Bi-matching Mechanism to Combat Long-tail Senses of Word Sense Disambiguation

Junwei Zhang[2,1], Ruifang He[1,2(✉)], and Fengyu Guo[3(✉)]

[1] Tianjin Key Laboratory of Cognitive Computing and Application, College of Intelligence and Computing, Tianjin University, Tianjin, China.
{junwei,rfhe}@tju.edu.cn
[2] State Key Laboratory of Communication Content Cognition, People's Daily Online, Beijing, China.
[3] College of Computer and Information Engineering, Tianjin Normal University, Tianjin, China. fyguo@tjnu.edu.cn

Abstract. The long-tail phenomenon of word sense distribution in linguistics causes Word Sense Disambiguation (WSD) to face both head senses with a large number of samples and tail senses with only a few samples. Traditional recognition methods are suitable for head senses with sufficient training samples, but they cannot effectively deal with tail senses. Inspired by the diverse memory and recognition abilities of children's linguistic behavior, we propose a bi-matching mechanism approach for WSD. Considering that tail senses are often presented in the form of fixed collocations, a collocation feature matching method suitable for tail senses is designed; the traditional definition matching method is used for head senses; finally, the two matching methods are combined to construct a WSD model with the bi-matching mechanism (called Bi-MWSD). Bi-MWSD can effectively combat the difficulty of identifying the tail senses due to insufficient training samples. The experiments are implemented in the standard English all-words WSD evaluation framework and the training data augmented evaluation framework. The experimental results outperform the baseline models and achieve state-of-the-art performance under the data augmentation evaluation framework.

Keywords: Word Sense Disambiguation · Long Tail Senses · Bi-matching Mechanism.

1 Introduction

Word Sense Disambiguation (WSD) is to assign the correct sense to the target word according to the given context [1, 2]. WSD occupies an important position in the field of Natural Language Processing (NLP) [3], and the correct identification of word senses has a direct and profound impact on subsequent semantic understanding tasks, such as machine translation [4, 5] and natural language understanding [6, 7].

However, due to the long-tail phenomenon of word sense distribution in linguistics, the WSD model needs to face both head senses with a large number of

samples and tail senses with only a few samples [8, 9]. For example, the verb form of the word *Play*[4] has 35 senses in WordNet 3.1, of which the most commonly used is "*Participate in games or sports*", and the vast majority are rarely used tail senses, such as "*Contend against an opponent in a sport, game, or battle*". In addition, due to the long-tail phenomenon of vocabulary usage frequency in linguistics, the occurrence frequency of tail senses is severely reduced, which makes it more difficult for the WSD model to identify long-tail senses. Note that the long-tail senses here refer to the tail senses under the long-tailed distribution.

Traditional recognition methods can effectively deal with head senses with sufficient training samples, but it is difficult to take into account tail senses with insufficient training samples. BEM, proposed by Blevins et al. [10], attempts to employ BERT [11] to obtain a context-based embedding of the target word, and then determines possible sense by calculating the similarity between this embedding and the textual embedding of each gloss. For head senses, this method can obtain effective sense representations, but for tail senses, it is difficult to obtain highly recognizable representations. The reason is that embeddings of all senses can be easily obtained based on glosses, but it is difficult to effectively improve the accuracy of embeddings when training samples are lacking or not. GlossBERT, proposed by Huang et al. [12], combines the sentence containing the target word with each gloss separately to obtain shared embeddings, and then treats the WSD task as a sentence-level classification task to achieve word sense recognition. This method has similar drawbacks to BEM, that is, it is difficult to obtain reliable representations when training samples are lacking or not. In addition, some researchers attempt to treat the WSD task as a few-shot learning problem to deal with insufficient training samples for tail senses. For example, Holla et al. [13] propose a meta-learning framework to deal with few-shot WSD, which aims to learn features from labeled instances to disambiguate unseen words. See also Refs. [14, 8, 9].

Inspired by the diverse memory and recognition abilities of children's linguistic behavior [15] (see Sec. 3.2 for a detailed analysis), we propose a bi-matching mechanism approach for WSD. Analysis of a large number of tail senses finds that tail senses are mostly presented in the form of fixed collocations, that is, they often appear together with fixed words or often appear in fixed contexts. This is also the main reason for insufficient samples of tail senses. Considering that the collocation words of tail senses are fixed, and the collocation words are clear, this paper proposes a collocation feature matching method to combat the challenge of insufficient training samples of tail senses. This paper extracts collocation words from the example sentences provided by the corresponding word senses in the dictionary, and collectively calls them the collocation feature. When there are multiple example sentences, the collocation feature integrates all the collocation words in the example sentences; when there is no example sentence, the collocation feature directly uses the gloss instead. Considering the outstanding performance of definition matching in traditional recognition methods, this paper adopts traditional definition matching to deal with head senses.

[4] http://wordnetweb.princeton.edu/perl/webwn?s=play

Finally, the two matching methods together constitute a WSD model with the bi-matching mechanism.

The contributions of this paper are summarized as follows:

- By mining the characteristics of long-tail senses, a collocation feature matching method against insufficient training samples of tail senses is proposed.
- Inspired by the diverse memory and recognition abilities of children's linguistic behavior, a WSD model with the bi-matching mechanism is constructed, which fills the gap of using different matching methods for head and tail senses.
- The experiments are carried out under the evaluation framework of English all-words WSD, and the experimental results are better than the baseline models. Moreover, state-of-the-art performance is achieved under data-augmented evaluation framework.

Codes and pre-trained models are available at https://github.com/yboys0504/wsd.

2 Related Work

In the early development of WSD, researchers did not focus on long-tail senses, but more on dealing with all senses by adopting a unified approach. During this period, WSD models used a single recognition method to complete the recognition process at the end of the model [3, 1]. These recognition methods are also often used in other tasks in NLP, so we call them **traditional recognition methods**. Subsequently, with the continuous improvement of the overall level of WSD models, long-tail senses became the bottleneck of development, and researchers began to focus on **few-shot learning methods** to combat long-tail senses [14, 16].

2.1 Traditional Recognition Methods for WSD

According to the classical classification method, WSD models can be roughly divided into two categories, namely supervised models and knowledge-based models.

Supervised models usually employ a deep network structure to process the target word with context, and connect a classifier at the end of the model to calculate the probability of each sense [17, 18]. For example, Recurrent Neural Network (RNN) suitable for sequence features is often used to build the core network structure of the WSD models, and a fully connected layer with normalization constraints is added as a classifier in the output layer [19, 20]. Subsequent WSD models based on pre-trained language models only replace the core network structure with pre-trained models, but the classifiers are still implemented using a traditional fully connected layer [21–23]. The reason why supervised models are accustomed to this design is that the model can be trained end-to-end as a whole.

Knowledge-based models attempt to employ external knowledge to improve the recognition rate of WSD models, such as dictionary knowledge [10, 24], semantic network knowledge [25, 27], and multilingual knowledge [28, 21]. Among them, glosses in the dictionary are often trained as text embeddings to replace word sense labels [10, 26, 9]. Such definition matching methods are good for identifying head senses, but they are not good for identifying tail senses. The fundamental reason is that tail senses often appear in the form of fixed collocations and they are difficult to give a clear definition.

2.2 Few-shot Learning Methods for WSD

Subsequently, the researchers realized the importance of long-tail senses in WSD, and adopted some targeted solutions for tail senses, such as meta-learning, zero-shot learning, reinforcement learning, etc. Holla et al. [13] proposed a meta-learning framework for few-shot WSD, where the goal is to learn features from labeled instances to disambiguate unseen words. See also Refs. [14, 16]. Blevins et al. [10] noticed the long-tail phenomenon of word sense distribution, and proposed a dual encoder model, that is, one BERT is used to extract the word embedding of the target word with contextual information, and another BERT is used to obtain the text embeddings of the glosses. The innovation of this work is that the model adopts a joint training mechanism of dual encoders, but the disadvantage is that the model still adopts a single matching method to deal with both head and tail senses.

3 Methodology

In this section, we first formalize the WSD task, then clarify the cognitive basis of the bi-matching mechanism derived from children's literacy behavior, and finally describe the structure of our model in the formal language.

3.1 Word Sense Disambiguation

WSD is to predict the senses of the target word in a given context [1, 2]. The formal definition can be expressed as: the possible sense $s \in S_{\hat{w}}$ of the target word $\hat{w}$ in the given context $C_{\hat{w}}$ is formally described as

$$f(\hat{w}, C_{\hat{w}}) = s \in S_{\hat{w}} \tag{1}$$

where $f(\cdot)$ refers to the WSD model, and $S_{\hat{w}}$ is the candidate list of the senses of the target word.

All-words WSD is to predict all ambiguous words in a given context [1, 2]. This means that the WSD model may predict the noun, verb, adjective, and adverb forms of ambiguous words. In this case, the input and output of the WSD model are defined as $C = (..., w_i, ...)$ and $S = (..., s^x_{w_i}, ...)$, respectively, where $s^x_{w_i}$ represents the x^{th} sense of the target word w_i.

3.2 Cognitive Basis of Bi-matching Mechanism

Masaru Ibuka [15], a Japanese educator, pointed out that children's literacy behavior is mainly based on mechanical memory and recognition ability in the early stage, and then gradually develops concept-oriented memory and recognition ability in the later stage. The mechanical method rigidly remembers the structure of the word itself and its application scenarios, such as collocation features of words. The concept-oriented method establishes the relationship between the structure, meaning, and usage of words through analysis and comparison, such as the definitions given in the dictionary.

For the WSD task, we should not only pay attention to head senses with a large number of samples, but also tail senses with only a few samples, because long-tail senses are an important bottleneck for the development. For head senses, it is reasonable to distinguish senses through the definition system, because theoretically, the definition system of word senses can clearly distinguish different head senses. But for tail senses, it is difficult to define a clear and non-confusing definition system for each sense. For example, "*Go to plant fish*", where the word *plant* means "*Place into a river*". This sense of the word *plant* mostly appears in such a collocation form. Therefore, considering the characteristics of tail senses, the collocation feature matching method is more suitable for identifying tail senses.

In this paper, we propose a bi-matching mechanism approach to construct a WSD model (called **Bi-MWSD**), namely the **collocation feature matching method** for tail senses and the **definition matching method** for head senses. We describe the construction details and operation process of Bi-MWSD in Sec. 3.3.

3.3 Bi-matching Mechanism for WSD

The architecture of Bi-MWSD is shown in Fig. 1. Bi-MWSD uses two pre-trained language models as text feature encoders, and the pre-trained model adopts the widely used BERT [11]. One encoder is used to extract the collocation features of the target word in the training samples and the example sentences, which is called the **collocation feature encoder**. The other is used to learn the definition system in the glosses of the target word, which is called the **definition encoder**. The example sentences and glosses come from the examples and definitions corresponding to each sense in WordNet. The last step is the matching process of head senses and tail senses, which is called **word sense matching**.

Collocation Feature Encoder: The function of the collocation feature encoder is to memorize the collocation features of the target word, such as the structure and relationship between the target word and the collocation words, and the entire application scenario. The encoder process two kinds of texts:

– One is the example sentences corresponding to each sense of the target word in WordNet, $E^x = (..., e^x_k, ...)$ where e^x_k represents the k^{th} word of the example sentence E^x of the x^{th} sense of the target word.

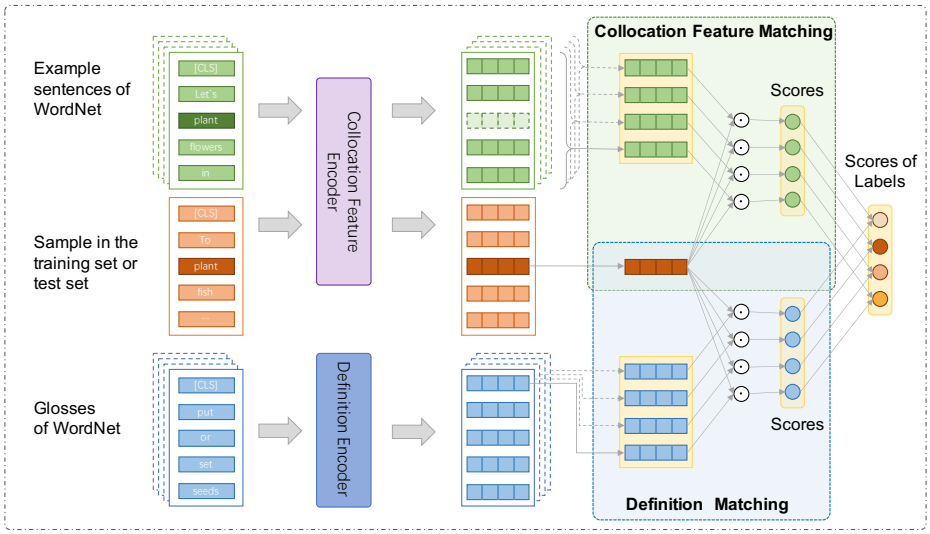

Fig. 1. Schematic diagram of the Bi-MWSD architecture, which illustrates the disambiguation process of the target word *Plant*. The collocation feature encoder is used to encode target words and example sentences; the definition encoder is only used to encode glosses. The symbol $\odot$ represents the dot product of matrices.

– And the other is the training samples containing the target word, $C = (..., w_i, ...)$ where w_i represents the i^{th} word.

The texts are encoded using BERT standard processing rules, that is, adding $[CLS]$ and $[SEP]$ marks at the beginning and end of the text respectively, such as

$$E^x = ([CLS], ..., e^x_k, ..., [SEP]) \tag{2}$$
$$= (e^x_{cls}, ..., e^x_k, ..., e^x_{sep}). \tag{3}$$

The processing method of the training samples is also the same. The encoder encodes each word, including the added $[CLS]$ and $[SEP]$, to obtain a corresponding 768-dimensional vector.

The reason why we use one encoder to process two kinds of texts here is that both the example sentences and the training samples contain the target word, which can all be considered that there are collocation features of the target word. Moreover, the advantage of this processing is that the training sample will truly reflect the frequency of each sense of the target word, and the example sentences can provide the collocation features of tail senses. Processing them together can make up for the lack of scene information of tail senses, but it will not (seriously) change their frequency. In WordNet 3.0, sometimes multiple example sentences are given for one sense, and we integrate all the example sentences by default; when no example sentences are given, we use the embedded representation of the gloss instead.

After processing by the collocation feature encoder, we can get the vector representation of the target word in the training sample, which is defined as $v_{\hat{w}}$, and the vector representation of the collocation features of each sense x provided by the example sentences, which is defined as V_{E^x}. $v_{\hat{w}}$ is the vector representation corresponding to the target word in the output of the pretrained model BERT. For V_{E^x}, we here provide two calculation methods, namely the overall text vector minus the target word vector,

$$V_{E^x} = v_{e^x_{cls}} - v_{e^x_{\hat{w}}}, \tag{4}$$

and the vectors except the target word vector are added,

$$V_{E^x} = \sum_k v_{e^x_k} - v_{e^x_{\hat{w}}}. \tag{5}$$

Through experimental analysis of these two methods, we find that the first one is relatively better. The possible reason is that it can not only characterize the collocation features of the target word, but also remember the entire text, namely the application scenario.

Definition Encoder: The definition encoder constructs the definition system of the target word by learning the glosses G^x for each sense x in WordNet, $G^x = (..., g^x_j, ...)$ where g^x_j represents the j^{th} word of the gloss text of the x^{th} sense of the target word. The glosses are simple and accurate generalizations of word senses and are therefore suitable for refining the definition system of the target word. What needs to be emphasized here is that the target word itself is not included in the glosses, so glosses cannot be used to extract the collocation features of the target word. Following standard processing rules of BERT, $[CLS]$ and $[SEP]$ marks are also added for the glosses,

$$G^x = ([CLS], ..., g^x_j, ..., [SEP]) \tag{6}$$

$$= (g^x_{cls}, ..., g^x_j, ..., g^x_{sep}). \tag{7}$$

The encoder encodes each word, including the added $[CLS]$ and $[SEP]$, to obtain a corresponding 768-dimensional vector. Here we choose the output vector corresponding to $[CLS]$, i.e., $v_{g^x_{cls}}$, to represent the entire gloss text, i.e., $V_{G^x} = v_{g^x_{cls}}$. This method is a common practice in the industry.

Word Sense Matching: At this point, we can calculate the score of each sense of the target word $\hat{w}$ in a given context C,

$$Score(\hat{w}|C) = F(\{v_{\hat{w}} \odot (\alpha V_{G^x} + \beta V_{E^x})\}^x) \tag{8}$$

where α and β respectively represent the proportion of the definition matching method and the collocation feature matching method. $F(\cdot)$ can be a standard *Softmax* or other distribution function. When $F(\cdot)$ is selected as *Softmax*,

$Score(\hat{w}|C)$ is a probability distribution of all senses of the target word in a given context. Finally, we can conclude that the one with the highest probability is the most likely sense.

Here α and β can be the weights learned by the model itself, or they can be the proportions of each sense provided by WordNet. Through experimental analysis, we find that they work best when they are set to the same value. It needs to be explained that it is difficult to know in advance which sense of the target word is, so it is appropriate to use the equal probability method, that is, the possibility of the head sense or the tail sense is the same.

Parameter Optimization: We use a cross-entropy loss on the scores of the candidate senses of the target word to train Bi-MWSD. The loss function is

$$Loss(Score, index) \tag{9}$$

$$= -\log\left(\frac{\exp(Score^{[index]})}{\sum_{i=1}\exp(Score^{[i]})}\right) \tag{10}$$

$$= -Score^{[index]} + \log\sum_{i=1}\exp(Score^{[i]}) \tag{11}$$

where *index* is the index of the list of the candidate senses of the target word.

Bi-MWSD employs an Adam optimizer [29] to update the parameters of the model, and the specific settings of the optimizer are given in the experimental section.

4 Experiments

4.1 Datasets and Evaluation Metrics

Bi-MWSD adopts the unified evaluation framework of English all-words WSD proposed by Raganato et al. [1] to implement training and evaluation. In the **standard evaluation experiment**, the training set is SemCor[5]; in the **evaluation experiment under data augmentation**, the training set is SemCor and WNGT[6] (WordNet Gloss Tagged). Following common practice, SemEval-2007 (SE07; [30]) is designated as the development set, and Senseval-2 (SE2; [31]), Senseval-3 (SE3; [32]), SemEval-2013 (SE13; [33]), and SemEval-2015 (SE15; [34]) are used as the test sets. The statistical information of each dataset is shown in Tab. 1. Also, we concatenate the development set and all the test sets to reconstruct the test sets of verbs (V), nouns (N), adjectives (A), and adverbs (R), and treat them as a whole as a test set (**ALL**).

In this paper, we select all word senses in WordNet 3.0 [35] as candidate senses of the target word. All experimental results in the figures and tables are reported as a percentage of the F1-score.

[5] http://lcl.uniroma1.it/wsdeval/training-data
[6] https://wordnetcode.princeton.edu/glosstag.shtml

Table 1. Statistics of the datasets: the number of documents (Docs), sentences (Sents), tokens (Tokens), sense annotations, sense types covered, annotated lemma types covered and ambiguity level in each dataset, where the ambiguity level implies the difficulty of the dataset.

Dataset	Docs	Sents	Tokens	Annotations	Sense types	Lemma types	Ambiguity
SE2	3	242	5,766	2,282	1,335	1,093	5.4
SE3	3	352	5,541	1,850	1,167	977	6.8
SE07	3	135	3,201	455	375	330	8.5
SE13	13	306	8,391	1,644	827	751	4.9
SE15	4	138	2,604	1,022	659	512	5.5

4.2 Baseline Models

To evaluate the comprehensive performance of Bi-MWSD in the community, we select state-of-the-art models in the past three years, including LMMS [36], EWISE [9], and GlossBERT [12] in 2019, SREF [37], ARES [26], EWISER [38], BEM [10], and SparseLMMS [39] in 2020, and COF [40], ESR [41], Multi-Label [42], and SACE [43] in 2021. All experimental results of the above models are taken from the data published in the original paper.

From these, we select three most comparable models as baseline models, which are GlossBERT [12] with similar external resources, BEM [10] with similar framework structure, and Multi-Label [42] with multi-label classification method. GlossBERT and BEM employ typical and traditional word sense recognition methods. GlossBERT employs a fully connected layer with normalization constraints as the output layer of the model. BEM implements word sense matching by calculating the similarity between the target word vector and the definition vectors. Multi-Label designs the WSD model as a multi-label classification task. Although this method has the ability to match multiple times, it is not the same as the bi-matching mechanism proposed in this paper.

In addition, we select three models as baselines for the evaluation experiment under data augmentation, which are SparseLMMS [39], EWISER [38], and ESR [41].

4.3 Experimental Setting

The hardware platform of Bi-MWSD is Ubuntu 18.04.3, which installs two GPUs whose version is NVIDIA Tesla P40. The development platform is Python 3.8.3[7], and the learning framework is Pytorch 1.8.1[8]. The pre-trained language model is provided by Transformers 4.5.1[9]. Under the **standard evaluation experiment**, the encoders of Bi-MWSD use *BERT-base-uncased*; under the **evaluation experiment of data augmentation**, the encoders of Bi-MWSD use

[7] https://www.python.org/
[8] https://pytorch.org/
[9] https://huggingface.co/transformers/v4.5.1/

BERT-large-uncased. The hyperparameter *Learning Rate*, *Context Batch Size*, *Gloss Batch Size*, *Epochs*, *Context Maximum Length* and *Gloss Maximum Length* of the model are set to $[1E\text{-}5, 5E\text{-}6, 1E\text{-}6]$, 4, 256, 20, 128 and 32, respectively. Super-parameters not listed are given in the published code.

Table 2. F1-score (%) on the English all-words WSD task. Dev refers to the development set, and *N*, *V*, *A*, *R*, and **ALL** refer to the nouns, verbs, adjectives, adverbs, and overall datasets constructed by concatenating the development set and the test sets, respectively. The experimental results are organized according to the standard evaluation experiment (that is, Training data: SemCor) and the evaluation experiment under data augmentation (that is, Training data: SemCor + WNGT). The underlined and the bolded results refer to the overall and the regional best results, respectively.

Model	**Dev**	**Test sets**				**Concatenation**				
	SE07	SE2	SE3	SE13	SE15	*N*	*V*	*A*	*R*	**ALL**
Training data: SemCor										
Prior work										
LMMS (ACL, 2019, [36])	68.1	76.3	75.6	75.1	77.0	-	-	-	-	75.4
EWISE (ACL, 2019, [9])	67.3	73.8	71.1	69.4	74.5	74.0	60.2	78.0	82.1	71.8
SREF (EMNLP, 2020, [37])	72.1	78.6	76.6	78.0	80.5	80.6	66.5	82.6	84.4	77.8
ARES (EMNLP, 2020, [26])	71.0	78.0	77.1	77.3	83.2	80.6	68.3	80.5	83.5	77.9
EWISER (ACL, 2020, [38])	71.0	78.9	78.4	78.9	79.3	81.7	66.3	81.2	85.8	78.3
COF (EMNLP, 2021, [40])	69.2	76.0	74.2	78.2	80.9	80.6	61.4	80.5	81.8	76.3
ESR (EMNLP, 2021, [41])	75.4	80.6	78.2	79.8	82.8	82.5	69.5	82.5	87.3	79.8
SACE (ACL, 2021, [43])	74.7	80.9	79.1	82.4	84.6	83.2	71.1	85.4	87.9	80.9
Baseline models										
GlossBERT (EMNLP,2019,[12])	72.5	77.7	75.2	76.1	80.4	79.8	67.1	79.6	87.4	77.0
BEM (ACL, 2020, [10])	74.5	79.4	77.4	79.7	**81.7**	81.4	68.5	**83.0**	**87.9**	79.0
Multi-Label (EACL, 2021, [42])	72.2	78.4	77.8	76.7	78.2	80.1	67.0	80.5	86.2	77.6
Bi-MWSD	**75.2**	**80.2**	**78.0**	**79.8**	81.4	**82.8**	**69.5**	82.5	87.5	**79.4**
Training data: SemCor + WNGT										
SparseLMMS (EMNLP,2020,[39])	73.0	79.6	77.3	79.4	81.3	-	-	-	-	78.8
EWISER (ACL, 2020, [38])	75.2	80.8	79.0	80.7	81.8	81.7	66.3	81.2	85.8	80.1
ESR (EMNLP, 2021, [41])	**77.4**	**81.4**	78.0	81.5	**83.9**	83.1	71.1	**83.6**	**87.5**	80.7
Bi-MWSD$_{\text{large}}$	77.3	80.8	**79.9**	**83.8**	83.7	**84.0**	**71.7**	81.5	86.5	**81.5**

4.4 Experimental Results

The experimental results are shown in Tab. 2, where according to common practice, all results are presented as a percentage of the F1-score. The experimental results are organized according to the **standard evaluation experiment** and the **evaluation experiment under data augmentation**.

– **In the standard evaluation experiment**, compared with previous work, Bi-MWSD is in an upper-middle position; compared with baseline models,

Bi-MWSD achieves state-of-the-art in multiple metrics. The experimental results confirm that the bi-matching mechanism is indeed beneficial to improve the recognition ability of the model. Compared with GlossBERT [12], it shows that the matching mechanism of Bi-MWSD is superior to the recognition method constructed by a fully connected layer with normalization constraints. The possible reason is that the recognizer constructed by a fully connected layer has a large number of parameters that need to be learned, and the lack of training samples of long-tail senses makes it difficult to learn the parameters effectively. Compared with BEM [10], it shows that the bi-matching mechanism of Bi-MWSD will improve the recognition ability compared with the single-matching mechanism model with a similar structure. For the contribution of the collocation feature matching method, we will give an analysis in the ablation study.

– **In the evaluation experiment under data augmentation**, Bi-MWSD also achieves state-of-the-art performance in multiple metrics, indicating that Bi-MWSD has great potential. Moreover, it also shows that when the training sample size of tail senses is expanded, it is beneficial to improve the performance of Bi-MWSD.

Analysis of poor performance on indicators A (adjectives) and R (adverbs) of Tab. 2: In linguistics, nouns and verbs are words with a serious long-tail, and adjectives and adverbs are relatively weaker. In other words, there are fewer tail senses in adjectives and adverbs. For datasets where the proportion of tail senses is not high, the method of not distinguishing or ignoring tail senses has advantages.

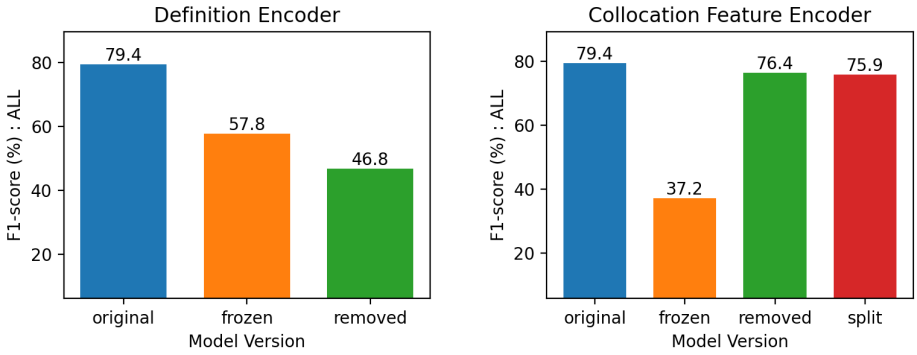

Fig. 2. Experimental results of ablation studies on the definition encoder and the collocation feature encoder. All values are experimental results under the test set **ALL** and are presented as a percentage of the F1-score.

4.5 Ablation Study

Bi-MWSD employs a bi-matching mechanism to replace the traditional single-matching mechanism of the WSD model, namely **definition matching** and **collocation feature matching**. To clarify the contribution of various matching mechanisms to the overall representation, and to determine their value for the target task, we perform ablation experiments.

Ablation Study for Definition Matching: For the analysis of the definition matching mechanism, we use the method of **ablation function** (i.e., freeze the encoder) and **ablation module** (i.e., directly remove the encoder). The method of freezing the encoder will prevent the encoder from fine-tuning the parameters on the training set, that is, preventing the encoder from learning more semantic information on the training set. We know that tail senses are marked in the training set. Preventing the encoder from fine-tuning the parameters on the training set will hinder the encoder's ability to recognize tail senses. Compared with the original model, this method will directly reflect the contribution of the definition encoder to solving tail senses. The method of removing the encoder is more direct, which directly reflects the contribution of the definition matching method to the overall representation.

We separately freeze and remove the definition encoder on the original model, and adjust the hyperparameters to get the best results. The experimental results are shown in Fig. 2.

1. Comparing the original version and the frozen version, it can be seen that the definition encoder can indeed learn new semantic knowledge by fine-tuning the parameters on the training set, and it can greatly improve the overall representation.
2. Comparing the original version and the removed version, it can be seen that the contribution of the definition encoder to the overall representation is huge. This result is in line with reality, because head senses are indeed far greater than the usage rate of tail senses in life, and the function of the definition encoder is reflected in the recognition of head senses. Again, comparing the frozen version with the deleted version confirms this conclusion.

Ablation Study for Collocation Feature Matching: For the analysis of the collocation feature matching mechanism, in addition to the **ablation function** and **ablation module**, we also need to **disassemble the two functions** of the collocation feature encoder, that is, target word vectorization and example sentence vectorization. It should be emphasized that the removed version here only removes the example sentence learning function of the encoder.

We fine-tune the hyperparameters of the modified versions to obtain the best results. The experimental results are shown in Fig. 2.

1. Comparing the original version and the frozen version, it can be seen that the model shows the worst case without fine-tuning the parameters under the

training set. The main reason is that the encoder is responsible for the learning of the target word vector. If there is no good target word representation, it will directly affect the overall representation.

2. Comparing the original version with the removed version, that is, removing the collocation feature matching method, it can be seen that introducing this matching mechanism can indeed improve the effectiveness of the model. Although there is only two percentage point improvement, considering the difficulty of tail sense recognition, it also shows that the bi-matching mechanism does contribute to the recognition of tail senses.
3. Regarding whether the training process of merging the target word and the collocation feature can improve the overall representation of the model, we can compare the results of the original version and the split version. An improvement of close to 3% proves that this design is reasonable. Example sentences of tail senses in the dictionary improve the ability of the pre-trained model to represent low-tail words.

5 Experiments under Head and Tail Senses

To confirm the effectiveness of the bi-matching mechanism for various word senses, namely, head senses and tail senses, we conduct experiments under the reconstructed head sense and tail sense test sets respectively. The ablation experiments focus more on analyzing the effectiveness of each module, while the experiments here can more clearly present the specific contribution of the bi-matching mechanism to various word senses.

Datasets: The training set and development set still employ the settings of the standard evaluation experiment. The test sets are divided into head sense (HS) and tail sense (TS) datasets obtained by reconstructing **ALL**.

- The construction method of the head sense datasets is to obtain the dataset by **removing** the specified word sense samples in **ALL**. We construct two head sense datasets: a dataset constructed by removing data with only one sample (called Removed 1-shot TS); and a dataset constructed by removing data with less than three samples (called Removed 2-shot TS).
- The construction method of the tail sense datasets is to obtain the dataset by **retaining only** the specified word sense samples in **ALL**. We construct two tail sense datasets: a dataset constructed by retaining only data with only one sample (called Retained 1-shot TS); a dataset constructed by retaining only data with less than three samples (called Retained 2-shot TS).

Experimental Setting and Baseline Models: The experimental setting is still carried out according to the setting method of the standard evaluation experiment. The baseline models select the most comparable GlossBERT [12] and BEM [10] as the control group. Bi-MWSD adopts the setup of the standard evaluation experiment.

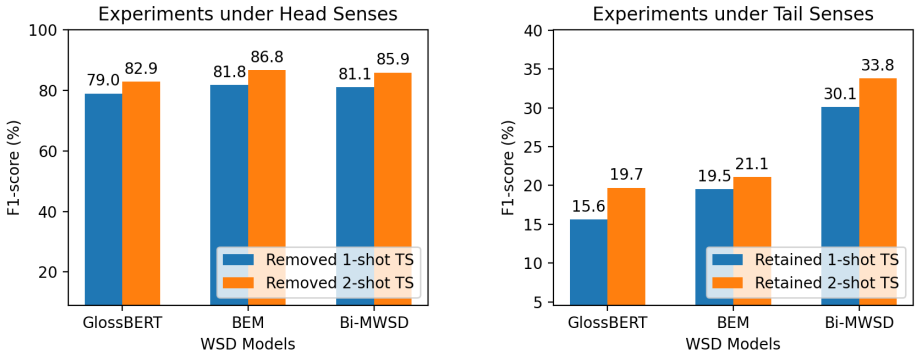

Fig. 3. Experimental results on the head sense and the tail sense datasets reconstructed by **ALL**. All values are presented as a percentage of the F1-score. *Removed ∗-shot TS* and *Retained ∗-shot TS* refer to different kinds of head sense (HS) and tail sense (TS) datasets, respectively.

5.1 Bi-MWSD for Head Senses

The experimental results under the head sense datasets are shown in Fig. 3. From the overall data performance, Bi-MWSD outperforms GlossBERT but is inferior to BEM on both head sense datasets, indicating that the bi-matching mechanism is stronger than the single-matching mechanism constructed by the fully connected layer but weaker than the single-matching mechanism constructed by the definition identification method on datasets with all head senses. This conclusion shows that there is a certain interference between the double matching mechanisms, and it is difficult to obtain the best performance when only one class of word senses is processed.

5.2 Bi-MWSD for Tail Senses

The experimental results under the tail sense datasets are shown in Fig. 3. From the overall data performance, Bi-MWSD outperforms the control models on both tail sense datasets, indicating that the bi-matching mechanism has significant advantages in dealing with tail senses. This conclusion fully proves that the collocation feature matching method can effectively deal with the long-tail senses; the multi-matching mechanism (not limited to the bi-matching mechanism proposed in this paper) can be used to achieve the purpose of dealing with various word senses in a targeted manner.

6 Conclusion

Inspired by the diverse memory and recognition abilities of children's linguistic behavior, this paper proposes a method of bi-matching mechanism to deal with the head and tail senses in Word Sense Disambiguation (WSD). We design

a collocation feature matching method for tail senses, and leverage traditional definition matching method to deal with head senses, which together constitute a WSD model with the bi-matching mechanism (called Bi-MWSD). Bi-MWSD can effectively combat the difficulty of insufficient tail sense training samples caused by the long tail distribution of word sense. In addition, Bi-MWSD outperforms baseline models and achieves state-of-the-art performance under data-augmented evaluation framework. The contribution of this work is to fill the gap of bi-matching mechanism in WSD, and moreover explore the feasibility of bi-matching mechanism against insufficient training samples.

In future work, we will build a hierarchical multi-matching mechanism to better address the imbalance of training samples caused by the long-tailed phenomenon of word sense distribution. Moreover, we will further subdivide the word senses, and employ this multi-matching method to deal with various word senses in a targeted manner to improve the accuracy of word sense recognition.

Acknowledgements Our work is supported by the National Natural Science Foundation of China (61976154), the National Key R&D Program of China (2019YFC1521200), the State Key Laboratory of Communication Content Cognition, People's Daily Online (No. A32003), and the National Natural Science Foundation of China (No. 62106176).

References

1. Navigli, R., Camacho-Collados, J., Raganato, A.: Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. EACL (2017).
2. Navigli, R.: Word sense disambiguation: A survey. ACM Comput. Surv. (2009).
3. Bevilacqua, M., Pasini, T., Raganato, A., Navigli, R.: Recent Trends in Word Sense Disambiguation: A Survey. IJCAI (2021).
4. Neale, S., Gomes, L.-M., Agirre, E., Lacalle, O.-L., Branco, A.-H.: Word Sense-Aware Machine Translation: Including Senses as Contextual Features for Improved Translation Models. LREC (2016).
5. Rios Gonzales, A., Mascarell, L., Sennrich, R.: Improving Word Sense Disambiguation in Neural Machine Translation with Sense Embeddings. WMT (2017).
6. Dewadkar, D.-A., Haribhakta, Y.-V., Kulkarni, P.-A., Balvir, P.-D.: Unsupervised Word Sense Disambiguation in Natural Language Understanding. ICAI (2010).
7. Mills, M.-T., Bourbakis, N.-G.: Graph-Based Methods for Natural Language Processing and Understanding—A Survey and Analysis. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2014).
8. Li, W., Madabushi, H.-T., Lee, M.-G.: UoB_UK at SemEval 2021 Task 2: Zero-Shot and Few-Shot Learning for Multi-lingual and Cross-lingual Word Sense Disambiguation. SEMEVAL (2021).
9. Kumar, S., Jat, S., Saxena, K., Talukdar, P.-P.: Zero-shot Word Sense Disambiguation using Sense Definition Embeddings. ACL (2019).
10. Blevins, T., Zettlemoyer, L.: Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. ACL (2020).
11. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. NAACL (2019).

12. Huang, L., Sun, C., Qiu, X., Huang, X.: GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. EMNLP (2019).
13. Holla, N., Mishra, P., Yannakoudakis, H., Shutova, E.: Learning to Learn to Disambiguate: Meta-Learning for Few-Shot Word Sense Disambiguation. EMNLP (2020).
14. Du, Y., Holla, N., Zhen, X., Snoek, C.-G., Shutova, E.: Meta-Learning with Variational Semantic Memory for Word Sense Disambiguation. ACL (2021).
15. Ibuka, M.: Kindergarten is Too Late!. Souvenir Press London (1977).
16. Chen, H., Xia, M., Chen, D.: Non-Parametric Few-Shot Learning for Word Sense Disambiguation. NAACL (2021).
17. Yuan, D., Richardson, J., Doherty, R., Evans, C., Altendorf, E.: Semi-supervised Word Sense Disambiguation with Neural Models. COLING (2016).
18. Raganato, A., Bovi, C.-D., Navigli, R.: Neural Sequence Learning Models for Word Sense Disambiguation. EMNLP (2017).
19. Le, M.-N., Postma, M., Urbani, J., Vossen, P.: A Deep Dive into Word Sense Disambiguation with LSTM. COLING (2018).
20. Kågebäck, M., Salomonsson, H.: Word Sense Disambiguation using a Bidirectional LSTM. COLING (2016).
21. Scarlini, B., Pasini, T., Navigli, R.: SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. AAAI (2020).
22. Hadiwinoto, C., Ng, H.-T., Gan, W.-C.: Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations. EMNLP (2019).
23. Du, J., Qi, F., Sun, M.: Using BERT for Word Sense Disambiguation. ArXiv, abs/1909.08358 (2019).
24. Luo, F., Liu, T., Xia, Q., Chang, B., Sui, Z.: Incorporating Glosses into Neural Word Sense Disambiguation. ACL (2018).
25. Fernandez, A.-D., Stevenson, M., Martínez-Romo, J., Araujo, L.: Co-occurrence graphs for word sense disambiguation in the biomedical domain. Artificial intelligence in medicine (2018).
26. Scarlini, B., Pasini, T., Navigli, R.: With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All-Round Word Sense Disambiguation. EMNLP (2020).
27. Dongsuk, O., Kwon, S., Kim, K., Ko, Y.: Word Sense Disambiguation Based on Word Similarity Calculation Using Word Vector Representation from a Knowledge-based Graph. COLING (2018).
28. Pasini, T.: The Knowledge Acquisition Bottleneck Problem in Multilingual Word Sense Disambiguation. IJCAI (2020).
29. Kingma, D.-P., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980 (2015).
30. Pradhan, S., Loper, E., Dligach, D., Palmer, M.: SemEval-2007 Task-17: English Lexical Sample, SRL and All Words. Fourth International Workshop on Semantic Evaluations (2007).
31. Edmonds, P., Cotton, S.: SENSEVAL-2: Overview. *SEMEVAL (2001).
32. Snyder, B., Palmer, M.: The English all-words task. ACL (2004).
33. Navigli, R., Jurgens, D., Vannella, D.: SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. *SEMEVAL (2013).
34. Moro, A., Navigli, R.: SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. *SEMEVAL (2015).
35. Fellbaum, C.-D.: WordNet : an electronic lexical database. Language (2000).
36. Loureiro, D., Jorge, A.-M.: Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation. ACL (2019).

37. Wang, M., Wang, Y.: A Synset Relation-enhanced Framework with a Try-again Mechanism for Word Sense Disambiguation. EMNLP (2020).
38. Bevilacqua, M., Navigli, R.: Breaking Through the 80% Glass Ceiling: Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information. ACL (2020).
39. Berend, G.: Sparsity Makes Sense: Word Sense Disambiguation Using Sparse Contextualized Word Representations. EMNLP (2020).
40. Wang, M., Zhang, J., Wang, Y.: Enhancing the Context Representation in Similarity-based Word Sense Disambiguation. EMNLP (2021).
41. Song, Y., Ong, X.C., Ng, H.T., Lin, Q.: Improved Word Sense Disambiguation with Enhanced Sense Representations. EMNLP (2021).
42. Conia, S., Navigli, R.: Framing Word Sense Disambiguation as a Multi-Label Problem for Model-Agnostic Knowledge Integration. EACL (2021).
43. Wang, M., Wang, Y.: Word Sense Disambiguation: Towards Interactive Context Exploitation from Both Word and Sense Perspectives. ACL (2021).