Rare and Zero-shot Word Sense Disambiguation using Z-Reweighting

Ying Su[1], **Hongming Zhang**[1,2], **Yangqiu Song**[1], **Tong Zhang**[1]
[1]HKUST
[2]Tencent AI lab, Seattle
ysuay@connect.ust.hk, {hzhangal,yqsong}@cse.ust.hk,
tongzhang@ust.hk

Abstract

Word sense disambiguation (WSD) is a crucial problem in the natural language processing (NLP) community. Current methods achieve decent performance by utilizing supervised learning and large pre-trained language models. However, the imbalanced training dataset leads to poor performance on rare senses and zero-shot senses. There are more training instances and senses for words with top frequency ranks than those with low frequency ranks in the training dataset. We investigate the statistical relation between word frequency rank and word sense number distribution. Based on the relation, we propose a Z-reweighting method on the word level to adjust the training on the imbalanced dataset. The experiments show that the Z-reweighting strategy achieves performance gain on the standard English all words WSD benchmark. Moreover, the strategy can help models generalize better on rare and zero-shot senses.

1 Introduction

Word sense disambiguation (WSD) has been a long-standing problem in natural language processing community. The task can benefit many downstream applications (Navigli, 2009), including but not limited to machine translation (Vickrey et al., 2005; Pu et al., 2018) and information retrieval (Stokoe et al., 2003; Zhong and Ng, 2012).

The goal of the WSD task is to disambiguate word senses given contexts. For example, the word "*lift*" in the context "*Lift* a load" and "The detective carefully *lifted* some fingerprints from the table" has different meanings. The former one means "raise from a lower to a higher position" and the latter one means "remove from a surface". From semantic recognition of human being, the former sense is easier to disambiguate as it is the most common sense of the word while the latter one is a relatively rare one.

A skewed distribution exists in SemCor (Miller et al., 1993), a commonly used human-labeled dataset for the WSD task, where most common senses have many training examples while rare senses have much fewer examples. A large coverage of senses are not accompanied with training examples, which are called zero-shot senses. Many deep neural-networks-based methods are affected by this imbalanced training corpora (Luo et al., 2018; Huang et al., 2019b).

Previous approaches attempt to address this problem by designing a new dataset or task specifically for the rare senses and zero-shot senses (Holla et al., 2020; Blevins et al., 2021; Barba et al., 2021) or enriching the sense embeddings by incorporating external lexical knowledge (Kumar et al., 2019; Scarlini et al., 2020; Blevins and Zettlemoyer, 2020). Different from these methods, we address the unbalanced training issue from the perspective of adjusting the learning process.

An interesting human language phenomenon is that it follows a statistical distribution described by Zipf's law (Zipf, 1949), which also exists in many corpora including SemCor. From the linguistic perspective, an explanation for Zipf's law is that people tend to use more common words to minimize the communication effort (Zipf, 1949). Inspired by this, we consider a word with top rank in frequency should be assigned high training weight.

From the statistical perspective, two laws have been proposed to explain Zipf's law in word frequency, namely the meaning-frequency law (Zipf, 1945) and Zipf's law of abbreviation (Florence, 1950; Grzybek, 2006). The meaning-frequency law proposes that more frequent words have larger number of word senses, which we also denote as larger word #sense. Based on this, we calculate the word #sense distribution in SemCor and use a mathematical function to fit the relation between word rank and word #sense. Based on the relation, we design the Z-reweighting strategy on the word

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4713 - 4723
May 22-27, 2022 ©2022 Association for Computational Linguistics

level to help models generalize better to rare and zero-shot senses.

To the best of our knowledge, we are the first to leverage linguistic distribution to address the training bias on the WSD task. Our method improves the generalization ability of deep neural models on rare senses and zero-shot senses. Results on all English words WSD evaluation benchmarks show that our system achieves improvement on rare and zero-shot senses by 2.1% and 3.6% on F1 score. Furthermore, our strategy outperforms the system without any reweighting strategy and achieves a performance gain on the F1 score on all senses. We open source our code.[1]

2 Related works

2.1 Word Sense Disambiguation

Word sense disambiguation is to distinguish the sense of a specific word given a context sentence. Current methods can be broadly classified into two streams, supervised-learning-based and knowledge-based. Supervised-learning-based approaches view the WSD task as a classification problem. For example, Zhong and Ng (2010) learn classifiers independently for each word. Knowledge-based methods, such as (Banerjee et al., 2003; Basile et al., 2014), mainly exploit two kinds of knowledge: 1) the gloss, usually in the form of a sentence defining the word sense; 2) graph structure of lexical resources.

Recent researches integrate supervised learning and knowledge into a unified system and achieve better performance than systems relying on knowledge only. For utilizing gloss, GlossBert (Huang et al., 2019b) constructs context-gloss pairs and conducts sentence-pair classification training. Bi-encoder (Blevins and Zettlemoyer, 2020) proposes an end-to-end learning system to train the embedding space of context words and senses together. For utilizing structure properties, EWISE (Kumar et al., 2019) injects gloss and knowledge graph embedding into sense embeddings. EWISER (Bevilacqua and Navigli, 2020) further injects relational knowledge as additional supervision.

Different with the previous approaches, we focus on addressing training bias caused by the imbalanced distribution in the training dataset. In this paper, we analyze the formulation of the distribution and propose the Z-reweighting method to improve performance on rare and unseen senses.

2.2 Zipf's Law in Word Frequency

Power law distribution widely exists in human language, where the word frequency can be described by Zipf's law (Zipf, 1949). Previous works show that the linguistic law exists in many corpora, including SemCor (Miller et al., 1993), CHILDES (MacWhinney, 2000), and Wikipedia (Grefenstette, 2016). SemCor is also one of the largest training datasets for the WSD task, which also includes Ontonotes (Marcus et al., 2011) and OMSTI (Taghipour and Ng, 2015).

Manin (2008) argues from the semantic view and proposes that the word semantics are influenced by the expansion of word meanings and competition of synonyms results in the law. Zipf (1945) proposes that word frequency is related to its word #sense, in which more frequent words have larger word #sense. Recently, Casas et al. (2019) investigates the law from the perspective of both word #sense and word length. Similarly, our work takes consideration of word #sense distribution and utilizes it for balanced training on the WSD task.

2.3 Learning Imbalanced Dataset

There are many approaches to address the influence on learning brought by imbalanced training data under a supervised setting. Most of the algorithms belong to re-weighting (Huang et al., 2016, 2019a) or re-sampling (Buda et al., 2018; Cui et al., 2019). Re-weighting methods adjust the weights of different classes. Re-sampling methods balance the learning by over-sampling minority classes or under-sampling the frequent classes. Another line of works incorporates the idea of angular margin, aiming to enlarge the intra-class margin (Liu et al., 2016; Wang et al., 2018; Cao et al., 2019).

Our work follows the line of re-weighting. We take consideration of the word #sense distribution and propose Z-reweighting method for the WSD task, which is quite different with previous re-weighting methods.

3 Distribution Analysis in SemCor

In this section, we first show the overall word and sense distribution in SemCor[2] (Miller et al., 1993). Since we propose to utilize the word #sense distribution as the basis for the Z-reweighting strategy,

[1]Code is available: https://github.com/suytingwan/WSD-Z-reweighting.

[2]http://lcl.uniroma1.it/wsdeval/training-data

Type	Total num	MCS	LCS
Instance	226,036	166,361	59,675
Sense	33,316	22,320	10,996
Avg. Ins.	6.78	7.45	5.43
Word num	22,436	22,320	5,495
Word #sense	54,203	53,795	26,217
Avg. #sense	2.41	2.41	4.77

Table 1: Distribution of all words in SemCor on the word level and sense level, with **MCS** for the most common sense and **LCS** for the least common sense. *Avg. Ins.* means the average training instance number for senses.

we further look into the relationship between word rank, frequency, and word #sense.

3.1 Imbalanced Data Distribution in SemCor

As mentioned in (Kilgarriff, 2004), a Zipfian distribution exists in the word senses of human language. In this part, we investigate the details of the distribution in training data of SemCor on both the word level and the sense level.

Senses in WordNet are generally ordered from most to least frequently used [3]. The most common sense is ranked first, denoted as **MCS**. We denote other senses of a word as least common senses **LCS**. Following this definition, we calculate the distribution of training data in the SemCor corpus, and the resulting distribution is shown in Table 1. SemCor contains 226,036 training instances, where each instance is a sentence with a labeled sense of one word. Among all the instances, 73.5% are training instances for MCS, belonging to 22,320 words and the rest are for LCS. LCS has 5.43 training instances for each sense, much lower than MCS which has 7.45 instances on average.

We further investigate the word #sense distribution of training words labeled with MCS and LCS respectively. The word #sense defined in WordNet[4] is utilized to calculate the distribution. The average word #sense for training words labeled with LCS is 4.77, much greater than that of MCS. This shows that words labeled with LCS have a larger coverage of senses to distinguish. The words with LCS in training data SemCor has higher word #sense while with fewer training instances. Therefore, we can see that disambiguating LCS is much more challenging than MCS in the WSD task.

[3]https://wordnet.princeton.edu/documentation/wndb5wn

[4]https://wordnet.princeton.edu

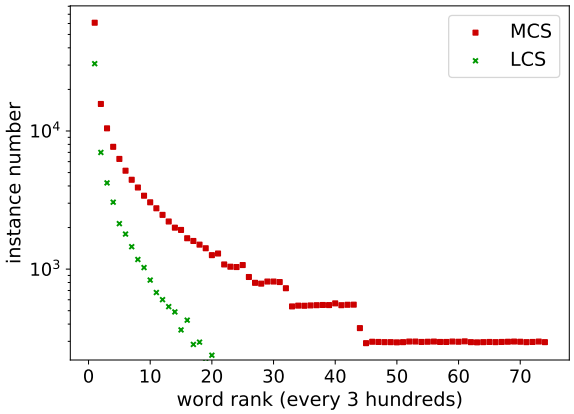

Figure 1: Word frequency distribution for MCS and LCS with sorted rank. One point represents the total instance number of three hundred words.

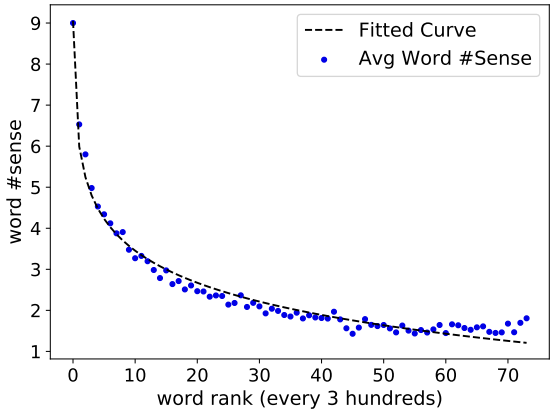

Figure 2: Average word #sense distribution with sorted rank. One point represents the average #sense of three hundred words.

3.2 Word Rank, Frequency, and Word #Sense

To investigate the details of the Zipfian distribution in SemCor, we calculate the number of training instances and word #sense for each word and sort them by frequency in descending order.

We apply a binning technique to reduce noise and get a better view of Zipf's law on word distribution. Specifically, every adjacent 300 words belong to a bin for clear analysis in this part. The distribution of instance number with sorted word rank by decreasing frequency is shown in Figure 1. As we can see, top ranked words have much more training cases than low ranked words both for training words labeled with MCS and LCS.

To get a deeper understanding of the statistical law in word frequency, we further analyze the relation between word #sense and sorted word rank. Similar to training instances, we calculate the average #sense of every 300 words in a bin and get the distribution of word #sense with the sorted word rank by decreasing frequency. As shown in Figure 2, the words with top rank have larger #sense than

words with low rank. This shows that words with the top frequency rank have more senses to disambiguate. Moreover, words with LCS are mostly with top ranks.

4 Algorithms

In this section, we first introduce the terminology for the WSD task. Then we illustrate our Z-reweighting strategy on adjusting the training loss for the imbalanced training dataset.

4.1 Terminology

The WSD task is to disambiguate the meanings of a set of words $w = \{w_1, w_2, ..., w_n\}$ given a context sentence S. Each context word $w_i, i \in [1, n]$ in a sentence S has several candidate senses $\{s_1, s_2, ..., s_m\}$. Each sense is described by a definition sentence, also called gloss in WordNet (Miller, 1998). The candidate senses have a corresponding gloss set $\{g_1, g_2, ..., g_m\}$.

4.2 Z-Reweighting Strategy

To alleviate the influence brought by the imbalanced training dataset, we propose the Z-reweighting strategy to balance the learning between MCS and LCS during training, resulting in a stronger capability of the model in disambiguating rare and zero-shot senses while maintaining comparable performance on MCS at the same time.

Training words in SemCor are denoted in form $W = \{W_1, W_2, ..., W_N\}$ with descending order of frequency. The #sense of a word represents the number of senses belonging to the word. $P = \{p_1, p_2, ..., p_N\}$ is the #sense array of the words.

To facilitate the analysis of word #sense, we use a bin parameter K to group the words. Average #sense array is calculated for every K words as:

$$p_o^{'} = \sum_{d=oK}^{o(K+1)} \frac{p_d}{K}, o \in [1, \frac{N}{K}], d \in [1, N], \quad (1)$$

$$P^{'} = \{p_1^{'}, ..., p_{\frac{N}{K}}^{'}\}, \quad (2)$$

As analyzed in (Casas et al., 2019), a power law exists between word frequency and #sense in the corpora CHILDES (MacWhinney, 2000). Similarly, we utilize a function $f(x) = a\ln(x + b) + c$ to fit the relation between word #sense $P^{'}$ and word rank $o = [1, 2, ..., \frac{N}{K}]$ in SemCor mathematically, where a, b, c are parameters. The fitting function is monotonic decreasing with word rank. An example of the fitting curve and original word #sense distribution with word frequency at $K = 300$ is shown in Figure 2.

With the same word ranks, a smoothed word #sense array can be calculated from the fitting curve as:

$$P^f = \{p_1^f, ..., p_{\frac{N}{K}}^f\}, \quad (3)$$

The discrete fitting word #sense array is normalized for further processing:

$$P^r = \frac{P^f}{\max(P^f)}, \quad (4)$$

Since the number of words is too large to assign each word a weight, the $\frac{N}{K}$ bins of words are further split into M groups. For word in k-th bin, $k \in [1, \frac{N}{K}]$, belonging to group $j \in [1, M]$, the regularized #sense satisfies:

$$p_{j+1}^t \leq p_k^r < p_j^t, \quad (5)$$

where $P^t = \{p_1^t, ..., p_M^t\}$ is the threshold array to split the groups. The words in group $j \in [1, M]$ are assigned weight:

$$\alpha_j = (p_j^t)^\eta, \quad (6)$$

where η is a power parameter.

Assume the predicted output probabilities from a model for candidate sense set as $z = [z_1, z_2, ..., z_m]$, the standard cross entropy loss given true word sense label y is:

$$loss(w_i, y) = -\log(\frac{\exp(z_y)}{\sum_{l=1}^m \exp(z_l)}). \quad (7)$$

In Z-reweighting strategy, the weight α_j is used to adjust the training on word level. The new weighted training loss is:

$$loss(w_i, j, y) = -\alpha_j \log(\frac{\exp(z_y)}{\sum_{l=1}^m \exp(z_l)}), \quad (8)$$

where $i \in [1, N]$ and $j \in [1, M]$, representing the word with rank i in group j has training weight α_j.

5 Experiments

In this section, we first introduce the training dataset and evaluation metrics. Then we show different baseline methods. Finally, details of the training process are presented.

5.1 Dataset

SemCor 3.0 is used as the training dataset. Five standard WSD datasets from Senseval and SemEval competitions are used as evaluation set. Among them, semeval 2007 (Pradhan et al., 2007) is used as development dataset for selecting the best model. Other four datasets including senseval-2 (Palmer et al., 2001), senseval-3 (Snyder and Palmer, 2004), semeval2012 (Navigli et al., 2013) and semeval2015 (Moro and Navigli, 2015) are used as test datasets. We select F1 as the evaluation metric. We also follow previous works (Raganato et al., 2017) to report the overall performance on all datasets. For further analysis, F1 scores on MCS, LCS, and zero-shot senses are also calculated.

5.2 Baselines

BEM framework (Blevins and Zettlemoyer, 2020) without any balanced strategy is a baseline system. In addition, different balanced training methods applied on BEM framework are used as three more baseline systems. The balanced methods are classified into two levels, namely, the sense-level (balanced reweighting and margin based method LDAW (Cao et al., 2019)), and the word-level (balanced resampling).

Biencoder Model (BEM). The model utilizes the gloss knowledge from WordNet. The two encoders are initialized with the same pre-trained language model. The encoders take a context sentence and glosses as input, generating representations for word w_i and corresponding gloss set $\{g_1, g_2, ..., g_m\}$ as E_i and $\{G_1, G_2, ..., G_m\}$ separately. Based on the representations, the similarity score between words and glosses are calculated as:

$$z_j = E_i \cdot G_j, j \in [1, m]$$

A standard cross-entropy loss is used in training as Equation 7.

Balanced Reweighting Method (B-reweighting). The B-reweighting strategy is applied on the sense level. For each word, the weights of senses is proportional to the inverse of training instances.

$$loss_{bal}(w_i, y) = -\beta_y \log \frac{\exp(z_y)}{\sum_{l=1}^{m} \exp(z_l)},$$

where $\beta_l = \frac{\sum_{l=1}^{m} n_l}{n_l}$ for $l \in [1, ..., m]$ and n_l is the number of training instances on sense l of w_i.

Balanced Resampling Method (B-resampling). The B-resampling method is applied on the word level. Firstly each word is sampled with the same probability. Then the training cases of the selected word are sampled randomly. Standard cross-entropy loss is used in this method.

LDAM. The margin-based method adjusts the training on the sense level. The goal of LDAM is to solve the class-imbalance problem by utilizing a label-distribution-aware margin loss. We apply the LDAW loss on the sense level as another baseline.

The smoothed relaxation of LDAM in the cross-entropy loss with enhanced margins is as follows:

$$loss_{margin}(w_i, y) = -\log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \sum_{l \neq y} e^{z_l}},$$

where $\Delta_l = \frac{C}{n_l^{1/4}}$, for $l \in \{1, \ldots, m\}$ and C is a constant. n_l is the training instances number of sense j for word w_i.

Standard LDAW is trained in two stages. Firstly the label-distribution-aware margin loss is applied to train the model for three epochs and B-reweighting loss is used for further training. In both stages, the learning rate is always 1e-5. For first stage training, C is set as 0.5.

5.3 Implementation details

Baseline Systems. Each system is trained for 20 epochs. AdamW (Kingma and Ba, 2015) is selected as the optimization algorithm. The learning rate is fixed at 1e-5 during training. The encoders in the biencoder framework both are initialized with bert-base (110M parameters) or bert-large (336M parameters) (Kenton and Toutanova, 2019). The experiments in which encoders initialized with bert-base are run on RTX 2080 and the experiments in which encoders initialized with Bert-large are run on RTX 3090. Average running hours is 30 hours for Bert-base and 40 hours for Bert-large.

Z-reweighting. To simplify the mathematical function fitting of word #sense distribution, we first split the words into bins by setting a fixed group number K. For the second grouping stage, to simplify the reweighting strategy on word level, we use thresholds to group the smoothed values calculated by fitting curve given word rank. In the experiments, we use weights of 1 decimal place as thresholds. The defined threshold array is as

model	Test Datasets					
	SE07	SE2	SE3	SE13	SE15	ALL
WordNet S1	55.2	66.8	66.2	63.0	67.8	65.2
MFS	54.5	65.6	66.0	63.8	67.1	65.5
EWISE (Kumar et al., 2019)	67.3	73.8	71.1	69.4	74.5	71.8
BERT-base (Kenton and Toutanova, 2019)	68.6	75.9	74.4	70.6	75.2	73.7
GlossBERT (Huang et al., 2019b)	72.5	77.7	75.2	76.1	80.4	77.0
BEM (Blevins and Zettlemoyer, 2020)	**72.8**$_{\pm 1.2}$	78.8$_{\pm 0.1}$	**77.2**$_{\pm 0.4}$	77.8$_{\pm 1.1}$	81.4$_{\pm 0.5}$	78.1$_{\pm 0.1}$
B-resampling	60.4$_{\pm 0.6}$	71.5$_{\pm 0.3}$	68.8$_{\pm 1.1}$	72.8$_{\pm 1.2}$	74.7$_{\pm 0.9}$	70.9$_{\pm 0.1}$
B-reweighting	71.3$_{\pm 0.3}$	78.4$_{\pm 0.7}$	75.5$_{\pm 0.5}$	75.6$_{\pm 1.3}$	80.4$_{\pm 1.3}$	77.1$_{\pm 0.3}$
LDAW	71.3$_{\pm 0.3}$	78.6$_{\pm 0.3}$	75.4$_{\pm 0.6}$	76.6$_{\pm 0.6}$	80.3$_{\pm 0.2}$	77.1$_{\pm 0.1}$
Z-reweighting	71.9$_{\pm 0.5}$	**79.6**$_{\pm 0.2}$	76.5$_{\pm 0.2}$	**78.9**$_{\pm 0.5}$	**82.5**$_{\pm 0.9}$	**78.6**$_{\pm 0.2}$

Table 2: F1(%) score on English all-words WSD task. For a fair comparison, the experiments on BEM and systems with balancing strategy are run three times. Each time an initial seed is randomly selected. The mean scores and standard deviation values of the three experiments are reported.

model	Senses		
	MCS	LCS	Zero-shot
BEM	**93.4**$_{\pm 0.3}$	51.7$_{\pm 0.3}$	67.2$_{\pm 0.9}$
B-Resampling	84.9$_{\pm 0.4}$	46.4$_{\pm 0.9}$	70.0$_{\pm 0.6}$
B-Reweighting	89.5$_{\pm 0.8}$	**55.4**$_{\pm 1.7}$	70.2$_{\pm 0.9}$
LDAW	92.1$_{\pm 1.3}$	52.8$_{\pm 0.7}$	68.6$_{\pm 0.5}$
Z-reweighting	92.9$_{\pm 0.1}$	53.8$_{\pm 0.4}$	**70.8**$_{\pm 1.1}$

Table 3: F1(%) score on MCS, LCS and zero-shot senses on the ALL test dataset. The mean scores and standard deviation values of three experiments are reported for each system.

$P^t = [1.0, 0.9, ..., 0.1]$, where the gap between thresholds is 0.1. For assigning weights, $\eta = 1, 2$ is used to adjust the value of weight. For example, words with regularized word #sense in $[0.3, 0.4)$ are in a group, assigning weight 0.16 when $\eta = 2$. The weight is further rounded with one decimal number as 0.2. If the rounding weight is less than 0.1, we use a weight of 0.1. For comparison with baselines, we set $K = 300, \eta = 2$ in the Z-reweighting strategy.

6 Results

In this section, we first analyze the overall performance on the test datasets using different training strategies. Then the details of improvement on MCS, LCS, and zero-shot senses for word groups are presented. Finally, we analyze the influences brought by hyper-parameters in Z-reweighting and influences brought by backbone models.

6.1 Overall Performance

The performance on test datasets by different systems are shown in Table 2. WordNet S1 uses the most common sense in WordNet and MFS uses the most frequent sense in the training dataset. Both the baselines achieve much lower performances than previous learning based systems, including BERT-base (Kenton and Toutanova, 2019), GlossBERT (Huang et al., 2019b) and BEM[5]. The sense embeddings in EWISE is fixed during training, which explains its much lower performance than GlossBERT and BEM.

The F1 score on MCS, LCS and zero-shot senses in ALL testset, with 4,603, 2,650, and 1,139 test instances respectively, are reported in Table 3. Comparing BEM with systems with balancing strategies, only Z-reweighting achieves performance gain on LCS and zero-shot senses while maintaining comparable performance on overall performance at the same time. Details show that though Z-reweighting slightly drops on MCS, performance on LCS and zero-shot senses increases 2.1% and 3.6% separately.

Besides the Z-reweighting method, B-reweighting and LDAW also show performance improvement on LCS and zero-shot senses, comparing with the BEM baseline. However, these balanced strategies deteriorate the system ability in

[5]We use original open source code for paper (Blevins and Zettlemoyer, 2020): https://github.com/facebookresearch/wsd-biencoders.

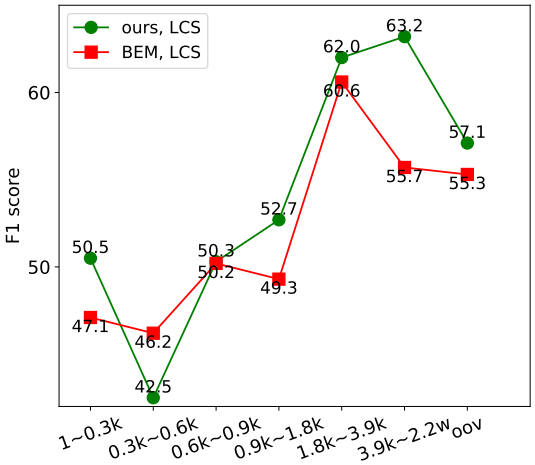

Figure 3: F1(%) score on LCS on ALL test dataset. *1-0.3k* means words with rank from 1 to 0.3k belongs to first group. *oov* means the group of words not appearing in SemCor.

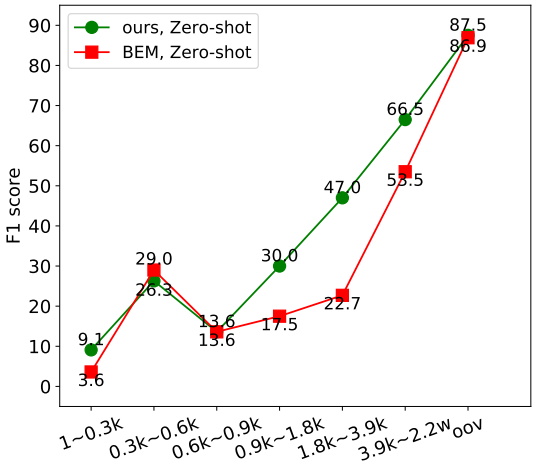

Figure 4: F1(%) score of zero-shot senses on ALL test dataset.

disambiguating MCS, resulting in the drop of F1 score on the ALL dataset.

Among all the balanced training strategies, B-resampling performs the worst. Equally sampling the words leads to insufficient training of top-ranked words, which results in poor performance. Our Z-reweighting strategy outperforms all the other balanced training strategies, indicating that our method is effective in improving the generalization ability of the model.

6.2 LCS, Zero-shot Senses in Word Groups

Among all the balanced strategies, only Z-reweighting outperforms baseline system BEM on the F1 score of ALL dataset. To look into how the Z-reweighting strategy works, we analyze the details of performance in the word groups. Noting that according to the Z-reweighting strategy, words in each group are assigned the same weight. The hyper-parameters are $K = 300, \eta = 2$ for our results. Under the setting, there are six groups of words from training dataset. These six groups of words are sorted by decreasing frequency order. The left words belong to a *oov* group in which words are not shown in the training dataset.

We calculate the F1 score of LCS and zero-shot senses of ALL test set and plot the results in Figure 3 and Figure 4 separately. In Figure 3, our system outperforms BEM on group one, in which the words are with highest frequency. The Z-reweighting strategy assigns the largest weight in this group and the F1 score improves 3.4%. For group 4 to group 7, our algorithm also shows consistent improvements. The performance gain drops in group 2. The reason behind this is that we manage to improve the performance of all words on the WSD task and use semeval2007 as development set for model selection.

For zero-shot senses shown in Figure 4, our system achieves improvement in five out of seven word groups, at most 24.3% for group five. The results show that the Z-reweighting strategy enables the model to generalize better to unseen senses. For words in group seven, the performance of zero-shot senses also improves, which shows that our method can further generalize better to senses of unseen words.

6.3 Impact of K and η in Z-reweighting

Each different bin number K results in a set of distinct weights for training words. In our experiments, we set $K = 50, 100, 200, 300, 400$ and $\eta = 1, 2$.

The performances of our system under different hyper-parameter settings are shown in Figure 5 and Table 4. In Figure 5, we can see that $\eta = 2$ achieves higher performance than $\eta = 1$ in most settings. This shows that the training weights with larger disparity on top and low ranked words result in higher performance on the overall score. The best performance achieves at $K = 300, \eta = 2$. It is interesting to see that with different hyper-parameters, the system has various overall scores. When $K = 400$, the overall score achieves lowest both for $\eta = 1$ and $\eta = 2$. It indicates that large K eliminates the weight distinctness between words during training, leading to drop on overall performance.

To explore the details of effects brought by

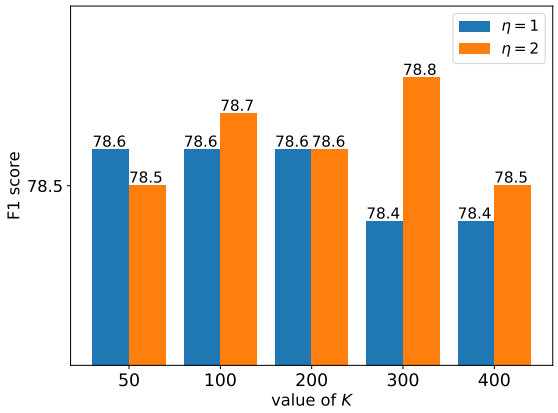

Figure 5: F1(%) score on ALL test dataset for different group parameter K and η.

hyper-parameters, we further show the F1 score of MCS, LCS, and zero-shot senses in Table 4. The accuracy of MCS varies from 92.8% to 93.3% under different settings. For most of the groups, MCS achieves higher performance with $\eta = 1$ than $\eta = 2$. When the gap between weights becomes larger with $\eta = 1$ changing to $\eta = 2$, the influences on LCS and zero-shot senses are greater than those on MCS. LCS achieves the best score 54.3%, 1.4% higher than the lowest score. Zero-shot senses achieve the best score of 72.3%, 1.4% higher than the lowest score. For all the combinations, we can see improvements on LCS and zero-shot senses compared to baseline BEM, demonstrating the effectiveness of our strategy.

6.4 Impact of Backbone Models

In this section, we show the influences brought by backbone models in BEM. The encoders of BEM are initialized by Bert-base and Bert-large models respectively for comparison. For Z-reweighting strategy, we use $K = 300, \eta = 2$. Training parameter settings are the same with the two backbone models. We experiment with different balanced training strategies. The results are presented in Figure 6. From the figure we can see that Bert-large achieves better performance on BEM and LDAM systems. For B-reweighting and Z-reweighting systems, the overall scores remain almost the same. However, for the B-resampling strategy, the performance drop 1%. Since the performances on Bert-large are nearly the same or even worse than Bert-base, we use Bert-base as the backbone model for training efficiency.

Parameter	Senses MCS	LCS	Zero-shot
K=50, η=1	**93.3**	53.0	70.9
K=50, η=2	93.0	53.5	71.6
K=100, η=1	**93.3**	53.1	71.5
K=100, η=2	93.1	53.7	**72.3**
K=200, η=1	93.2	53.4	71.6
K=200, η=2	92.9	54.0	71.9
K=300, η=1	92.9	53.4	71.5
K=300, η=2	93.0	**54.3**	72.2
K=400, η=1	93.2	52.9	71.3
K=400, η=2	92.8	53.7	72.2

Table 4: F1(%) score on MCS, LCS and zero-shot senses on ALL test dataset with different group number K and η.

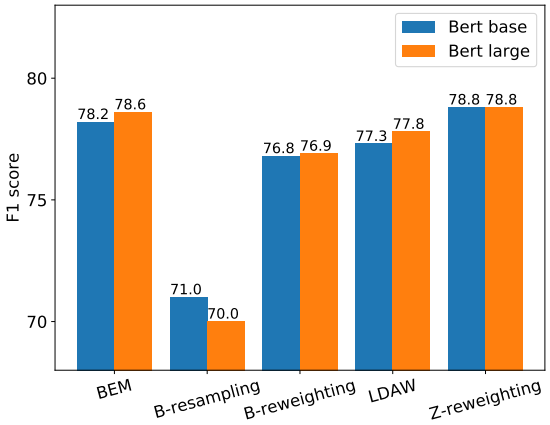

Figure 6: F1(%) score on ALL test dataset for balanced training strategies with different pretrained language models in BEM framework.

7 Conclusion

In this paper, we address the problem in learning imbalanced training dataset on the WSD task. Words with top frequency rank have more senses to disambiguate both for MCS and LCS. We assume these words should be assigned larger weights during training. Specifically, we use a mathematical function to fit the relation between word rank and word #sense, and utilize smoothed #sense to design the Z-reweighting strategy for all words English WSD task. The strategy leads to improvement on the performance of LCS and zero-shot senses on standard English WSD evaluation benchmarks. Furthermore, our method achieves performance gain on the F1 score for all senses. The results demonstrate the effectiveness of our methods.

Acknowledgements

This research was partially supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520) from RGC of Hong Kong, the MHKJFS (MHP/001/19) from ITC of Hong Kong with special thanks to HKMAAC and CUSBLT, and the Jiangsu Province Science and Technology Collaboration Fund (BZ2021065). We thank our colleague Tianqing Fang for providing insightful discussion and help in the research.

References

Satanjeev Banerjee, Ted Pedersen, et al. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Ijcai*, volume 3, pages 805–810. Citeseer.

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

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

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864.

Terra Blevins, Mandar Joshi, and Luke Zettlemoyer. 2021. Fews: Large-scale, low-shot word sense disambiguation with the dictionary. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 455–465.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *ACL*.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Bernardino Casas, Antoni Hernández-Fernández, Neus Catala, Ramon Ferrer-i Cancho, and Jaume Baixeries. 2019. Polysemy and brevity versus frequency in language. *Computer Speech & Language*, 58:19–50.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9268–9277.

P Sargant Florence. 1950. Human behaviour and the principle of least effort.

Gregory Grefenstette. 2016. Extracting weighted language lexicons from wikipedia. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1365–1368.

Peter Grzybek. 2006. *Contributions to the science of text and language: word length studies and related issues*, volume 31. Springer Science & Business Media.

Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Learning to learn to disambiguate: Meta-learning for few-shot word sense disambiguation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4517–4533.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2019a. Deep imbalanced learning for face recognition and attribute prediction. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2781–2794.

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

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

Adam Kilgarriff. 2004. How dominant is the commonest sense of a word? In *International conference on text, speech and dialogue*, pages 103–111. Springer.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. 2016. Large-margin softmax loss for convolutional neural networks. In *ICML*, volume 2, page 7.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482.

Brian MacWhinney. 2000. The childes project: Tools for analyzing talk: Volume i: Transcription format and programs, volume ii: The database.

Dmitrii Y Manin. 2008. Zipf's law and avoidance of excessive synonymy. *Cognitive Science*, 32(7):1075–1098.

Ralph Weischedel Eduard Hovy Mitchell Marcus, Martha Palmer, Robert Belvin Sameer Pradhan Lance Ramshaw, and Nianwen Xue. 2011. Ontonotes: A large training corpus for enhanced processing. *Joseph Olive, Caitlin Christianson, andJohn McCary, editors, Handbook of Natural LanguageProcessing and Machine Translation: DARPA GlobalAutonomous Language Exploitation*.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Andrea Moro and Roberto Navigli. 2015. Semeval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 288–297.

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

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231.

Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. Semeval-2007 task-17: English lexical sample, srl and all words. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 87–92.

Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110.

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

Benjamin Snyder and Martha Palmer. 2004. The english all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43.

Christopher Stokoe, Michael P Oakes, and John Tait. 2003. Word sense disambiguation in information retrieval revisited. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 159–166.

Kaveh Taghipour and Hwee Tou Ng. 2015. One million sense-tagged instances for word sense disambiguation and induction. In *Proceedings of the nineteenth conference on computational natural language learning*, pages 338–344.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 771–778.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 system demonstrations*, pages 78–83.

Zhi Zhong and Hwee Tou Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 273–282.

George Kingsley Zipf. 1945. The meaning-frequency relationship of words. *The Journal of general psychology*, 33(2):251–256.

George Kingsley Zipf. 1949. Human behaviour and the principle of least-effort. cambridge ma edn. *Reading: Addison-Wesley*.