Leveraging Gloss Knowledge in Neural Word Sense Disambiguation by Hierarchical Co-Attention

Fuli Luo[1], Tianyu Liu[1], Zexue He[2], Qiaolin Xia[1], Zhifang Sui[1] and Baobao Chang[1]
[1]Key Lab of Computational Linguistics, School of EECS, Peking University
[2]Computer Science and Technology, College of IST, Beijing Normal University
{luofuli, tianyu0421, xql, chbb, szf}@pku.edu.cn
zexueh@mail.bnu.edu.cn

Abstract

The goal of Word Sense Disambiguation (WSD) is to identify the correct meaning of a word in the particular context. Traditional supervised methods only use labeled data (context), while missing rich lexical knowledge such as the gloss which defines the meaning of a word sense. Recent studies have shown that incorporating glosses into neural networks for WSD has made significant improvement. However, the previous models usually build the context representation and gloss representation *separately*. In this paper, we find that the learning for the context and gloss representation can benefit from each other. Gloss can help to highlight the important words in the context, thus building a better context representation. Context can also help to locate the key words in the gloss of the correct word sense. Therefore, we introduce a co-attention mechanism to generate co-dependent representations for the context and gloss. Furthermore, in order to capture both word-level and sentence-level information, we extend the attention mechanism in a hierarchical fashion. Experimental results show that our model achieves the state-of-the-art results on several standard English all-words WSD test datasets.

1 Introduction

Word Sense Disambiguation (WSD) is a crucial task and long-standing problem in Natural Language Processing (NLP). Previous researches mainly exploit two kinds of resources. Knowledge-based methods (Lesk, 1986; Moro et al., 2014; Basile et al., 2014) exploit the lexical knowledge like gloss to infer the correct senses of ambiguous words in the context. However, supervised feature-based methods (Zhi and Ng, 2010; Iacobacci et al., 2016) and neural-based methods (Kågebäck and Salomonsson, 2016; Raganato et al., 2017a) usually use labeled data to train one or more classifiers.

Context	As they often **play** football together, they know each other quite well
Glosses	g_1: participate in games or sports
	g_2: perform music on an instrument
	g_3: behave in a certain way

Table 1: An example of the context and three glosses of different senses according to the target word *"**play**"*. It shows that the words *"games/sports"* in the gloss g_1 can help to highlight the important words *"football"* in the context and ignore the words *"know each other"* which are useless for distinguishing the sense of word *"play"*. Meanwhile, the context can potentially help to stress on the words *"games/sports"* of the gloss g_1 which is actually the correct sense for the target word.

Although both lexical knowledge (especially gloss) and labeled data are of great help for WSD, previous supervised methods rarely take the integration of knowledge into consideration. To the best of our knowledge, Luo et al. (2018) are the first to directly incorporate the gloss knowledge from WordNet into a unified neural network for WSD. This model *separately* builds the context representation and the gloss representation as distributed vectors and later calculates their similarity in a memory network. However, we find that the learning of the representations of the context and gloss can contribute to each other. We use an example to illustrate our ideas. Table 1 shows that the red words are more important than the blue words when distinguishing the sense of the target word. In other words, we should pay more attention to the words which can "overlap" between the context and the gloss when generating the representations of context and gloss. Therefore, we introduce a co-attention mechanism to model the mutual influence between the representations of context and gloss.

Moreover, we find that both word-level and sentence-level information are crucial to WSD. As

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1402–1411
Brussels, Belgium, October 31 - November 4, 2018.

shown in Table 1, the local word *"football"* is crucial for distinguishing the sense of word *"play"*. However, in more complex sentences such as *"Investors played it carefully for maximum advantage"* [1], sentence-level information is necessary. Therefore, we extend the co-attention model in a hierarchical fashion to capture both the word-level and sentence-level semantic information.

The main contributions are listed as follows.

- We propose a novel way to integrate gloss knowledge into a neural network for WSD via a co-attention mechanism in order to build better representations of context and gloss. In this way, our model can benefit from both labeled data and lexical knowledge.
- We further extend the attention mechanism into a hierarchical architecture, since both word-level and sentence-level information are crucial to disambiguating the word sense.
- We conduct a series of experiments, which show that our models outperform the state-of-the-art systems on several standard English all-words WSD test datasets.

2 Related work

Lexical knowledge is a fundamental component of Word Sense Disambiguation and provides rich resources which are essential to associate senses with words (Navigli, 2009). Unsupervised knowledge-based methods have shown the effectiveness of textual knowledge such as gloss (Lesk, 1986; Basile et al., 2014) and the structural knowledge (Moro et al., 2014; Agirre et al., 2014) of the lexical databases. However, the prime shortcoming of knowledge-based methods is that they have worse performance than supervised methods, but they have wider coverage for the polysemous words, thanks to the use of large-scale knowledge resources (Navigli, 2009).

There are many other tasks such as Chinese Word Segmentation (Zhang et al., 2018), Language Modeling (Ahn et al., 2016), and LSTMs (Xu et al., 2016; Yang and Mitchell, 2017) show that integrating knowledge and labeled data into a unified system can achieve better performance than other methods which only learn from large scale labeled data. Therefore, it's a promising and challenging study to integrate labeled data and lexical knowledge into a unified system.

A few recent studies of WSD have exploited several ways to incorporate lexical resources into supervised systems. In the field of traditional feature-based methods (Chen et al., 2015; Rothe and Schütze, 2015), they usually utilize knowledge (to train word sense embeddings) as features of the classifier like the support vector machine (SVM). In the field of neural-based methods, Raganato et al. (2017a) regard lexical resource LEX which is extracted from the WordNet as an auxiliary classification task, and propose a multi-task learning framework for WSD and LEX. Luo et al. (2018) integrate the context and glosses of the target word into a unified framework via a memory network. It encodes the context and glosses of the target word *separately*, and then models the semantic relationship between the context vector and gloss vector in the memory module. What's more, Luo et al. (2018) utilize much more knowledge about gloss via its semantic relations such as hypernymy and hyponymy in WordNet. All studies listed above show that integrating lexical resources especially gloss into supervised systems of WSD can significantly improve the performance. Therefore, we follow this direction and seek a new way of better integrating gloss knowledge.

Instead of building representations for context and gloss separately, we use the inner connection between the gloss and the context to promote the representation of each other. The interaction process can be modeled by a co-attention mechanism which has made great progress in the question answering task (Xiong et al., 2016; Seo et al., 2016; Hao et al., 2017; Lu et al., 2016). We are enlightened by this iterative procedure and introduce it into WSD. We then make some adaptations to the output of the original co-attention model to get the score of each word sense.

3 The Co-Attention Model for WSD

In this section, we first give an overview of the **CAN**: **c**o-**a**ttention **n**eural network for WSD (Figure 1). And then, we extend it into a **hi**erarchical architecture **HCAN** (Figure 2).

3.1 Overview

The overall architecture of the proposed non-hierarchical co-attention model is shown in Figure 1. It consists of three parts:

[1] *Play* in the sentence means *behave in a certain way*.

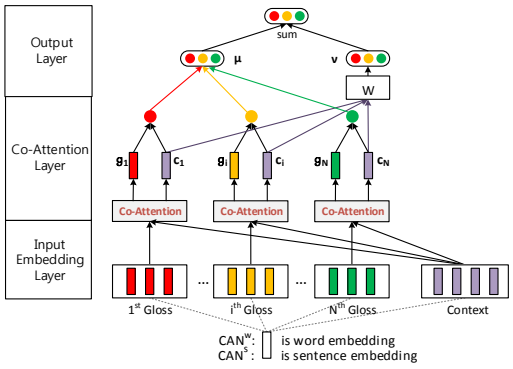

Figure 1: The proposed co-attention neural network.

- **Input Embedding Layer**: First of all, we encode the input context and each gloss [2] into distributed representations C and G, which are also called embeddings in the paper. In Figure 1, if C and G are word embeddings, we call the model CAN^w in the paper. If C and G are sentence embeddings, we call the model CAN^s.

- **Co-Attention Layer**: Then, each co-attention mechanism in this layer generates a context vector and a gloss vector according to the corresponding gloss and context representations. The outputs of the co-attention layer are N pairs of context vector and gloss vector.

- **Output Layer**: Finally, the output layer takes the N pairs of context vector and gloss vector as inputs and calculates the score of each word sense.

Figure 1 shows the non-hierarchical co-attention model which generates either word-level representations (CAN^w) or sentence-level representations (CAN^s). Since both the word-level and sentence-level representations can help to disambiguate the word sense, we extend CAN into a hierarchical model, named as HCAN (Figure 2). The extensions of each layer are listed as follows:

1. The input embedding layer is extended to two sub-layers in the hierarchical architecture which encodes both word-level and sentence-level representations.

2. The co-attention layer is also extended to two attention layers for capturing two different levels' attention.

3. The output layer merges the outputs of the two levels' co-attention layers and generates a sense probability over all word senses.

Since the non-hierarchical model CAN is a subset or a simplified version of the hierarchical model HCAN, the next sections are organized to illustrate the hierarchical co-attention model HCAN [3] shown in Figure 2.

3.2 Input Embedding Layer

We denote each input sentence (context or gloss) as a sequence of words $[x_1, x_2, \ldots, x_{T_x}]$, where T_x is the length of the input sentence.

3.2.1 Word Embedding

After looking up a pre-trained word embedding matrix $E_w \in \mathbb{R}^{d_w \times V}$, we transfer a one-hot vector x_i into a d_w-dimensional vector e_i. We treat $[e_1, e_2, \ldots, e_{T_x}]$ as the word-level representations of the sentence. Specifically, context's word-level representations are denoted as $[e_1^c, e_2^c, \ldots, e_n^c]$ and i-th gloss's word-level representations are denoted as $[e_1^{g_i}, e_2^{g_i}, \ldots, e_m^{g_i}]$, where n and m represent the max length of context and gloss.

3.2.2 Sentence Embedding

We utilize a bi-directional long short-term memory network (Bi-LSTM) to generate the hidden states of the input sentence. Each hidden state h_i is computed by the concatenation of the *forward* hidden state $\overrightarrow{h_i}$, and *backward* hidden state $\overleftarrow{h_i}$. So we treat $[h_1, h_2, \ldots, h_{T_x}]$ as the sentence-level representations. Specifically, context's sentence-level representations are denoted as $[h_1^c, h_2^c, \ldots, h_n^c]$ and i-th gloss's sentence-level representations are denoted as $[h_1^{g_i}, h_2^{g_i}, \ldots, h_m^{g_i}]$.

3.3 Co-Attention Layer

3.3.1 Co-Attention Mechanism

The right part of Figure 2 illustrates the co-attention mechanism which is the most crucial part of the model. The inputs are context representations $C \in \mathbb{R}^{d \times n}$ and gloss representations $G \in \mathbb{R}^{d \times m}$, where d is the dimension of the input representation vector. The outputs are the gloss-aware context vector $c \in \mathbb{R}^d$ and the context-aware gloss vector $g \in \mathbb{R}^d$. Therefore, we can define the co-attention mechanism as a function

$$(c, g) = CoAt(C, G) \quad (1)$$

[2] There are N glosses in total, where N is the sense number of the target word in the context.

[3] CAN^w and CAN^s will also be expressed in Section 3.4.

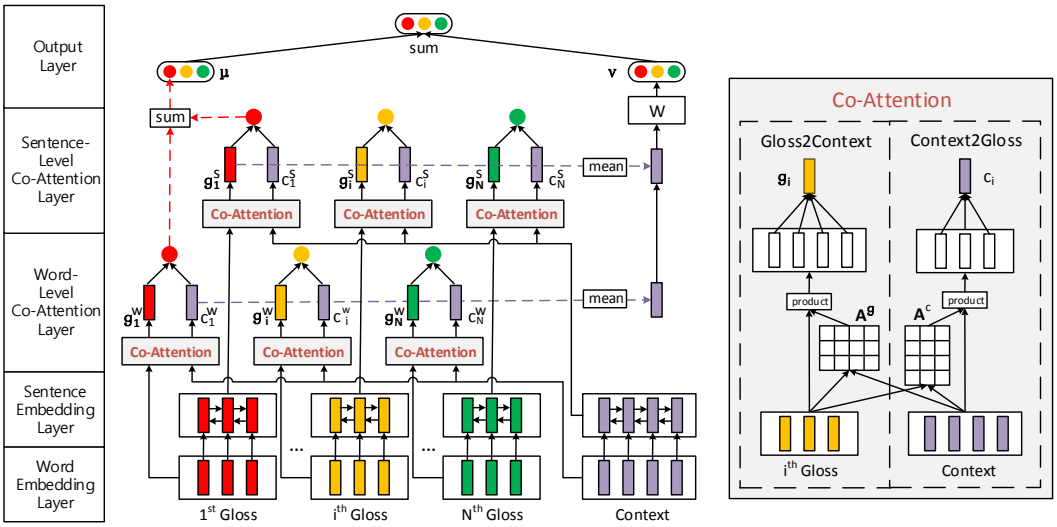

Figure 2: Hierarchical co-attention model for WSD.

Next, we give the detailed definition of the co-attention mechanism function $CoAt$. We begin to compute a similarity matrix A, in which each element A_{ij} indicates the similarity between i-th context word and j-th gloss word. The similarity matrix A is computed by

$$A = C^\top U G \in \mathbb{R}^{n \times m} \tag{2}$$

where $U \in \mathbb{R}^{d \times d}$ is a trainable parameter.

Based on the similarity matrix A, we can compute the gloss-to-context attention matrix A^c and context-to-gloss attention matrix A^g.

Gloss-to-Context Attention. Since each gloss word may focus on different context words, we can generate a context representation which is aware of a particular gloss word. Note that each element of A in the j-th column indicates the similarity between j-th gloss word and each context word. Thus, we can get the attention weight for each context word through a softmax function across the column of A:

$$A^c_{:j} = softmax(A_{:j}) \tag{3}$$

where $A_{:j}$ denotes j-th column of A and $A^c_{:j}$ denotes j-th column of $A^c \in \mathbb{R}^{n \times m}$.

Hence we can get the gloss-aware context representations $\hat{C}$ by a product of the initial context representations C and attention weight matrix A^c:

$$\hat{C} = CA^c \in \mathbb{R}^{d \times m} \tag{4}$$

Note that j-th column in $\hat{C}$ means the context representation according to the j-th gloss word. Therefore, we can get the final context vector c by summing across the column of $\hat{C}$:

$$c = \sum_j \hat{C}_{:j} \in \mathbb{R}^d \tag{5}$$

Context-to-Gloss Attention. Conversely, each context word may focus on different gloss words, we can generate a gloss representation which is aware of a particular context word. Since each element of A in the i-th row indicates the similarity between i-th context word and each gloss word, we can get the attention weight of each gloss word through a softmax function across the row of A (or across the column of $A^\top$)

$$A^g_{:j} = softmax(B_{:j}) \tag{6}$$

where $B = A^\top$, $B_{:j}$ denotes j-th column of B (also j-th row of A) and $A^g_{:j}$ denotes j-th column of $A^g \in \mathbb{R}^{m \times n}$.

Now we can get the context-aware gloss representations in the same way as Equation 4:

$$\hat{G} = GA^g \in \mathbb{R}^{d \times n} \tag{7}$$

Note that j-th column in $\hat{G}$ denotes the gloss representation according to j-th context word. Therefore, we can get the final gloss vector g by summing across the column of $\hat{G}$:

$$g = \sum_j \hat{G}_{:j} \in \mathbb{R}^d \tag{8}$$

3.3.2 Word-Level Co-Attention Layer

Since there are N glosses according to N different word senses, we use N independent co-attention mechanisms in both word-level and sentence-level co-attention layers. And each layer shares a same parameter U in Equation 2. For i-th word-level co-attention mechanism, the inputs are word embeddings of the context and i-th gloss (in Section 3.2.1). Define $C^w = [e_1^c, e_2^c, \ldots, e_n^c]$ and $G_i^w = [e_1^{g_i}, e_2^{g_i}, \ldots, e_m^{g_i}]$, thus the outputs of i-th word-level co-attention mechanism are computed as

$$(c_i^w, g_i^w) = CoAt(C^w, G_i^w) \tag{9}$$

Inspired by the well-known Lesk algorithm (Lesk, 1986) and its variants (Basile et al., 2014), the score of the i-th word sense can be computed as the similarity of the context vector c_i^w and the gloss vector g_i^w:

$$\beta_i^w = c_i^w \cdot g_i^w \tag{10}$$

The word-level context embedding vector $\hat{c}^w$ can be computed as the average of the N gloss-aware context vectors c_i^w:

$$\hat{c}^w = \frac{1}{N} \sum_{i=1}^{N} c_i^w \tag{11}$$

3.3.3 Sentence-Level Co-Attention Layer

Same to word-level co-attention layer, for the i-th co-attention mechanism, the inputs of sentence-level co-attention layer are Bi-LSTM hidden states of context and the i-th gloss (in Section 3.2.2). Define $C^s = [h_1^c, h_2^c, \ldots, h_n^c]$ and $G_i^s = [h_1^{g_i}, h_2^{g_i}, \ldots, h_m^{g_i}]$, thus the outputs of i-th sentence-level co-attention mechanism are computed as

$$(c_i^s, g_i^s) = CoAt(C^s, G_i^s) \tag{12}$$

Like Equation 10, we can also calculate a sentence-level score for the i-th word sense by a dot product of the context vector c_i^s and the gloss vector g_i^s:

$$\beta_i^s = c_i^s \cdot g_i^s \tag{13}$$

The sentence-level context embedding vector $\hat{c}^s$ is also computed as the average of N gloss-aware context vectors c_i^s:

$$\hat{c}^s = \frac{1}{N} \sum_{i=1}^{N} c_i^s \tag{14}$$

3.4 Output Layer

The output layer aims to calculate the scores of N senses of the target word x_t and finally outputs a sense probability distribution over the N senses. The final score of each sense is a weighted sum of two values: μ and ν. μ is the similarity score of gloss and context, which reveals the influence of knowledge. ν is generated by the context vector through a linear projection layer, which reveals the influence of labeled data. Finally, the probability distribution $\hat{y}$ over all the senses of the target word is computed as

$$\hat{y} = softmax(\lambda_{x_t}\mu + (1 - \lambda_{x_t})\nu))$$

where $\lambda_{x_t} \in [0, 1]$ is the parameter for word x_t.

For the non-hierarchical model CAN in Figure 1, the final score μ and ν are generated only by the outputs of the one level co-attention layer. Specifically, for the word-level co-attention model CAN^w:

$$\mu = [\beta_1^w, \beta_2^w, \ldots, \beta_N^w] \tag{15}$$

$$\nu = W_{x_t}\hat{c}^w + b_{x_t} \tag{16}$$

For the sentence-level co-attention model CAN^s:

$$\mu = [\beta_1^s, \beta_2^s, \ldots, \beta_N^s] \tag{17}$$

$$\nu = W_{x_t}\hat{c}^s + b_{x_t} \tag{18}$$

For the hierarchical co-attention model HCAN in Figure 2, the outputs of the word and sentence level layer are merged together to generate the final results. Therefore, the final similarity score between i-th gloss and context is computed as the weighted sum of word-level score β_i^w and sentence-level score β_i^s:

$$\beta_i = \alpha\beta_i^w + (1 - \alpha)\beta_i^s \tag{19}$$

Meanwhile, the final context embedding vector is also generated by a combination of two levels' context embedding vector: $\hat{c}^w$ and $\hat{c}^s$. In order to transfer from word-level encoding space to sentence-level encoding space, we introduce a non-linear projection layer on top of the word-level context vector $\hat{c}^w$. Therefore, the final context embedding vector $\hat{c}$ is generated by

$$\hat{c} = tanh(W\hat{c}^w + b) + \hat{c}^s \tag{20}$$

In total, for the hierarchical co-attention model HCAN:

$$\mu = [\beta_1, \beta_2, \ldots, \beta_N] \tag{21}$$

$$\nu = W_{x_t}\hat{c} + b_{x_t} \quad (22)$$

It's noteworthy that in Equation 16, 18 and 22, each ambiguous word x_t has its corresponding weight matrix W_{x_t} and bias b_{x_t}.

During training, all model parameters θ are jointly learned by minimizing a cross-entropy loss between $\hat{y}$ and the true label y.

$$L(\theta) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N_i} y_{ij}\log\hat{y}_{ij} \quad (23)$$

where M is the number of examples in the dataset, N_i is the word sense number of i-th example, y_{ij} and $\hat{y}_{ij}$ are the true and predict probability of the i-th example belongs to j-th label.

4 Experiments and Evaluation

4.1 Datasets

Validation and Evaluation Datasets: We evaluate our model on several English all-words WSD datasets. For a fair comparison, we use the benchmark datasets proposed by Raganato et al. (2017b) which include five standard all-words fine-grained WSD datasets from the Senseval and SemEval competitions:

- Senseval-2 (Edmonds and Cotton, 2001, **SE2**): It consists of 2282 sense annotations, including nouns, verbs, adverbs and adjectives.
- Senseval-3 task 1 (Snyder and Palmer, 2004, **SE3**): It consists of 1850 sense annotations from three different domains (editorial, news story and fiction), including nouns, verbs, adverbs and adjectives.
- SemEval-07 task 17 (Pradhan et al., 2007, **SE7**): It consists of 455 sense annotations of nouns and verbs, which is the smallest among the five datasets. Like Luo et al. (2018) and Raganato et al. (2017a), we choose SE7 as the validation set.
- SemEval-13 task 12 (Navigli et al., 2013, **SE13**): It consists of 1644 sense annotations from thirteen documents of various domains. SE13 contains nouns only.
- SemEval-15 task 13 (Moro and Navigli, 2015, **SE15**): It's the latest WSD dataset, which consists of 1022 sense annotations from three heterogeneous domains.

Training Dataset: SemCor 3.0 is the largest manually annotated corpus for WSD, which was also used by Luo et al. (2018), Raganato et al. (2017a), Raganato et al. (2017b), Iacobacci et al. (2016), Zhi and Ng (2010), etc. It consists of 226,036 sense annotations from 352 documents, which includes nouns, verbs, adverbs and adjectives.

Knowledge Base: The original WordNet version of sense inventory for SemCor 3.0, SE2, SE3, SE7, SE13, SE15 are 1.4, 1.7, 1.7.1, 2.1, 3.0 and 3.0, respectively. Raganato et al. (2017b) map all the sense annotations in the training and test datasets to WordNet 3.0 via a semi-automatic method. Therefore, We choose WordNet 3.0 as the sense inventory for extracting the gloss.

Data	Noun	Verb	Adj	Adv
SE2	1066	517	445	254
SE3	900	588	350	12
SE7	159	296	0	0
SE13	1644	0	0	0
SE15	531	251	160	80
SemCor	87002	88334	31753	18947

Table 2: Statistics of the different parts of speech annotations in English all-words WSD train and test datasets.

.

4.2 Settings

We use the validation set (SE7) to find the optimal hyper parameters of our models: the word embedding size d_w, the hidden state size d_s of LSTM, the optimizer, etc. However, since there are no adverbs and adjectives in SE7, we randomly sample some adverbs and adjectives from training dataset into SE7 for validation. We use the pre-trained word embeddings [4]. The hidden state size d_s is 256. The mini-batch size is set to 32. The optimizer is Adam (Kingma and Ba, 2014) with 0.001 initial learning rate. In order to avoid over-fitting, we use dropout regularization on the outputs of LSTM and set drop rate to 0.5. Orthogonal initialization is used for initialing weights in LSTM and random uniform initialization with range [-0.1, 0.1] is used for others. Training runs for up to 50 epochs with early stopping if the validation loss doesn't improve within the last 5 epochs.

[4]We download the pre-trained word embeddings from https://github.com/stanfordnlp/GloVe

	Test Datasets				Concatenation of Test Datasets				
System	SE2	SE3	SE13	SE15	Noun	Verb	Adj	Adv	**All**
MFS baseline*	65.6	66.0	63.8	67.1	67.7	49.8	73.1	80.5	65.5
Lesk$_{ext+emb}$ (Basile et al., 2014)	63.0	63.7	66.2	64.6	70.0	51.1	51.7	80.6	64.2
Babelfy (Moro et al., 2014)	67.0	63.5	66.4	70.3	68.9	50.7	73.2	79.8	66.4
IMS (Zhi and Ng, 2010)*	70.9	69.3	65.3	69.5	70.5	55.8	75.6	82.9	68.9
IMS$_{+emb}$ (Iacobacci et al., 2016)*	72.2	70.4	65.9	71.5	71.9	56.6	75.9	84.7	70.1
Bi-LSTM (Kågebäck and Salomonsson, 2016)*	71.1	68.4	64.8	68.3	69.5	55.9	76.2	82.4	68.4
Bi-LSTM$_{+att.+LEX}$ (Raganato et al., 2017a)	72.0	69.4	66.4	72.4	71.6	57.1	75.6	83.2	69.9
Bi-LSTM$_{+att.+LEX+POS}$ (Raganato et al., 2017a)	72.0	69.1	66.9	71.5	71.5	57.5	75.0	83.8	69.9
GAS (Linear) (Luo et al., 2018)	72.0	70.0	66.7	71.6	71.7	57.4	76.5	83.5	70.1
GAS (Concatenation) (Luo et al., 2018)	72.1	70.2	67.0	71.8	72.1	57.2	76.0	84.4	70.3
GAS$_{ext}$ (Linear) (Luo et al., 2018)	72.4	70.1	67.1	72.1	71.9	58.1	76.4	84.7	70.4
GAS$_{ext}$ (Concatenation) (Luo et al., 2018)	72.2	**70.5**	67.2	72.6	72.2	57.7	76.6	**85.0**	70.6
CAN^{w}	72.3	69.8	65.5	71.1	71.1	57.3	76.5	84.7	69.8
CAN^{s}	72.2	70.2	**69.1**	72.2	**73.5**	56.5	76.6	80.3	70.9
HCAN	**72.8**	70.3	68.5	**72.8**	72.7	**58.2**	**77.4**	84.1	**71.1**

Table 3: F1-score (%) for fine-grained English all-words WSD on the test sets. **Bold** font indicates best systems. The * represents the systems which don't use any lexical knowledge. The five blocks list the baseline, 2 knowledge-based systems, 2 supervised feature-based systems, 7 neural-based systems and our models, respectively.

4.3 Results and Discussion

4.3.1 English all-words results

Table 3 shows the results on four test datasets and different parts of speech. Note that all the systems in Table 3 are trained on SemCor 3.0.

In the first block, we show the MFS baseline, which simply selects the most frequent sense in the training dataset.

In the second block, we show two latest knowledge-based (unsupervised) systems. Lesk$_{ext+emb}$ is a variant of the well-known Lesk algorithm (Lesk, 1986) which computes the overlap of gloss and context as the score of word sense. Babelfy (Moro et al., 2014) is a graph-based system performed on BabelNet (Navigli and Ponzetto, 2012). We can find that MFS is a strong baseline for knowledge-based systems.

In the third block, we show two traditional supervised systems which only learn from labeled data based on manual designed features. IMS (Zhi and Ng, 2010) is a flexible framework which trains K SVM classifiers for K polysemous words. Its variant IMS$_{+emb}$ (Iacobacci et al., 2016) adds word embedding features into IMS. Both of them train a dedicated classifier for each word individually. In other words, each target word has its own parameters. Therefore, IMS$_{+emb}$ is a hard to beat system for many neural networks which also only uses labeled data but builds a unified system for all the polysemous words.

In the fourth block, we show four latest neural networks. Except for Bi-LSTM (Kågebäck and Salomonsson, 2016), which is a baseline for neural models, the others all utilize not only labeled data but also lexical knowledge. Bi-LSTM$_{+att.+LEX}$ (Raganato et al., 2017a) and its variant Bi-LSTM$_{+att.+LEX+POS}$ are multi-task learning frameworks for WSD, POS tagging and LEX with context self-attention mechanism. GAS (Luo et al., 2018) is a gloss-augmented neural network in an improved memory network paradigm. The best neural network is GAS$_{ext}$ which extends from GAS and uses more gloss knowledge via the semantic relations in WordNet. [5]

In the last block, we give the performance of our proposed co-attention models for WSD. We can see that our best model HCAN improves state-of-the-art result by 0.5% on the concatenation of four datasets. Even though we use less gloss knowledge than the previous best system GAS$_{ext}$, our co-attention models can still get the best results on three test datasets. For non-hierarchical models, CAN^{s} performs much better than the CAN^{w}, which reveals that global sentence-level information is much more useful than local word-level information. Integration of these two levels' information (HCAN) can further boost the performance. What's more, we find that our best model HCAN performs best on all parts of speech, except for *adverbs*. However, there are only 346 examples about adverbs which account for 5% of the four test datasets, thus 1% drop on adverbs means only 4 examples are wrongly classified which will make little influence on the overall score.

[5] The released code can be found in https://github.com/luofuli/word-sense-disambiguation

	Test Datasets				Concatenation of Test Datasets				
System	SE2	SE3	SE13	SE15	Noun	Verb	Adj	Adv	**All**
Full Model	**72.8**	70.3	**68.5**	**72.8**	72.7	**58.2**	**77.4**	**84.1**	**71.1**
No Attention	70.2	68.1	67.6	68.9	71.2	54.3	74.3	82.1	68.8
W/O Word-level Attention	71.5	**70.4**	68.2	71.7	72.7	57.1	75.3	81.8	70.5
W/O Sentence-level Attention	70.0	69.5	66.8	70.3	71.3	55.2	74.4	83.0	69.1
W/O Context2Gloss Attention	70.7	69.7	68.2	71.0	72.2	55.2	75.5	82.7	69.9
W/O Gloss2Context Attention	72.3	70.3	68.2	71.9	**73.2**	56.4	75.4	83.8	70.7

Table 4: Ablation study of the proposed model HCAN.

.

4.4 Ablation Study

In this part, we further discuss the impacts of the components of our hierarchical model HCAN. In order to ablate the co-attention mechanism, we replace the co-attention function $CoAt$ in Equation 1 with a function Avg which simply calculates the average of input representation vectors. Specifically, in function Avg, the outputs $c = \sum_j C_{:j}$ and $g = \sum_j G_{:j}$.

We re-train HCAN by ablating certain components:

- *No Attention*: We totally replace the co-attention function $CoAt$ with Avg in both word-level and sentence-level co-attention layers. This is the baseline for comparison.

- *W/O Word-level Attention*: We replace the word-level co-attention function $CoAt$ with Avg. Note that this ablation model is different from CAN^s, for that the word-level representation vector $\hat{c}^w$ is used to calculate the final score in this ablation model.

- *W/O Sentence-level Attention*: We replace the sentence-level co-attention function $CoAt$ with Avg. Note that this ablation model is not same as CAN^w, for the sentence-level representation vector $\hat{c}^s$ is also used to calculate the final score in this ablation model.

- *W/O Context2Gloss Attention*: We remove the attention of generating the context vector, which means all elements in A^g are set to 1.

- *W/O Gloss2Context Attention*: We remove the attention of generating the gloss vector, which means all elements in A^c are set to 1.

Table 4 indicates the effectiveness of different components in the proposed model HCAN. It shows that without any attention mechanism, the overall score declines 2.3%.

Ablated versions without word, sentence level co-attention decline 0.6% and 2.0%, respectively. It reveals that sentence-level co-attention mechanism seems much more important to HCAN, which is consistent with the scores of CAN^s and CAN^w. However, we find that the results of ablated versions without word-level and sentence-level co-attention are worse than CAN^s and CAN^w. We hypothesize that it is because that the context and gloss vector generated from the layer (or level) which doesn't use attention mechanism may bring some noise to the final scores.

Without the context-to-gloss attention, the score declines 1.2% on concatenation of the four test datasets. Conversely, without the gloss-to-context attention, the score declines 0.4%. It is probably due to that the context-to-gloss attention which generates the context-aware gloss vector is more *direct* to find out the correct word sense.

In conclusion, the results in Table 4 show that all components in the proposed hierarchical co-attention model HCAN can contribute to boosting the performance of WSD.

5 Conclusions

In this paper, we investigate the problem of incorporating gloss knowledge into neural network for Word Sense Disambiguation. We find that the gloss can highlight the important words in the context, and later contribute to the representation of the context. Meanwhile, context can also help to focus on the words in gloss of the right word sense. Therefore, we propose a co-attention mechanism to model the gloss-to-context and context-to-gloss attention. Furthermore, in order to capture not only local word-level features but also global sentence-level features, we extend the co-attention model into a hierarchical architecture. The experimental results show that our proposed models achieve the state-of-the-art results on several standard English all-words WSD datasets.

Acknowledgments

We would like to thank Lei Sha, Jianmin Zhang and Junbing Liu for insightful comments and suggestions. This paper is supported by NSFC project 61772040 and 61751201. The contact authors are Baobao Chang and Zhifang Sui.

References

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

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *CoRR*, abs/1608.00318.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced lesk word sense disambiguation algorithm through a distributional semantic model. In *Roceedings of COLING 2014, the International Conference on Computational Linguistics: Technical Papers*.

T. Chen, R. Xu, Y. He, and X. Wang. 2015. Improving distributed representation of word sense via wordnet gloss composition and context clustering. *Atmospheric Measurement Techniques*, 4(3):5211–5251.

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

Yanchao Hao, Yuanzhe Zhang, Kang Liu, Shizhu He, Zhanyi Liu, Hua Wu, Jun Zhao, Yanchao Hao, Yuanzhe Zhang, and Kang Liu. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Meeting of the Association for Computational Linguistics*, pages 221–231.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *The Meeting of the Association for Computational Linguistics*.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional lstm. *arXiv preprint arXiv:1606.03568*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computer Science*.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries:how to tell a pine cone from an ice cream cone. In *Acm Special Interest Group for Design of Communication*, pages 24–26.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 289–297.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

Roberto Navigli. 2009. Word sense disambiguation:a survey. *Acm Computing Surveys*, 41(2):1–69.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *The International Workshop on Semantic Evaluation*.

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

Sameer S. Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task 17: English lexical sample, srl and all words. In *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 *Conference on Empirical Methods in Natural Language Processing*.

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

Sascha Rothe and Hinrich Schütze. 2015. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv preprint arXiv:1507.01127*.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603.

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604.

Zhen Xu, Bingquan Liu, Baoxun Wang, Chengjie Sun, and Xiaolong Wang. 2016. Incorporating loose-structured knowledge into LSTM with recall gate for conversation modeling. *CoRR*, abs/1605.05110.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Meeting of the Association for Computational Linguistics*, pages 1436–1446.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for chinese word segmentation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*.

Zhong Zhi and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *ACL 2010, Proceedings of the Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden, System Demonstrations*, pages 78–83.