Bi-Branch Meta-Learning for Few-Shot Word Sense Disambiguation

Qingying Chen, Jing Zhang, Peng Zhang[1], Hui Gao
qchen414@gatech.edu, {zhang_jing, pzhang, hui_gao}@tju.edu.cn
Tianjin University, Peiyangyuan Campus: No.135 Yaguan Road, Tianjin, China

Abstract

Word Sense Disambiguation (WSD) has been a fundamental task for human language understanding. In specific contexts, a word may have different meanings. For rarely seen word senses, the disambiguation becomes challenging with limited examples. Meta-learning, as a widely adopted machine learning method for few-shot learning, addresses this by extracting metacognitive knowledge from training data, aiding models in "learning to learn". Hence, the advancement of meta-learning hinges on leveraging high-quality metacognitive knowledge. In light of this, we propose a *Bi-Branch Meta-Learning* method for WSD to enrich and accumulate metacognitive insights. Our method employs two branches during training and testing. During training, we use a bi-branch loss with original and augmented data from large language models to compensate for data scarcity. In testing, information from base classes generates bi-branch scores to refine predictions. Experiments show our method achieves a 74.3 F1 score in few-shot scenarios, demonstrating its potential for few-shot WSD.

Keywords: artificial intelligence; computer science; natural language processing; few-shot learning; word sense disambiguation; meta-learning

Introduction

Word sense disambiguation (WSD) is the cornerstone of many downstream tasks (Blevins & Zettlemoyer, 2020). Essentially, WSD aims to distinguish the sense of a word given a specific context and different candidate senses. Understood from the perspective of Natural Language Processing (NLP), WSD is a multi-classification task given specific inputs and classes.

However, one big challenge in WSD is the "long-tail" distributions of word senses. This means that most senses have scarce examples in the corpus while the examples of a small proportion of word senses dominate the majority of scenarios. Additionally, the most frequent sense (MFS) accounts for the majority of the corresponding samples for each word. Such an extremely skewed sense distribution often leads many WSD systems to favor selecting MFS for each word while neglecting the less frequent sense (LFS). Yet to gain reliable performance on WSD, a system ought to be equally proficient in distinguishing LFS and MFS. To address such limitations, few-shot learning (FSL) methods have long been used for the WSD task. FSL refers to a type of machine learning problem that only contains limited examples to learn the target task.

One promising paradigm among FSL methods is meta-learning. Essentially, meta-learning is "learning to learn" through successively training models on different tasks while traditional machine learning methods improve models over multiple data instances (Hospedales, Antoniou, Micaelli, & Storkey, 2021). The motivation behind meta-learning is to accumulate experiences across diverse tasks to enhance performance on new and unseen tasks. This intention aligns with the human learning process in the concept of metacognition, which is often simply defined as "thinking about thinking" (Hospedales et al., 2021; Livingston, 2003).

Building on this commonality, we propose that they are likely to have similar construction. Therefore, it is plausible that meta-learning also incorporates metacognitive knowledge, as it is regarded as a fundamental component in the initial definition of metacognition. Metacognitive knowledge encompasses insights into managing cognitive processes and achieving goals effectively (Flavell, 1979). Given this perspective, we infer that in meta-learning on few-shot tasks, the overfitting phenomenon (See section Preliminaries) might be caused by the lack of the accumulation of metacognitive insights. Therefore, to improve the performance of language models on WSD with extremely few examples, we propose a bi-branch meta-learning method in both the training and testing phases. We aim to compensate the insufficiency of the metacognitive knowledge in meta-learning with the information from the base classes and the ample augmented data.

During the training phase, we design a bi-branch training loss to guide the model to learn the WSD task and to alleviate overfitting at the same time. One branch is the training loss of the supervised meta-learning on the WSD task; for the other branch in training, we first employ Large Language Models (LLMs) to implement data augmentation by paraphrasing sentences, then compute an unsupervised paraphrasing training loss based on the original data and augmented data. Both branches of loss are calculated with a metric-based meta-learning method. They together constitute the bi-branch training loss for optimizing our bi-branch model.

In the testing phase, we aim to leverage information from base classes seen in the training process for calibrating the final predictions. Specifically, the final sense prediction is made by two branches of classification score. One branch is the classical similarity score adopted from Prototypical Network (Snell, Swersky, & Zemel, 2017), which is also applied for computing loss during training. The other is the base similarity score, the similarity between the distributions on the

[1]Corresponding author: Peng Zhang

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

base classes for the support set and query set (See section Preliminaries).

Our bi-branch meta-learning model is evaluated on a unified evaluation framework for WSD proposed by Raganato, Camacho-Collados, Navigli, et al.. Without any dictionary definitions (or gloss), our model achieves a 74.3 F1 score under constrained training and testing conditions, which is competitive to other non-gloss baselines without such settings. Our experiment further indicates that such performance of the bi-branch model resulted from a robust capacity to disambiguate in both MFS and LFS situations. The bi-branch meta-learning method thus is potentially a simple and effective solution not only to WSD but also to other few-shot tasks.

Preliminaries

In this section, we provide an overview of the preliminary knowledge relevant to our work. We aim to acquaint readers from diverse backgrounds with the essential concepts and terminologies for understanding our work.

Meta-Learning

Meta-learning, as described above, focuses on "learning to learn" through iterative tasks. More specifically, the training set and testing set in meta-learning both consist of episodes rather than individual data instances. Each episode is solving a specific task T_i, which includes a set of training examples called support set $\mathbb{D}_{support}$, and a query set $\mathbb{D}_{query}$ for evaluating. There is a forward propagation and update of the gradient after each episode. In this way, episodic learning is able to utilize limited resources effectively for better generalization and robustness. Typically, the setting where each episode has N classes in $\mathbb{D}_{support}$ and K examples per class, is called an N-way K-shot setting.

For WSD, the disambiguation of a word within a context is naturally a classification task that can be treated as an episode. Thus, the disambiguation of a word with N selected senses and K examples for each sense is an N-way K -shot task in episodic learning (Holla, Mishra, Yannakoudakis, & Shutova, 2020). When K is extremely small, the case becomes few-shot learning where the metacognitive knowledge might be activated during the training. This assertion is built on the opinion of Flavell (1979) - metacognitive knowledge is triggered when the task is novel or incomplete (Wenden, 1998).

Metric Learning

Metric-based meta-learning (or metric learning), is a typical genre of meta-learning. Its purpose is to train the model's feature-capture capacity by reducing the distances between similar instances and widening the distances between dissimilar ones (Kaya & Bilge, 2019). The computation of these distances (or metrics) may vary from specific task requirements, primarily in the way of different similarity metrics. Higher similarity values indicate a shorter distance between samples, while lower values imply a greater distance.

Specifically, for classification tasks like WSD, metric learning models first learn to extract feature information from base classes by optimizing the similarity between query samples from $\mathbb{D}_{query}$ and support samples from $\mathbb{D}_{support}$. During this process, models are encouraged to discern the patterns and similarities within the training data, just as children develop their metacognition through pertinent factual knowledge about the cognitive process (Alexander, Carr, & Schwanenflugel, 1995). Subsequently, during the testing phase, the similarity is used as the classification score to classify the query samples. In this work, we introduce two types of similarity to form the bi-branch classification score in the testing phase - classical similarity from Prototypical Network, and base similarity proposed in the work of Wang, Zhao, Li, and Tian (2020) (See section Methodology).

Data Augmentation

Data Augmentation (DA) is a widely applied technique for FSL. It enlarges the data with newly created synthetic data or modified instances of the original data (Li, Hou, & Che, 2022). As described in Introduction, we incorporate augmented data along with the original data to compute a branch of training loss, compensating for the insufficiency of metacognitive knowledge. Such an insufficiency might result in overfitting, a phenomenon in machine learning where the model tries to fit to the noise rather than identify a prediction rule during training (Dietterich, 1995). Overfitting leads to poor performance on testing despite extremely great performance during training. In this work, we apply the *paraphrasing* technique to paraphrase the original data as the augmented data.

However, paraphrasing-based DA is a persistent challenge. Back translation, as a mainstream approach, relies on bilingual parallel corpora. Another approach that involves original text-oriented augmentation through synonym replacement may alter the intended meaning (Sun, Ouyang, Zhang, & Dai, 2021). In contrast, PROTAUGMENT (Dopierre, Gravier, & Logerais, 2021) fine-tunes a paraphrasing model for effective augmentation without domain-specific training. Yet it still needs an extra training process.

Different from the above approaches, we use Large Language models (LLMs) to augment data. LLMs have been proven to be a powerful tool in a wide range of NLP tasks, including text generation. Therefore, LLMs offer a simple and efficient solution to DA by generating paraphrased texts - DA can be achieved in the way of natural human dialogues. In this work, we select ChatGPT (Ouyang et al., 2022) to paraphrase the samples in the training dataset.

Methodology

In this section, we will formally introduce the task definition of WSD and illustrate how our bi-branch meta-learning method works. The entire training and testing phases of our model are shown in Fig 1.

Task Definition

WSD, from the perspective of artificial intelligence, is to computationally identify the sense of a word with a specific

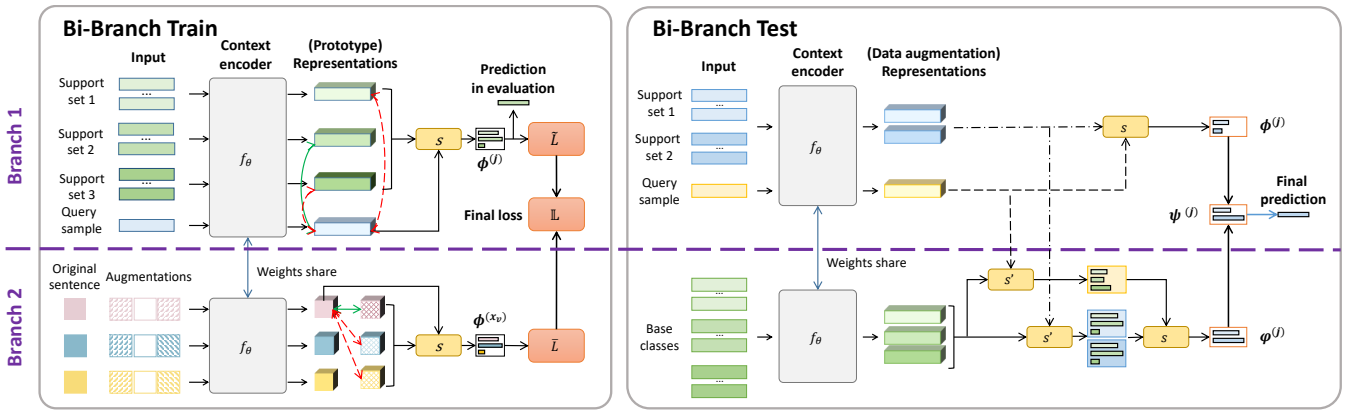

Figure 1: The bi-branch meta-learning method illustration, where the purple dashed line separates two branches in each phase. The left part illustrates the training phase, with supervised learning loss $\widetilde{L}$ and unsupervised paraphrasing loss $\overline{L}$. Green solid lines with arrows indicate pulling the representations closer while red dashed ones indicate pushing them away. The right part shows the testing phase, with two similarity scores $\phi^{(j)}$ and $\varphi^{(j)}$ constituting the final prediction $\psi^{(j)}$.

context (Navigli, 2009). Given a context C that consists of a sequence of k words $(w_1,...,w_k)$, a WSD system can assign a sense y to each to-be-disambiguated word w_i, where $y_i \in S_{w_i} \subset S$, S_{w_i} is the collection of all the candidate senses of word w_i, and S is all the senses included in the corpus. We further denote the input of a sample for the WSD system as $\overline{C} = (C,t)$, where t is the position of the target word in the context. Theoretically, each sample in $\mathbb{D}_{support}$ and $\mathbb{D}_{query}$ can be considered as the input $\overline{C}$ of the WSD system. The sense prediction $\hat{y}$ of a sample generated by a WSD system can be expressed as $\hat{y} = f(\overline{C})$, where the function f is the WSD system itself.

Bi-Branch Train

We adopt a bi-branch training loss on supervised and unsupervised learning during the training process, as shown in the left part of Fig 1.

Branch 1: Supervised Meta-Learning In the supervised meta-learning branch, we first calculate the prototype representation for each support set and the context representation of each query sample. Then we use the representations to calculate the supervised learning loss[2].

1) Prototype representation In Prototypical Network, each class (i.e. sense in our task) is treated as a prototype. The purpose of Prototypical Network is to minimize the representation distance between the query sample and the prototype of its corresponding class, while maximizing the distances between the query sample and other prototypes (See Fig. 1).

The prototype representation $\boldsymbol{p}_j$ of a sense j for a word w is computed as follows:

$$\boldsymbol{p}_j = \frac{1}{|C_j(w)|} \sum_{\overline{C} \in C_j(w)} f_\theta(\overline{C}), \tag{1}$$

[2]We adopt the structure of MetricWSD (Chen, Xia, & Chen, 2021) to compute supervised learning loss.

where $C_j(w)$ contains all the support examples of sense j for the word w, $f_\theta(\overline{C})$ is the representation of a context $\overline{C}$ for the word in the tth position.

In this work, we use the pre-trained language model BERT (Devlin, Chang, Lee, & Toutanova, 2019) to initialize our context encoder. The context representation is the tth output of the context encoder if the position of the to-be-disambiguated word is t. $f_\theta(\overline{C})$ is thereby represented as:

$$f_\theta(\overline{C}) = \text{BERT}(C)[t], \tag{2}$$

where f_θ is the context encoder. If a word is split into multiple pieces, we take the average of these pieces' context encoding as its representation. For unlabeled context without word position in unsupervised paraphrasing loss, we take the first encoding output as its representation, i.e. $t = 0$.

2) Supervised learning loss To construct the supervised learning loss based on prototypical representations, we introduce the classical similarity, which reflects the distance between a query sample and a support set. In the first branch, the classical similarity between a query sample within a context $\overline{C}'$ and the prototype of sense j is computed as follows:

$$\phi^{(j)} = \text{sim}(\boldsymbol{p}_j, f_\theta(\overline{C}')), \tag{3}$$

where $\text{sim}(\cdot,\cdot)$ is the similarity function. During the evaluation of training, the prediction $\hat{y}$ is made by:

$$\hat{y} = \arg\max_n(\phi^{(n)}), \tag{4}$$

as the higher the classical similarity is, the closer the query sample and the support set are. Then the supervised learning loss $\widetilde{L}$ is computed with $\phi^{(j)}$ in the way of cross-entropy loss:

$$\widetilde{L} = \frac{1}{|C(w)|} \sum_{i=1}^{|C(w)|} L_i, L_i = -\sum_{j \in S_w} y_{ij} \log(p_{ij}), \tag{5}$$

where L_i is the loss of each query sample, $C(w)$ is all the query samples for a word w, S_w is all the senses of w. y_{ij} is a sign function and p_{ij} is the probability that the sense of the query sample is j:

$$y_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}, p_{ij} = \frac{\phi^{(j)}}{\sum_{k \in S_w} \phi^{(k)}} \tag{6}$$

Branch 2: Unsupervised Paraphrasing Learning The second component of the training loss is derived from unsupervised paraphrasing learning[3]. The minimization of this loss pulls closer a sentence's entire representation and its data augmentation representation while pushing away other original sentences' data augmentation representations (See the bottom left of Fig 1). In this way, the model is encouraged to absorb more metacognitive knowledge by generalizing the unlabeled sentences, and so as to improve overfitting caused by insufficient data in the original WSD task.

1) Data augmentation representation The data augmentation representation of an original sentence is computed as the average of representations of its augmented sentences[4]. The prompt we used for ChatGPT to paraphrase data is:

> *user:* {*Input:* ***[Sentence to be paraphrased]***}
> *Please generate* ***M*** *variants of the user's input and the numbering style should be 1. 2. 3. and so on.*

where **M** is the number of all the paraphrased instances for an original sentence $x_u \in U$, and U is the collection of all the sentences in the training dataset, with the labels removed. The data augmentation representation $\boldsymbol{p}_{x_u}$ for an unlabeled sentence x_u is calculated as:

$$\boldsymbol{p}_{x_u} = \frac{1}{M} \sum_{m=1}^{M} f_\theta(\overline{x_u^m}),\ \overline{x_u^m} = (x_u^m, 0), \tag{7}$$

where x_u^m is the m^{th} paraphrased sentence for x_u.

2) Unsupervised learning loss Same to supervised learning loss, we calculate the classical similarity $\phi^{(x_v)}$ between the representation of a sentence u and the data augmentation representation $\boldsymbol{p}_{x_v}$ of another sentence v:

$$\phi^{(x_v)} = \text{sim}(\boldsymbol{p}_{x_v}, f_\theta(x_u)). \tag{8}$$

During training, the unsupervised paraphrasing learning encourages $\phi^{(x_v)}$ to approach 1 when $u = v$ and approach 0 otherwise. For consistency, we compute the unsupervised paraphrasing loss $\overline{L}$ with the cross-entropy method similar to Eq. 5 and 6. The final bi-branch training loss $\mathbb{L}$ is computed as:

$$\mathbb{L} = (1 - \omega T)\widetilde{L} + \omega T \overline{L}, \tag{9}$$

where T is the consumed training time in each epoch, and ω is a positive hyperparameter to adjust the proportions of $\overline{L}$ and $\widetilde{L}$. The effect of $\overline{L}$ increases as time goes by.

[3]Here we follow the computation of PROTAUGMENT.

[4]The representation is the first output since there are no labels.

Bi-Branch Test

During testing, the model predicts the sense through two similarity scores: classical similarity score ϕ and base similarity score φ. The former is introduced in Bi-Branch Train. The latter refers to the similarity between two distributions: 1) distribution of the query sample on base classes; 2) distribution of a support set (in the way of a prototype) on base classes.

Bi-branch classification score intends to calibrate the final prediction by leveraging implicit information from the base classes, which can be considered as one origin of metacognitive knowledge in meta-learning. The bi-branch classification score has been proven to be effective in image classification (Wang et al., 2020), which is built on the observation that most current metric learning strategies focus on the inner relationship between $\mathbb{D}_{support}$ and $\mathbb{D}_{query}$ but do not make full use of the base classes. More specifically, most existing metric learning methods only involve ϕ while the bi-branch method further introduces φ to adjust the final prediction score, as depicted in the bottom right of Fig 1.

Branch 1: Classical similarity classification score For meta-learning methods, the structure of the episode is shared within the training and testing. Therefore, the computation of classical similarity classification score $\phi^{(j)}$ for assigning a query sample to the sense j is the same as in Eq. 3.

Branch 2: Base similarity classification score While originally used for image classification, we believe that the base similarity classification score is more effective for WSD. In WSD, base classes refer to the senses of a word that appeared in training. Unlike in the image classification task, where classes in an episode are randomly chosen from the entire training dataset, there is some conceptual overlap between different senses of a word (Klein & Murphy, 2002). Therefore, the base similarity built on base classes is more likely to provide additional information in few-shot WSD.

Accordingly, the base similarity $\varphi^{(j)}$ is computed as:

$$\varphi^{(j)} = \text{sim}(\rho_{query}, \rho_{support}^{(j)}). \tag{10}$$

where ρ_{query} is the query sample's distribution on base classes, and $\rho_{support}^{(j)}$ is the distribution of prototype $\boldsymbol{p}_j$ for sense j on base classes [5]:

$$\begin{aligned} \rho_{query} &= \text{sim}'(f_\theta(\overline{C}'), \boldsymbol{B}), \\ \rho_{support}^{(j)} &= \text{sim}'(\boldsymbol{p}_j, \boldsymbol{B}). \end{aligned} \tag{11}$$

$\overline{C}'$ is the current query sample input, $\boldsymbol{B}$ is the matrix consisting of the prototype vectors for all base classes:

$$\begin{aligned} \boldsymbol{B} &= [\boldsymbol{b}^{(1)}, \boldsymbol{b}^{(2)}, \ldots, \boldsymbol{b}^{(|S_w|)}], \\ \boldsymbol{b}^{(i)} &= \frac{1}{|C_{(i)}(w)|} \sum_{\overline{C} \in C_{(i)}(w)} f_\theta(\overline{C}), \end{aligned} \tag{12}$$

[5]$\text{sim}'(\cdot,\cdot)$ refers to another similarity function that can be used to calculate distributions. It is not necessarily identical with $\text{sim}(\cdot,\cdot)$ introduced in Eq. 3. We use the cosine similarity function for both since it performs better than other similarity functions in this work.

Table 1: F1 scores (%) of our model and previous work on WSD. Dataset **ALL** concatenates all the test datasets and development dataset **SE07**. Our bi-branch model adopts $\alpha = 0.35$ (Eq. 13) to calculate the final classification score in the experiments.

	Gloss	SE07	SE02	SE03	SE13	SE15	ALL
WordNet S1	✗	55.2	66.8	66.2	63.0	67.8	65.2
Most frequent sense (MFS)	✗	54.5	65.6	66.0	63.8	67.1	65.5
Bi-LSTM	✗	64.8	72.0	69.1	66.9	71.5	69.9
BERT-kNN	✗	64.6	74.7	73.5	70.3	73.9	72.6
BERT-classifier	✗	68.6	75.9	74.4	70.6	75.2	73.5
1sent	✗	67.0	75.0	71.6	69.7	74.4	72.7
Bi-Branch-1shot (ours)	✗	67.0	76.4	72.8	72.0	75.1	**74.3**
Bi-Branch-2shot (ours)	✗	**69.2**	76.7	**74.8**	72.2	76.1	74.0
Bi-Branch-3shot (ours)	✗	68.1	**77.0**	74.5	**72.6**	76.0	**74.3**
MetricWSD-1shot	✗	67.5	75.1	72.6	71.2	75.0	72.6
MetricWSD-2shot	✗	68.8	76.4	74.1	71.5	76.1	73.7
MetricWSD-3shot	✗	68.1	76.6	74.2	72.0	**76.5**	73.8
EWISE	✓	67.3	73.8	71.1	69.4	74.5	71.8
HCAN	✓	/	72.8	70.3	68.5	72.8	71.1
SVC	✓	74.1	**79.7**	76.1	78.6	80.4	78.3
GlossBERT	✓	72.5	77.7	75.2	76.1	80.4	77.0
BEM	✓	**74.5**	79.4	**77.4**	**79.7**	**81.7**	**79.0**

where $\boldsymbol{b}^{(i)}$ represents a base class for a word w, and $C_{(i)}(w)$ annotates the selected samples in this class.

The final prediction score $\psi^{(j)}$ for assigning the sense j to a query sample is the linear combination of the two branches of similarity scores, adjusted by a positive hyperparameter α:

$$\psi^{(j)} = \alpha\phi^{(j)} + (1-\alpha)\varphi^{(j)}. \tag{13}$$

Apparently, during the testing phase, the bi-branch classification score is a non-parametric approach as it only calibrates on existing data and does not require any update for the model. Hence, we consider this approach to be a convenient and resource-efficient method for handling few-shot scenarios in the real world.

Experiments

In this section, we perform different experiments to verify our bi-branch model's effectiveness. We first assess its overall WSD performance against various baseline models. Next, we analyze its performance across different word frequencies to validate its effectiveness in few-shot scenarios. Lastly, we conduct ablation and performance experiments to delve deeper into our model's capabilities.

Dataset

We evaluate our bi-branch method on the WSD framework. Following the previous work, we train our model on **Semcor** and use **SemEval-2007 (SE07)** as the development set. **Senseval-2 (SE02)**, **Senseval-3 (SE03)**, **SemEval-2013 (SE13) SemEval-2015 (SE15)** are used for testing.

Baselines

To analyze the performance of our bi-branch model, we compare it to a series of baseline systems on the WSD task. The first two are **WordNet S1** and **MFS** (Most Frequent Sense), where **WordNet S1** predicts the first sense in the corpus and **MFS** predicts the most frequent sense. We also introduce several competitive baselines: **Bi-LSTM** (Raganato, Bovi, & Navigli, 2017); **BERT-kNN** and **BERT-fine-tuned classifier**, both trained with Bert-base model; **1sent** (Hadiwinoto, Ng, & Gan, 2019); **MetricWSD** (Chen et al., 2021). Furthermore, we include baselines that leverage gloss information for reference: **EWISE** (Kumar, Jat, Saxena, & Talukdar, 2019), **HCAN** (Luo et al., 2018), **SVC** (Vial, Lecouteux, & Schwab, 2019), **GlossBERT** (Huang, Sun, Qiu, & Huang, 2019), and **BEM** (Blevins & Zettlemoyer, 2020).

During experiments, to mimic the real-world word sense distribution, where the majority of the senses only have extremely few samples, we set the number K of samples in the support set in testing as 1, 2, and 3 for episodic learning systems, corresponding to 1-shot, 2-shot, and 3-shot settings.

Overall Results under Few-Shot Setting

In this section, we present the overall results of our bi-branch model and the baselines under few-shot settings. Our bi-branch model achieves the highest F1 scores among most test datasets when compared to non-gloss WSD systems. It also outperforms two gloss-involved systems: **EWISE** and **HCAN**. Further, our model is evaluated with few shots during both training and testing, distinguishing itself from other

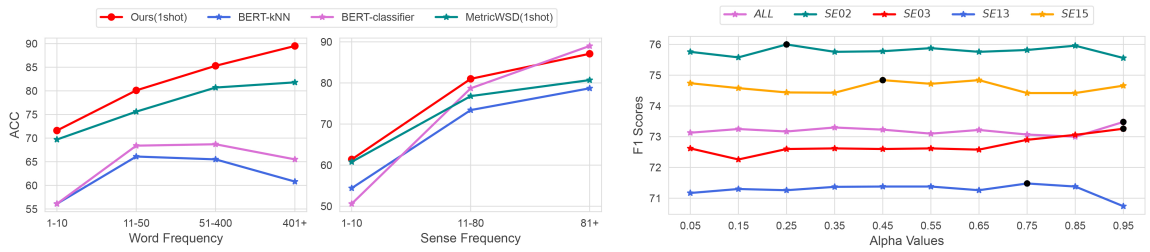

Figure 2: Left: accuracy across different word and sense frequencies on the test dataset **ALL**. Right: F1 scores with different values of α on the test datasets where the highest F1 score of each dataset is annotated with a black dot.

baselines that lack similar settings. In addition, the shot number K under extreme situations does not necessarily result in better performance of our model, which might be attributed to the reliable generalization capacity derived from the accumulation of metacognitive knowledge. In essence, our bi-branch model sustains relatively comparable performance when trained and tested with limited evaluation resources.

We have also observed that incorporating gloss information into the WSD system may potentially elevate its performance significantly, as evidenced by notably higher F1 scores compared to other systems. Such performance is probably due to the integration of a novel source of metacognitive knowledge - embedding gloss allows the system to better regulate cognitive processes on WSD, just like humans disambiguate senses with the assistance of a dictionary. Hence, we plan to incorporate gloss as an additional branch in future iterations.

Performance Across Words and Senses

To assess our model's capability to distinguish rare senses, we compare the prediction accuracy of our model with other models (**BERT-kNN**, **BERT-classifier**) in disambiguating senses of different word or sense frequencies. As shown in the left part of Fig. 2, our bi-branch model performs the best across all different frequencies of words. For frequencies of senses, though **BERT classifier** works better than other models on senses that appear over 80 times, it has the lowest accuracy score on the most infrequent samples, meaning that it is probably overfitted on the frequent senses during training. The result thereby illustrates that our bi-branch model is an effective method to tackle the WSD task with long-tail distributions - it enhances the disambiguation of infrequent senses while maintaining great performance on common ones.

Ablation Study

To assess the impact of the bi-branch on the overall model performance, we ablate one branch in the training and testing phases respectively. The results are shown in Table 2. We first remove the unsupervised paraphrasing training loss $\overline{L}$ and evaluate the model on **SE07**. The performance thereby drops by 0.56 F1 score. This might be because the unsupervised paraphrasing loss prevents the model from overfitting by encouraging its capacity to extract context information, not only to fit the WSD task. We also ablate the branch of

Table 2: Ablation results of our model.

Ablation	SE07 F1	Difference
Original Model	65.21	/
No Unsupervised Paraphrasing Loss	64.65	0.56
No Base Similarity Score	64.11	1.10

base similarity φ in the classification scores. The ablation of φ hinders the performance on **SE07** by 1.1 F1 score. This illustrates the effectiveness of utilizing φ to adjust the sense predictions: there is implied information in base classes that calibrates the classical similarity on context representations.

Performance Experiment

To investigate the performance differences caused by the inner setting, we conduct performance experiments with different settings of α. The parameter α exhibits an inverse linear relationship with the impact of base similarity φ, as depicted in Eq. 13. The F1 scores on the testing sets are displayed in the right portion of Fig. 2. The best α value varies by dataset. This suggests that the linear relationship might not be enough to fully utilize the assistance from φ. Therefore, our future research may explore more sophisticated mechanisms, such as Gated Linear Units (GLU), for adjusting the bi-branch classification scores more effectively.

Conclusion

In this work, inspired by the metacognition framework, we proposed a bi-branch meta-learning method on WSD to tackle the long-tail distributions of word senses. Experiments have shown that the bi-branch model has the potential to solve few-shot problems on WSD as it achieves reliable performance with limited resources during training and testing. We infer such performances come from its great generalization ability on samples with both frequent and infrequent senses. This is likely attributed to the enhanced metacognitive knowledge implicated in the base class exploration and data augmentation. In the future, we plan to incorporate the gloss information as the third branch for prediction calibration and to introduce a more sensible strategy for branch adjustment.

Acknowledgements

This work is supported in part by the Natural Science Foundation of China (grant No.62276188), TJU-Wenge joint laboratory funding.

References

Alexander, J. M., Carr, M., & Schwanenflugel, P. J. (1995). Development of metacognition in gifted children: Directions for future research. *Developmental review*, *15*(1), 1–37.

Blevins, T., & Zettlemoyer, L. (2020). Moving down the long tail of word sense disambiguation with gloss-informed biencoders. *arXiv preprint arXiv:2005.02590*.

Chen, H., Xia, M., & Chen, D. (2021). Non-parametric few-shot learning for word sense disambiguation. *arXiv preprint arXiv:2104.12677*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding*.

Dietterich, T. (1995). Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, *27*(3), 326–327.

Dopierre, T., Gravier, C., & Logerais, W. (2021). Protaugment: Unsupervised diverse short-texts paraphrasing for intent detection meta-learning. *arXiv preprint arXiv:2105.12995*.

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, *34*(10), 906.

Hadiwinoto, C., Ng, H. T., & Gan, W. C. (2019). Improved word sense disambiguation using pre-trained contextualized word representations. *arXiv preprint arXiv:1910.00194*.

Holla, N., Mishra, P., Yannakoudakis, H., & Shutova, E. (2020). Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. *arXiv preprint arXiv:2004.14355*.

Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, *44*(9), 5149–5169.

Huang, L., Sun, C., Qiu, X., & Huang, X. (2019). Glossbert: Bert for word sense disambiguation with gloss knowledge. *arXiv preprint arXiv:1908.07245*.

Kaya, M., & Bilge, H. Ş. (2019). Deep metric learning: A survey. *Symmetry*, *11*(9), 1066.

Klein, D. E., & Murphy, G. L. (2002). Paper has been my ruin: Conceptual relations of polysemous senses. *Journal of Memory and Language*, *47*(4), 548–570.

Kumar, S., Jat, S., Saxena, K., & Talukdar, P. (2019). Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 5670–5681).

Li, B., Hou, Y., & Che, W. (2022). Data augmentation approaches in natural language processing: A survey. *Ai Open*, *3*, 71–90.

Livingston, J. A. (2003). Metacognition: An overview.

Luo, F., Liu, T., He, Z., Xia, Q., Sui, Z., & Chang, B. (2018). Leveraging gloss knowledge in neural word sense disambiguation by hierarchical co-attention. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 1402–1411).

Navigli, R. (2009). Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, *41*(2), 1–69.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... others (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Raganato, A., Bovi, C. D., & Navigli, R. (2017). Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 1156–1167).

Raganato, A., Camacho-Collados, J., Navigli, R., et al. (2017). Word sense disambiguation: a uinified evaluation framework and empirical comparison. In *Proceedings of the 15th conference of the european chapter of the association for computational linguistics: Volume 1, long papers* (Vol. 1, pp. 99–110).

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, *30*.

Sun, P., Ouyang, Y., Zhang, W., & Dai, X. (2021). Meda: Meta-learning with data augmentation for few-shot text classification. In *Ijcai* (pp. 3929–3935).

Vial, L., Lecouteux, B., & Schwab, D. (2019). Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. *arXiv preprint arXiv:1905.05677*.

Wang, Z., Zhao, Y., Li, J., & Tian, Y. (2020). Cooperative bi-path metric for few-shot learning. In *Proceedings of the 28th acm international conference on multimedia* (pp. 1524–1532).

Wenden, A. L. (1998). Metacognitive knowledge and language learning1. *Applied linguistics*, *19*(4), 515–537.