rely on the training data and don't utilize any lexical knowledge, such as glosses and hypernyms. In this group, we first consider the MFS baseline, which simply adopts the most frequent sense in training datasets as the right sense of each word. Then, BiLSTM (Kågebäck and Salomonsson, 2016) is adopted, which is a early neural-based method and trains bidirectional LSTM to obtain a unified model for all-word WSD task. Besides, we select BERT_{base} (Devlin et al., 2019) as another baseline, which learns a linear classifier based on frozen BERT representations .

The second group involves the neural-based methods which exploit glosses of candidate senses, i.e., GAS (Luo et al., 2018), LMMS (Loureiro and Jorge, 2019), GlossBERT (Huang et al., 2019), ARES (Scarlini et al., 2020), BEM (Blevins and Zettlemoyer, 2020), ESCHER (Barba et al., 2021). These models utilize glosses to represent the corresponding senses. GAS is the first model to incorporate glosses into neural-based WSD, which jointly optimizes the representations of contexts and glosses of ambiguous words. Both LMMS and ARES are the nearest neighbors methods (k-NN), which identify the right sense according to the similarity between context and sense representation. LMMS generates sense representation from sense-annotated data, which is further enhanced with sense glosses in WordNet. ARES generates sense representation by leveraging the contexts in SemCor and the glosses in WordNet, which is further enriched with synset embeddings. Gloss-

Model	Dev Set	Test Sets				Concatenation of all Datasets				
	SE07	SE2	SE3	SE13	SE15	Nouns	Verbs	Adj.	Adv.	ALL
Baselines without any lexical knowledge										
MFS baseline	54.5	65.6	66.0	63.8	67.1	67.7	49.8	73.1	80.5	65.5
BiLSTM	-	71.1	68.4	64.8	68.3	69.5	55.9	76.2	82.4	68.4
BERT$_{base}$	68.6	75.9	74.4	70.6	75.2	75.7	63.7	78.0	85.8	73.7
Baselines with gloss information										
GAS*	-	72.0	70.0	66.7	71.6	71.7	57.4	76.5	83.5	70.1
LMMS*	68.1	76.3	75.6	75.1	77.0	-	-	-	-	76.8
GlossBERT*	72.5	77.7	75.2	76.1	80.4	79.8	67.1	79.6	87.4	77.0
ARES*	71.0	78.0	77.1	77.3	83.2	80.6	68.3	80.5	83.5	77.9
BEM*	74.5	79.4	77.4	79.7	81.7	81.4	68.5	83.0	**87.9**	79.0
ESCHER*	76.3	81.7	77.8	82.2	**83.2**	83.9	69.3	83.8	86.7	80.7
Baselines with gloss and other knowledge										
EWISER†	71.0	78.9	**78.4**	78.9	79.3	81.7	66.3	81.2	85.8	78.3
ESR$^\dagger_{base}$	75.4	80.6	78.2	79.8	82.8	82.5	**69.5**	82.5	87.3	79.8
KELESC†	**76.7**	**82.2**	78.1	**82.2**	83.0	**84.3**	69.4	**84.0**	86.7	**81.2**

Table 2: Comparison of F1 scores (%) on the English all-words WSD task. $*$ indicates that the model exploit glosses of candidate sense, † indicates that the model utilizes sense glosses as well as other knowledge in WordNet. We bold the best score for each column.

BERT reformulates WSD as a text matching task, which evaluates the matching degree between sense glosses and the contexts of ambiguous words to identify the right sense. BEM utilizes two encoders to represent contexts and candidate senses independently, and identify the right sense by finding the nearest sense embedding for the context embedding. ESCHER reformulates WSD as a text span extraction task, which is optimized to extract the text span of the gloss expressing the right sense when the model is fed with a sentence containing an ambiguous word and all its candidate sense glosses.

The third group consist of the methods which exploit more lexical knowledge, such as hypernyms, example sentence and gloss information. EWISER (Bevilacqua and Navigli, 2020) learns sense information from WordNet, which considers semantic relations between senses, such as hypernyms and hyponyms. ESR$_{base}$ (Song et al., 2021) further enhance sense representations by incorporating synonyms, example phrases or sentence and sense gloss of hypernyms.

4.3 Parameter Settings

We select BART$_{large}$ (Lewis et al., 2020) as our based model, whose encoder and decoder have 12 layers, respectively. In the encoder, the original self-attention is replaced by the local self-attention with a window size of 512 to avoid the huge computing complexity induced by the additional information. The optimizer is RAdam (Liu et al., 2020). Besides, we set batch size to 900 tokens, learning rate to 2e-6, and weight decay to 0.01. F1 score is calculated on validation dataset every 2000 steps, and stop training is applied if the model does not improve in 15 successive times. Our model is trained on one A100 GPU, which takes about 10 hours.

4.4 Overall Results

We evaluate the performance of our method by comparing it with the baselines. The overall results on English all-words WSD task are summarized in Table 2. According to the table, we have several observations.

First, the methods exploiting gloss information (i.e., the second group) usually outperform the methods without any lexical knowledge (i.e., the first group), except for BERT$_{base}$ and GAS. This demonstrates that gloss information is critical and essential for WSD, which is beneficial for learning better sense representations. Besides, the exception of BERT$_{base}$ and GAS may be caused that GAS is realized with BiLSTM whose learning ability is much weaker than BERT.

Second, the methods exploiting gloss and other lexical knowledge (i.e., the third group) outperform most of the methods in second group and all methods in first group. This is because that the methods in third group incorporate more lexical knowledge, such as hypernyms and example sentence, to enhance neural-based models, and the enhanced knowledge is useful for sense representation, which can further improve WSD performance.

Third, our model consistently outperform all competitive baseline methods on **ALL**. Our model also achieves the best performance on **SE07**, **SE2**, **SE13**, **Nouns** and **Adj.**. The reason for the superiority of our model is two-fold. One is that our model reformulates WSD as a text extraction task, which can accept and perceive all candidate sense, simultaneously. The other is that our model enhances sense representation with more lexical knowledge including hypernyms and examples. Among the baselines, ESCHER is the most similar to our model. Both models reformulate WSD as a text span extraction task. However, our model is better than ESCHER and increases F1 score by 0.5% on ALL. This is because that our model utilize more lexical knowledge than ESCHER.

4.5 Ablation Study

To evaluate the effectiveness of different lexical knowledge in our model, i.e., example sentence and hypernym gloss, we conduct ablation studies by removing them one by one to observe the change of overall performance.

Reserved Lexical Knowledge	**ALL** (%)
Example sentence + hypernym gloss	81.2
Example sentence	81.0
Hypernym gloss	80.8

Table 3: Comparison of ablated models on ALL.

As shown in Table 3, if we remove the gloss of hypernyms from our model, this leads to 0.2% drop from 81.2% to 81.0%. And, if we remove the example sentence, there is 0.4% drop from 81.2% to 80.8%. The above results indicate that the role of example sentence is more important than the gloss of hypernyms in our model. One explanation is that the example sentence is more semantically representative for the target sense.

4.6 Window Size in Local Self-Attention

In order to evaluate model training complexity and effectiveness, we employ different configurations of local self-attention. At the same time, we compare local self-attention with self-attention. The results are shown in Table 4:

Model	**Window**	**GiB**	**ALL** (%)
Self-attention	Full	31.2	80.8
Local self-attention	128	18.0	80.3
Local self-attention	256	20.8	80.8
Local self-attention	512	23.4	81.2

Table 4: Comparison of memory requirement and performance with different window sizes.

In Table 4, we observe that local self-attention mechanism can significantly reduce the memory usage, which is beneficial for accelerating training speed and reducing the training burden. Moreover, we find that the performance with window size of 128 is 80.3, which is 0.5% lower than the original self-attention, which is due to the fact that the window is too small and the model cannot effectively model the sense representation.

5 Conclusion and Future Work

In this paper, we proposed a novel WSD method with knowledge-enhanced and local self-attention-based extractive sense comprehension. Specifically, a knowledge-enhanced method was devised to enrich semantic representation by incorporating additional examples and definitions of the related senses in WordNet. Then, in order to avoid the huge computing complexity induced by the additional information, a local self-attention mechanism was utilized to constrain attentions to be local, which allowed longer input texts without large-scale computing burdens. Extensive experimental results had demonstrated the effectiveness of the proposed model on public benchmark datasets.

Although our model achieved better performance, it still could be improved. Currently, we utilized example sentence and hypernym gloss. There are many other unexplored semantic relations in WordNet and BabelNet. We leave it as future work to explore more semantic relations in more lexical resources to further enhance WSD performance. Besides, a detailed qualitative analysis on rare senses and frequent ones should be considered. We will attempt to evaluate the performance on

different situations to further enhance our model.

Acknowledgements

The work is partly supported by National Nature Science Foundation of China (61502259), and Key Program of Science and Technology of Shandong (2020CXGC010901), and Studio Project of the Research Leader in Jinan (2019GXRC062).

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.

Satanjeev Banerjee and Ted Pedersen. 2002. An adapted Lesk algorithm for word sense disambiguation using wordnet. In *Proceeding of the International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145.

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4661–4672.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 1591–1600.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017.

Simone Conia and Roberto Navigli. 2021. Framing word sense disambiguation as a multi-label problem for model-agnostic knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3269–3275.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Philip Edmonds and Scott Cotton. 2001. Senseval-2: overview. In *Proceedings of the 2nd International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuan-Jing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3509–3514.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 897–907.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional LSTM. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon*, pages 51–56.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th International Conference on Systems Documentation*, pages 24–26.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *Proceedings of the 2020 International Conference on Learning Representations*.

Daniel Loureiro and Alipio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2473–2482.

Potsawee Manakul and Mark Gales. 2021. Long-Span summarization via local attention and content selection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 6026–6041.

George A Miller, Martin Chodorow, Shari Landes, Claudia Leacock, and Robert G Thomas. 1994. Using a semantic concordance for sense identification. In *Proceedings of the Workshop on Human Language Technology*.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, pages 288–297.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of BabelNet: A survey. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 4559–4567.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation*, pages 222–231.

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

Roberto Navigli and Paola Velardi. 2005. Structural semantic interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1086.

Tommaso Pasini and Roberto Navigli. 2020. Train-O-Matic: Supervised word sense disambiguation with no (manual) effort. *Artificial Intelligence*, 279:103215.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017a. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1156–1167.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017b. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 99–110.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3528–3539.

Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and Roberto Navigli. 2020. Personalized pagerank with syntagmatic information for multilingual word sense disambiguation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–46.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43.

Yang Song, Xin Cai Ong, Hwee Tou Ng, and Qian Lin. 2021. Improved word sense disambiguation with enhanced sense representations. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4311–4320.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 78–83.