Disentangled Representation for Long-tail Senses of Word Sense Disambiguation

Junwei Zhang
College of Intelligence and Computing, Tianjin University
Tianjin, China
junwei@tju.edu.cn

Ruifang He*
College of Intelligence and Computing, Tianjin University
Tianjin, China
rfhe@tju.edu.cn

Fengyu Guo*
College of Computer and Information Engineering, Tianjin Normal University
Tianjin, China
fyguo@tjnu.edu.cn

Jinsong Ma
College of Intelligence and Computing, Tianjin University
Tianjin, China
jsma@tju.edu.cn

Mengnan Xiao
College of Intelligence and Computing, Tianjin University
Tianjin, China
mnxiao@tju.edu.cn

ABSTRACT

The long-tailed distribution, also called the heavy-tailed distribution, is common in nature. Since both words and their senses in natural language have long-tailed phenomenon in usage frequency, the Word Sense Disambiguation (WSD) task faces serious data imbalance. The existing learning strategies or data augmentation methods are difficult to deal with the lack of training samples caused by the single application scenario of long-tail senses, and the word sense representations caused by unique word sense definitions. Considering that the features extracted from the Disentangled Representation (DR) independently describe the essential properties of things, and DR does not require deep feature extraction and fusion processes, it alleviates the dependence of the representation learning on the training samples. We propose a novel DR by constraining the covariance matrix of a multivariate Gaussian distribution, which can enhance the strength of independence among features compared to β-VAE. The WSD model implemented by the reinforced DR outperforms the baselines on the English all-words WSD evaluation framework, the constructed long-tail word sense datasets, and the latest cross-lingual datasets.

CCS CONCEPTS

• **Computing methodologies → Lexical semantics**; *Feature selection*; Natural language processing.

KEYWORDS

Word Sense Disambiguation; Disentangled Representation; Long-tail Senses; Data Imbalance; Long-tailed Distribution

ACM Reference Format:
Junwei Zhang, Ruifang He, Fengyu Guo, Jinsong Ma, and Mengnan Xiao. 2022. Disentangled Representation for Long-tail Senses of Word Sense Disambiguation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557288

1 INTRODUCTION

The long-tailed distribution (or heavy-tailed distribution) is a random variable distribution that is more common than the normal distribution [53]. Its intuitive characteristic is that head classes (categories) occupy a large number of samples, while tail classes (categories) have only a small number of samples. Data from the real world is difficult to be balanced and often presents an unbalanced phenomenon [70]..

For data-driven models in the field of Machine Learning (ML), the performance of state-of-the-art (SOTA) models degrades on the data following the long-tailed distribution [10, 70]. The core reason is that the training samples of the tail classes are insufficient, which makes it difficult to extract sufficient feature information from the limited training samples to achieve effective representations and correct recognition. In other words, it is difficult to learn the parameters of the model from limited training samples.

In the field of Natural Language Processing (NLP), it also faces the challenge of model failure caused by the long-tailed distribution [64]. In this regard, researchers have proposed data augmentation methods, such as Oversampling [3], Data Weighting [30] and Data Synthesis [38], etc., to deal with insufficient training samples. Although the existing methods can expand the sample size, it is difficult to provide more abundant unknown information. That is to say, the methods simply expand the number of samples based on known training data. In addition, some learning strategies (or models) for insufficient training samples are also proposed, such as Contrastive Learning [34], Generative Adversarial Networks [18] and Pretrained Models [51], etc.

However, for the WSD task, the above data augmentation methods and learning strategies against insufficient training samples have certain inapplicability. WSD is to assign the correct sense to

[*] Corresponding Author

CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557288

the target word in a given context, where the candidate list of senses is all senses listed in the dictionary [7, 45]. WSD is a lexical-level classification task, which needs to deal with the long-tail phenomenon of word and sense usage frequency at the same time. Its unique characteristics are as follows:

- Most of the long-tail senses are presented in the form of fixed collocations [46], that is, their application scenarios are narrow. So the data augmentation method that simply expands the number of samples cannot provide richer unknown information.
- In addition, the definitions of word senses (namely, glosses) are limited and fixed text descriptions in dictionaries, and the definition texts are very strict descriptions. So it is difficult for learning strategies based on knowledge transfer to accurately transfer comprehensive general description information from other places.

For the above reasons, it is difficult to mechanically expand the training sample size and definition of long-tail senses, and manual writing requires a high cost. Thus sufficient mining of existing data to obtain effective representations will be a correct option.

The concept of the **Disentangled Representation (DR)** was proposed as early as 2013 in a representation learning article published by Bengio et al. [5]. DR can be defined as the representation where single latent units are sensitive to changes in single generative factors, while being relatively invariant to changes in other factors [11]. That is to say, its goal is to learn features that are independent of each other, and each feature focuses on describing an independent primitive property. DR is widely used in Interpretable Machine Learning (IML) [39, 57] and Image Processing (IP) [12, 65]. In IML, independent features not only provide the possibility to clarify the meaning of each feature, but also further clarify what the algorithm has learned. In IP, independent features not only indicate the role of each feature, but also find a way to control image generation. However, the practical value of DR is much more than that, see Ref. [54] for more. The features obtained by DR are independent of each other, resulting in no need to learn complex correlations between features, which will greatly reduce the difficulty of learning the representation [11, 56].

Considering that the features extracted by DR are independent of each other, this paper leverages DR to obtain the features of single and fixed application scenarios of long-tail senses. And considering that DR does not require a deep feature extraction and fusion process, which greatly alleviates the dependence of the representation learning process on the number of training samples, this paper leverages DR to deal with glosses to obtain the correct word sense representations. Specifically, we first employ two pre-trained language models to obtain representations of target words and glosses, respectively, called **general representations**. Subsequently, on the basis of the general representations, we obtain their **disentangled representations** through the method proposed in this paper. Finally, we calculate the similarity over the general representation and the disentangled representation separately, and combine their output results into the final output, that is, the probability that the target word belongs to each word sense.

Our contributions can be summarized as follows:

- Rediscover the practical value of DR, that is, based on the superiority of DR in alleviating the complexity of feature extraction and fusion processes, we leverage DR to combat insufficient training samples;
- Propose a novel method to obtain DR by constraining the covariance matrix of a multivariate Gaussian distribution, which can enhance the strength of independence between features compared to β-VAE;
- Implement a WSD model using the reinforced DR, and validate the effectiveness on the English all-words WSD evaluation framework, the constructed long-tail sense datasets and the latest cross-lingual datasets.

2 RELATED WORK

2.1 Long-tailed Distribution for WSD

Due to the habits and preferences of language expressions, both words and senses in usage frequency show a severely imbalanced distribution. For the WSD task, dealing with long-tail words and long-tail senses caused by imbalanced distribution is the main challenge currently.

Researchers try to start directly from the data itself, and propose methods of Oversampling [72], Data Weighting [71], and Data Augmentation [36, 66] to deal with insufficient training samples caused by imbalanced distribution. The above methods alleviate the problem of insufficient training samples of target words to varying degrees, but cannot effectively deal with long-tail senses. Considering the characteristics of the description texts of the word senses (namely glosses), researchers propose to use multiple dictionaries to provide more description information of glosses, including Wikipedia [1, 19, 20], ConceptNet [13], and IndoWordNet [8]. Considering the phenomenon of fixed collocation of long-tail senses, researchers propose multi-lingual and cross-lingual schemes to provide collocation information in different languages [2, 4, 23].

In addition, learning strategies or models against insufficient training samples are also used in the WSD task, such as Meta-Learning [22, 28], Generative Adversarial Networks [29], and Pre-trained Models [24, 31].

2.2 Representation Methods for WSD

The importance of representations for the WSD task has been extensively explored by Iacobacci et al. [32]. The research on representation methods is mainly divided into two ideas: one is to include more knowledge for the representation, and the other is to constrain (or map) the representation to a specific space.

The methods of expanding knowledge try to include contextual information [58, 59], multi-sense information [55], multi-language information [23], domain information [63], or information contained in pre-trained models [25], etc. into the current representations. Obviously, such methods cannot alleviate the dependence of learned representations on training samples. Although such methods improve the overall performance of the models, the contribution cannot be clearly attributed to the efficient handling of long-tail words and senses.

The methods of imposing constraints attempt to map word or sense vectors into a compact or continuous space to improve recognition accuracy or reduce the dependence of learned representations on training samples [6, 37]. The method proposed by Kumar et al. [37] to constrain the representation of senses in a continuous space effectively copes with the shortage of training samples for long-tail senses. This work also inspires us to leverage the independence constraints between features, namely disentangled representation, to alleviate the dependence of learned representations on training samples.

Methods that integrate external knowledge or impose internal constraints can improve the accuracy of word sense representations to a certain extent, but have strong requirements for transferred knowledge. The approach adopted in this paper, namely disentangled representation, focuses on fully mining the knowledge of the training samples and does not need to rely on external assistance.

2.3 Disentangled Representation Methods

Since the concept of DR was proposed by Bengio et al. [5] in 2013, researchers have begun to try various methods to obtain DR, among which the most influential are implementations based on VAE (Variational Auto Encoder) [11, 14, 26] and GAN (Generative Adversarial Networks) [15, 33], and constrained methods based on Mutual Information Estimation [56].

β-VAE [11] is a variant of VAE, which enhances the ability of the VAE model to represent disentanglement. In VAE, we want to maximize the probability value of generating the true data and minimize the KL divergence of the true and estimated posterior distributions. In the corresponding Lagrangian function, the Lagrangian multiplier β is a hyperparameter. When β is 1, it is the standard VAE. A higher β value reduces the richness of the information represented by the variable space, but increases the disentanglement ability of the representation at the same time. So β can be used as a balance factor between representation ability and disentanglement ability. In addition, on the basis of β-VAE, its improved version β-TCVAE [14] is proposed.

InfoGAN [15] is an extension of GAN, which is capable of learning disentangled representations in a completely unsupervised manner and is able to maximize the mutual information between latent variables and observations for small-scale datasets. Specifically, InfoGAN successfully disentangles writing styles from digit shapes on the MNIST dataset, pose from the lighting of $3D$ rendered images, and background digits from the central digit on the SVHN dataset. It also discovers some visual concepts that include hair styles, presence/absence of eyeglasses, and emotions on the CelebA face dataset.

In addition, the method based on mutual information estimation obtains DR [56]: on the one hand, by maximizing the mutual information between the data and latent variables, and on the other hand, by minimizing the mutual information between the common latent variable and the exclusive latent variable, and finally achieve the purpose of obtaining DR.

On the basis of β-VAE, we propose a novel method to obtain disentangled representations inspired by quantum theory. Compared with β-VAE, this method can further improve the level of disentanglement between features.

3 METHODOLOGY

The principle and implementation details of obtaining DR are clarified in Sec. 3.1; the WSD model implemented by this representation is described in Sec. 3.2; the loss function and the chosen optimization method of the model are given in Sec. 3.3.

3.1 Disentangled Representation with Strong Constraints Among Features

From a mathematical point of view, DR is that the features of the representation are independent of each other. For a given random vector, the most common method to determine the strength of the correlation between its variables is mutual information. The larger the value of mutual information, the stronger the correlation; when the value is equal to zero, the variables are independent of each other [27]. Therefore, a natural idea is to employ the mutual information of this vector as a constraint term of the objective function to constrain the features of the representation to be independent of each other.

Assuming that we want to obtain DR

$$\vec{Z} = [z_1, z_2, ..., z_i, ...] \in \mathbb{R}^n \tag{1}$$

based on the known representation

$$\vec{X} = [x_1, x_2, ..., x_i, ...] \in \mathbb{R}^n, \tag{2}$$

the mutual information between the features of a random vector $\vec{Z}$, $\mathbf{I}(\vec{Z})$, can be defined by the Kullback-Leibler divergence as

$$\mathbf{I}(\vec{Z}) = \mathbf{KL}\left(P_1(\vec{Z}) || \prod_i P_2(z_i)\right) \tag{3}$$

where the subscripts 1 and 2 are used to distinguish different distribution functions later, $P(\vec{Z})$ refers to the joint distribution, and $\prod_i P(z_i)$ refers to the marginal distribution.

We can further assume that the probability density function of $\vec{Z}$ is a multivariate Gaussian distribution (also called multivariate normal distribution or joint normal distribution),

$$P_1(\vec{Z}) \sim \mathcal{N}(\vec{\mu}_1, \Sigma) \tag{4}$$

where $\vec{\mu}_1 \in \mathbb{R}^n$ is the mean vector,

$$\begin{aligned} \vec{\mu}_1 &= \mathbf{E}[\vec{Z}] \\ &= \left(\mathbf{E}[\vec{z_1}], \mathbf{E}[\vec{z_2}], ..., \mathbf{E}[\vec{z_i}], ...\right), \end{aligned} \tag{5}$$

and $\Sigma \in \mathbb{S}^{n \times n}_{++}$ is the covariance matrix,

$$\begin{aligned} \Sigma_{i,j} &= \mathbf{E}\left[(\vec{Z}_i - \vec{\mu}_i)(\vec{Z}_j - \vec{\mu}_j)\right] \\ &= Cov[\vec{Z}_i, \vec{Z}_j]. \end{aligned} \tag{6}$$

$\vec{\mu}_1$ can be obtained from the known representation $\vec{X}$ through a fully connected layer (or called linear layer) in the Neural Network,

$$\vec{\mu}_1 = linear(\vec{X}). \tag{7}$$

And Σ can also be obtained from $\vec{X}$,

$$\Sigma = \frac{1}{S} \sum_{s}^{S} \vec{X}_s'^T \vec{X}_s' \tag{8}$$

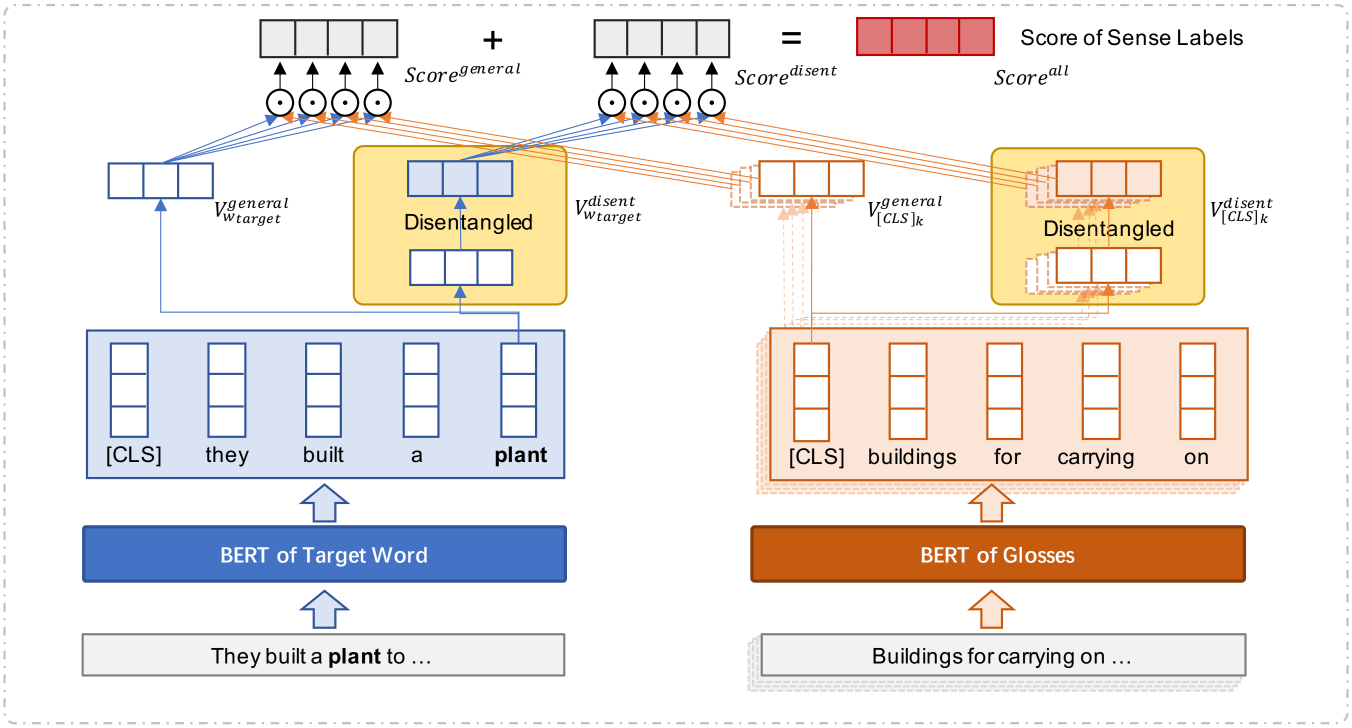

Figure 1: Schematic diagram of the model architecture. Our model consists of two BERTs to obtain embeddings for the target word and glosses, respectively. The obtained embeddings are copied in two copies: one with independence constraints imposed to obtain disentangled representations, and one without any processing. The scores under the disentangled representation are obtained by performing an inner product operation on the corresponding embeddings, the scores under the other representation are obtained in the same way, and they are added together as the final scores. The bold word *plant* refers to the target word, ⊙ refers to the inner product operation, and the embeddings in the yellow box represent the obtained disentangled representations.

where S represents the number of vectors that make up the Σ, and $\vec{X}'_s$ is defined as

$$\vec{X}'_s = SSN\left(linear_s(\vec{X})\right) \tag{9}$$
$$= \frac{linear_s(\vec{X})}{\sqrt{\Sigma\left(linear_s(\vec{X})\right)^2}}$$

where $SSN(\cdot)$ is a normalized function of the sum of squares. Because the covariance matrix is a positive semi-definite matrix, the purpose of the above complex operation is to obtain a qualified Σ. Through experimental analysis, it is found that the value of S cannot be too large or too small. When it is too large, the model is not easy to converge; when it is too small, the disentanglement effect of the representation is not good. The specific values are uniformly given at the model settings in the experimental section.

The reason for assuming a Gaussian distribution here is that the distribution has two parameters, which are easy to obtain, and the distribution is the most widespread distribution in nature. When the covariance matrix of the multivariate Gaussian distribution is a diagonal matrix Λ, it means that the variables are independent of each other, according to which the marginal distribution in Eq. (3) can also be described as a multivariate Gaussian distribution as

$$\prod_i P_2(z_i) \sim \mathcal{N}(\vec{\mu}_2, \Lambda) \tag{10}$$

where

$$\vec{\mu}_2 = linear(\vec{X}) \tag{11}$$

and

$$\Lambda = Diag\left(linear(\vec{X})\right). \tag{12}$$

The function of $Diag(\cdot)$ is to transform a vector into a diagonal matrix, that is, the elements of the vector are used as the diagonal elements of the new matrix, and the other off-diagonal elements of the matrix are set to 0.

In learning process of the model, the Kullback-Leibler divergence of the multivariate Gaussian distributions of $P_1(\vec{Z})$ and $\prod_i P_2(z_i)$,

$$\mathbf{KL}\left(P_1(\vec{Z})||\prod_i P_2(z_i)\right) \approx \mathbf{KL}\Big(\mathcal{N}(\vec{\mu}_1, \Sigma)||\mathcal{N}(\vec{\mu}_2, \Lambda)\Big) \tag{13}$$

is constrained. $\mathcal{N}(\vec{\mu}_1, \Sigma)$ with non-independent features tends to be $\mathcal{N}(\vec{\mu}_2, \Lambda)$ with independent features, and finally the features of $\mathcal{N}(\vec{\mu}_1, \Sigma)$ are independent of each other. At this point, we can obtain the disentangled representation $\vec{Z}$ by performing sampling on the above Gaussian distributions. However, since the sampling operation cannot be derived and the result of sampling can be derived, $\vec{Z}$ can be technically expressed as

$$\vec{Z} = \frac{1}{2}\Big((\vec{\mu}_1 + \vec{\mu}_2) + \vec{\varepsilon} * (diag(\Sigma) + diag(\Lambda))\Big) \tag{14}$$

where $\vec{\varepsilon}$ represents the perturbation (or noise) obtained by sampling from the Gaussian distribution $\mathcal{N}(0, \Lambda)$, $*$ represents bitwise multiplication, and the function of $diag(\cdot)$ is to obtain the diagonal elements of a matrix. The specific method of $diag(\cdot)$ is to extract the diagonal elements of the matrix as a new vector. The function of $Diag(\cdot)$ is to convert a vector to a matrix, while the function of $diag(\cdot)$ is to convert a matrix to a vector.

In addition, the constraint term of the disentangled representation of the objective function can be obtained by calculating the Kullback-Leibler divergence of the above multivariate Gaussian distribution,

$$\begin{aligned} &\mathbf{KL}\Big(\mathcal{N}(\vec{\mu}_1, \Sigma)||\mathcal{N}(\vec{\mu}_2, \Lambda)\Big) \qquad (15) \\ &= \mathbf{E}\Big(\log(\mathcal{N}(\vec{\mu}_1, \Sigma)) - \log(\mathcal{N}(\vec{\mu}_2, \Lambda))\Big) \\ &= \frac{1}{2}\Big\{\log\frac{|\Lambda|}{|\Sigma|} + tr(\Lambda^{-1}\Sigma) + (\vec{\mu}_2 - \vec{\mu}_1)\Lambda^{-1}(\vec{\mu}_2 - \vec{\mu}_1)^T - n\Big\}. \end{aligned}$$

In the model learning process, since obtaining the inverse of the matrix is time-consuming and Λ is a diagonal matrix, Λ^{-1} can be defined as

$$\Lambda^{-1} = \frac{\mathbf{1}}{\Lambda} \tag{16}$$

where$\Lambda_{i,i} \neq 0$ and $\mathbf{1}$ represents the identity matrix. Finally, the constraint term for obtaining the disentangled representation is defined as

$$\mathbf{KL} = \frac{1}{2}\Big\{\log\frac{|\Lambda|}{|\Sigma|} + tr(\frac{\Sigma}{\Lambda}) + \frac{(\vec{\mu}_2 - \vec{\mu}_1)(\vec{\mu}_2 - \vec{\mu}_1)^T}{\Lambda} - n\Big\}. \tag{17}$$

During model training, this KL divergence will be added to the loss function as a constraint term. The constraint term of DR of target words and glosses are all provided by Eq. (17), except that the corresponding parameters are selected from their respective mean vectors and covariance matrices.

3.2 Word Sense Disambiguation Using Reinforced DR

The overall architecture of our model is shown in Fig. 1. Our model imitates the bi-encoder architecture proposed by Blevins et al. [9]: one encoder (called **target word encoder**) is used to obtain the embedding of the target word, the other encoder (called **gloss encoder**) is used to obtain the embeddings of the glosses, and the model is optimized for both encoders through joint training. The reason why we also adopt the bi-encoder architecture is as follows:

- there are some differences between the text containing the target word and the text of the glosses, while the exclusive encoder will not ignore these differences;
- the disentangled representation proposed in this paper will amplify these differences, while the architecture will preserve them better.

Both the target word encoder and the gloss encoder are implemented with BERT [21] to obtain the corresponding embeddings, and the implementation details are as follows:

The target word encoder encodes the text containing the target word,

$$W = [w_1, w_2, ..., w_i, ...] \ni w_{target} \tag{18}$$

where w_{target} refers to the target word, into the corresponding embedding

$$V_W = [V_{w_1}, V_{w_2}, ..., V_{w_i}, ...] \ni V_{w_{target}} \tag{19}$$

where $V_{w_{target}}$ refers to the embedding of the target word. According to the processing rules of BERT, the symbols $[CLS]$ and $[SEP]$ are added as separators to the beginning and end of the text, respectively,

$$W = \Big[[CLS], w_1, w_2, ..., w_i, ..., [SEP]\Big], \tag{20}$$

and the corresponding embedding will be obtained,

$$V_W = \Big[V_{[CLS]}, V_{w_1}, V_{w_2}, ..., V_{w_i}, ..., V_{[SEP]}\Big]. \tag{21}$$

The reason for adopting BERT as the encoder is that BERT can learn the contextual information of the target word and contains a lot of public knowledge, which is very important for the WSD task. The embedding corresponding to the target word, i.e., $V_{w_{target}}$, is selected for the next processing steps.

The gloss encoder, also according to the processing rules of BERT, encodes the gloss texts corresponding to the target word,

$$G_k = \Big[[CLS]_k, w_{1,k}, w_{2,k}, ..., w_{i,k}, ..., [SEP]_k\Big], \tag{22}$$

and obtains the corresponding embeddings,

$$V_{G_k} = \Big[V_{[CLS]_k}, V_{w_{1,k}}, V_{w_{2,k}}, ..., V_{w_{i,k}}, ..., V_{[SEP]_k}\Big], \tag{23}$$

where k refers to the index of the gloss texts. The gloss texts are the description information of the senses of the target word, and do not contain the target word itself. The embeddings corresponding to $[CLS]_k$, i.e., $V_{[CLS]_k}$, are selected as text embeddings for the next processing steps. The reason for choosing $V_{[CLS]}$ as the text embedding of the gloss is because it contains all the information of the whole text, and is often used to represent the whole text in the industry.

Subsequently, **word sense recognition** of the target word is achieved based on the obtained embeddings. First, the embeddings obtained by the target word encoder and the gloss encoder are copied in two copies, namely $V^1_{w_{target}}$ and $V^1_{[CLS]_k}$ and $V^2_{w_{target}}$ and $V^2_{[CLS]_k}$, respectively.

$V^1_{w_{target}}$ and $V^1_{[CLS]_k}$ do not impose any processing and are directly regarded as **general representations**,

$$V^{general}_{w_{target}} \equiv V^1_{w_{target}} \tag{24}$$

and

$$V^{general}_{[CLS]_k} \equiv V^1_{[CLS]_k}. \tag{25}$$

The reason we call them general representations here is just to distinguish them from DRs. The similarity between the embedding of the target word and the embedding of each gloss is calculated to obtain the score of the corresponding sense under the general representation,

$$Score_k^{general} = V_{w_{target}}^{general} \odot V_{[CLS]_k}^{general} \tag{26}$$

where $\odot$ refers to the inner product operation.

$V_{w_{target}}^2$ is used to generate the mean vector and covariance matrix of the multivariate Gaussian distribution, that is, the mean vectors are denoted as:

$$\vec{\mu}_1 = linear(V_{w_{target}}^2) \tag{27}$$

and

$$\vec{\mu}_2 = linear(V_{w_{target}}^2) \tag{28}$$

and the covariance matrices are denoted as:

$$\Lambda = Diag\left(linear(V_{w_{target}}^2)\right) \tag{29}$$

and

$$\Sigma = \frac{1}{K}\sum_{k}^{K} V_k^T V_k, \tag{30}$$

where

$$V_k = SSN\left(linear_k(V_{w_{target}}^2)\right), \tag{31}$$

and then **disentangled representations** are obtained by Eq. (14), i.e.,

$$V_{w_{target}}^{disent} = \frac{1}{2}\left((\vec{\mu}_1 + \vec{\mu}_2) + \vec{\varepsilon} * (diag(\Sigma) + diag(\Lambda))\right). \tag{32}$$

Similarly, based on $V_{[CLS]_k}^2$, DRs of the glosses can be obtained, $V_{[CLS]_k}^{disent}$. The similarity between the embedding of the target word and the embedding of each gloss is calculated to obtain the score of the corresponding word sense under DR,

$$Score_k^{disent} = V_{w_{target}}^{disent} \odot V_{[CLS]_k}^{disent}. \tag{33}$$

At this point, we can obtain the final score of each sense of the target word by weighting the scores obtained under the general representation and the disentangled representation,

$$Score_k^{all} = \alpha Score_k^{general} + \beta Score_k^{disent} \tag{34}$$

where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ represent the corresponding weights, respectively, and k refers to the index of senses. The setting method of α and β can be determined by the experimental performance or by the distribution of word senses in the data. In this paper, we finally select equal values through experimental analysis.

3.3 Model Training

The objective function of our model consists of two parts, the cross-entropy loss of the final result and the constraints of obtaining the disentangled representation, where the constraints are further divided into the constraint of obtaining the disentangled representation of the target word and the constraint of obtaining the disentangled representation of the glosses.

The cross-entropy loss function is

$$\begin{aligned} &Loss(Score^{all}, index) \\ &= -\log\left(\frac{\exp(Score_{[index]}^{all})}{\sum_{i=1}\exp(Score_{[i]}^{all})}\right) \\ &= -Score_{[index]}^{all} + \log\sum_{i=1}\exp(Score_{[i]}^{all}) \end{aligned} \tag{35}$$

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

$$Score^{all} = [Score_1^{all}, Score_2^{all}, ..., Score_k^{all}, ...]. \tag{36}$$

Our model employs the Adam optimizer [35] to update the parameters, and the specific settings of the optimizer will be given in the experimental section.

4 EXPERIMENTS

4.1 Datasets

Our model is evaluated under the WSD evaluation framework proposed by Navigli et al. [47]. The training set is SemCor[1] [43]; the development set is selected as SemEval-2007 (SE07; [52]) by convention; the test sets include Senseval-2 (SE2; [49]), Senseval-3 (SE3; [61]), SemEval-2013 (SE13; [48]), SemEval-2015 (SE15; [44]), and the combination of all test sets (called **ALL**). In addition, the sets of *nouns*, *verbs*, *adjectives* and *adverbs* extracted from **ALL** are also used as the test set. The statistics of each dataset are shown in Tab. 1.

By convention, F1-score in percentage is used as an evaluation metric. All glosses come from WordNet 3.0[2] [42].

Table 1: Statistics of the datasets: the number of sentences (#Sents), tokens (#Tokens), sense annotations (#Annos), sense types covered (#Types) in each dataset. #Ambiguity refers to the ambiguity level, which implies the difficulty.

Dataset	#Sents	#Tokens	#Annos	#Types	#Ambiguity
SE2	242	5,766	2,282	1,335	5.4
SE3	352	5,541	1,850	1,167	6.8
SE07	135	3,201	455	375	8.5
SE13	306	8,391	1,644	827	4.9
SE15	138	2,604	1,022	659	5.5
SemCor	37,176	802,443	226,036	33,362	6.8

4.2 Baselines

To evaluate our model and determine its place in the WSD community, we divide the comparison models into two groups, namely the previous work and the baseline systems.

For the previous work, we select excellent models from the past three years, and these models are comparable to our models. GLU [25] and LMMS [40] in 2019 are selected; SREF [67], ARES [59] and SyntagRank [60] in 2020 are selected; COF [69], ESR [62] and

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

[2] http://wordnetweb.princeton.edu/perl/webwn

SACE [68] in 2021 are selected. Their results are taken from the published data of the original paper.

For the baseline systems, we choose BEM [9] with similar structure, GlossBERT [31] and BEM [9] using the same external resources, and EWISE [37] and EWISER [6] using different representations. Their results come from the original paper. In addition, based on the IMS [32] architecture, we obtain the results under Word2Vec [16], Context2Vec [41] and BERT [21] encoding methods, which are called IMS+word2vec, IMS+context2vec and IMS+bert, respectively.

4.3 Settings

The operating platform of the hardware is Ubuntu 18.04.3 and has a GPU whose version is NVIDIA TITAN Xp. The development platform is Python 3.8.3[3], and the learning framework is Pytorch 1.8.1[4]. The pretrained language model BERT is provided by Transformers 4.5.1[5]. Following the traditional comparison method, the versions BERT-base-uncased and BERT-large-uncased are used to build the encoders of the models, and the constructed models are called $\textbf{Our}_{\text{base}}$ and $\textbf{Our}_{\text{large}}$, respectively.

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. Parameters not listed will be given in the published code. The constant S in Eq. (8) is set to 3. The coefficients α and β of Eq. (34) take the same value and are both set to 0.5.

4.4 Results and Analysis

The experimental results are shown in Tab. 2 and divided into the comparison group of the previous work and the comparison group of the baseline systems.

In comparison with previous work, our model outperforms on most test sets and achieves excellent performance on the key metric **ALL**, indicating that our model has certain competitiveness. On test set SE13 and *Adjectives*, our model performs slightly lower than the comparison models. From the statistical information given by Tab. 1, SE13 has the lowest Ambiguity, indicating that the dataset is mostly high-frequency senses (that is, head senses), otherwise, the number of long-tail senses appears less. The purpose of the model constructed by DR proposed in this paper is to improve the recognition ability of long-tail senses. The results show that our model not only improves the recognition rate of long-tail senses, but also affects the recognition effect of high-frequency senses to a certain extent.

In comparison with baseline systems, our model also performs well on most test sets.

- Compared to BEM with a similar structure, our model outperforms this model, indicating that adding DR can improve the overall recognition ability of the model.
- Compared to models GlossBERT and BEM using similar external sources, our model also performs well, indicating that DR is effective.
- Compared to models EWISE and EWISER with different representations, and models with different encoding methods, it is proved that our model using both traditional and disentangled representations is the correct choice, and it outperforms models that only use a single representation.

4.5 Ablation Study

This section uses ablation experiments to analyze each component of the model in detail. To verify the contribution of DR, the method of removing the representations is adopted. To verify the value of the independence constraint to the parameters, the method of freezing parameters is adopted.

The experimental models are constructed as follows:

- Since the structure after removing DR is almost similar to BEM [9], we incorporate BEM into the comparative experiments.
- The original model $\textbf{Our}_{base}$ is listed here for comparison with the ablation and frozen models, and is called $\textbf{Our}_{original}$.
- The ablation model is the remaining structure after removing DR on the basis of the original model, and is called $\textbf{Our}_{ablation}$.
- The frozen model prevents updates of BERT's parameters when obtaining the representations, and is called $\textbf{Our}_{frozen}$.

The configuration and experimental results of BEM are taken from the original paper. The hyperparameters and other settings of the ablation model and frozen model are the same as those of the main experiment.

The experimental results are shown in Tab. 3. In terms of overall performance, $\textbf{Our}_{original}$ outperforms the contrasting models. The detailed comparison is:

- Comparing $\textbf{Our}_{ablation}$ with $\textbf{Our}_{original}$, the results show that the contribution of DR is more significant on SE2 and SE3 than on SE13 and SE15. The *#Ambiguity* values of SE2 and SE3 are lower (that is, there are more high-frequency senses than long-tail senses in the dataset), indicating that DR can further improve the accuracy of simple recognition tasks, but not for complex tasks.
- Comparing $\textbf{Our}_{frozen}$ with $\textbf{Our}_{original}$, the results show that the independence constraint is indeed beneficial to parameter update. Moreover, it also shows that DR purely relying on the original output of BERT also has some contribution to the final result.
- Comparing $\textbf{Our}_{ablation}$ and $\textbf{Our}_{frozen}$, the results show that the performance of $\textbf{Our}_{frozen}$ is better on SE2 and SE3, and the performance of $\textbf{Our}_{ablation}$ is better on SE13 and SE15, indicating that the performance of DR is positive under simple tasks (even without updating parameters); DR under complex tasks has a negative effect.

4.6 Experiments on Long-Tail Sense Datasets

In this section, we evaluate the performance of our model on long-tail sense datasets to further analyze the value of DR for long-tail senses.

Experimental Settings: On the basis of the training set SemCor and the test set **ALL**, we reconstruct the training sets and test sets

[3] https://www.python.org/
[4] https://pytorch.org/
[5] https://huggingface.co/

Table 2: F1-score (%) on the English all-words WSD task. The experimental results are divided into two parts: one is the previous work, and the other is the baseline systems. According to the characteristics of our model, the baseline models are grouped into models using BERT and Glosses, and models using different representations. SOTA performance is in bold.

Type	**Models**	**Dev set**	**Test sets**				**Concatenation of all test sets**				
		SE07	SE2	SE3	SE13	SE15	*Nouns*	*Verbs*	*Adjectives*	*Adverbs*	**ALL**
Previous Work											
	GLU (EMNLP 2019)	68.1	75.5	73.6	71.1	76.2	-	-	-	-	74.1
	LMMS (ACL 2019)	68.1	76.3	75.6	75.1	77.0	-	-	-	-	75.4
	SREF (EMNLP 2020)	72.1	78.6	76.6	78.0	80.5	80.6	66.5	82.6	84.4	77.8
	ARES (EMNLP 2020b)	71.0	78.0	77.1	77.3	**83.2**	80.6	68.3	80.5	83.5	77.9
	SyntagRank (ACL 2020)	59.3	71.6	72.0	72.2	75.8	-	-	-	-	71.2
	COF (EMNLP 2021)	69.2	76.0	74.2	78.2	80.9	80.6	61.4	80.5	81.8	76.3
	ESR (EMNLP 2021)	**75.4**	80.6	78.2	79.8	82.8	82.5	69.5	82.5	87.3	79.8
	SACE (ACL 2021)	74.7	80.9	79.1	**82.4**	84.6	83.2	71.1	**85.4**	87.9	80.9
Baseline Systems											
BERT + Glosses	GlossBERT (EMNLP 2019)	72.5	77.7	75.2	76.1	80.4	79.8	67.1	79.6	87.4	77.0
	BEM (ACL 2020)	74.5	79.4	77.4	79.7	81.7	81.4	68.5	83.0	87.9	79.0
different representations	IMS+word2vec	62.6	72.2	70.4	65.9	71.5	71.9	56.6	75.9	84.7	70.1
	IMS+context2vec	61.3	71.8	69.1	65.6	71.9	71.0	57.6	75.2	82.7	69.0
	IMS+bert	68.6	75.9	74.4	70.6	75.2	75.7	63.7	78.0	85.8	73.7
	EWISE (ACL 2019)	67.3	73.8	71.1	69.4	74.5	74.0	60.2	78.0	82.1	71.8
	EWISER (ACL 2020)	71.0	78.9	78.4	78.9	79.3	81.7	66.3	81.2	85.8	78.3
	Our$_{\text{base}}$	74.7	80.8	78.0	80.0	82.7	82.7	69.5	82.9	86.6	80.4
	Our$_{\text{large}}$	**75.4**	**81.1**	**79.2**	81.1	**83.2**	**84.9**	**72.1**	84.4	**88.5**	**81.4**

Table 3: F1-score (%) on the English all-words WSD task under ablation experiments. SOTA performance is in bold.

Models	**Dev set**	**Test sets**				**ALL**
	SE07	SE2	SE3	SE13	SE15	
BEM	74.5	79.4	77.4	79.7	81.7	79.0
Our$_{original}$	**74.7**	**80.8**	**78.0**	**80.0**	**82.7**	**80.4**
Our$_{ablation}$	73.5	76.1	75.2	77.7	81.4	77.0
Our$_{frozen}$	73.6	76.7	76.4	76.9	80.3	76.6

of long-tail senses. From the original dataset, we extract the samples whose sense samples are less than or equal to K as a dataset, where the values of K are 1, 3, 5 and 10 respectively. From this operation, we can get the training sets and test sets under 1, 3, 5 and 10 samples. We do not make any changes to the development set and still use the original SE07. Evaluation metrics and other settings are consistent with the main experimental setting method.

Experimental Models: Experimental models include the original version of our model (namely **Our**$_{original}$ in Sec. 4.5), the ablation version (namely **Our**$_{ablation}$ in Sec. 4.5), and BEM [9]. Since BEM is similar in structure to our model and is equivalent to our ablation model, it is also listed as a comparison model. The configurations and hyperparameter settings of the models are consistent with the main experimental setting method.

Experimental Results: The experimental results are shown in Fig. 2. Overall, our model outperforms the ablation model and BEM on the different long-tail sense datasets, indicating that DR is indeed beneficial for improving the recognition of long-tail senses. It can be further confirmed that the representation under independence constraints can indeed reduce the dependence on training samples.

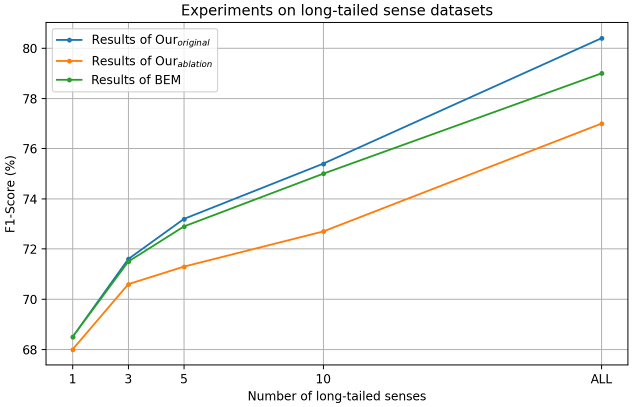

Figure 2: Evaluation experiments under the reconstructed long-tail sense datasets, where 1, 3, 5, and 10 refer to the number of long-tail senses in the dataset respectively (senses with a number of senses less than this value will be included); ALL refers to the entire dataset (i.e., the original training set SemCor and test set ALL).

The detailed comparison is:

- Comparing **Our**$_{original}$ with BEM, the results show that in the long-tail sense datasets of 1 and 3 sample, the performance of the two is comparable, and the advantage of

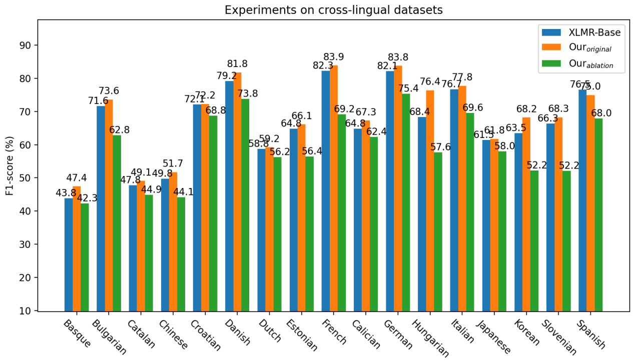

Figure 3: Evaluation experiments under cross-lingual datasets: The experimental models are XLMR-Base used in the original paper of the evaluation framework, and Our$_{original}$ and Our$_{ablation}$ in the ablation experimental section, where Our$_{original}$ is Our$_{base}$ in the main experimental section, and the structure and settings of Our$_{ablation}$ are similar to BEM.

Our$_{original}$ is prominent in the subsequent datasets, indicating that DR does not work in the case of extreme data scarcity. The results also indicate that a certain amount of data is needed to support the independence constraints among the features of the representation.

- Comparing **Our**$_{original}$ with **Our**$_{ablation}$, the results also support the above point of view, but **Our**$_{original}$ shows some advantages under the long-tail sense datasets of 1 and 3 sample, and **Our**$_{original}$ further enlarges the advantages under the long-tail sense datasets of 5 and 10 sample. The results indicate that our model can perform well in the case of extremely scarce training data and relatively sufficient training data.
- **Our**$_{ablation}$ and BEM are essentially the same models, but they have some gaps, indicating that our model still has the potential to improve by adding some tricks.

4.7 Experiments on Cross-Lingual Datasets

To further evaluate the robustness of the model, we conduct experiments on the latest cross-lingual evaluation framework[6] proposed by Pasini et al. [50] in 2021. This evaluation framework has a larger amount of data than the evaluation framework proposed by Navigli et al. [47], while covering other languages besides English. In this section, we only conduct experiments on languages other than English.

Furthermore, to present the contribution of DR to the model, we also adopt the settings of the models in the ablation experiments, namely the original model **Our**$_{original}$ and the ablation model **Our**$_{ablation}$. For the control model, we adopt the model XLMR-Base [17] used in the original paper as baselines. Note that since the cross-lingual datasets are constructed based on BabelNet[7], the glosses in the experiments are from BabelNet. Moreover, since most small languages in BabelNet use definitions in English, we directly use glosses in English to provide a candidate list of senses. Other unmentioned information about the model setting is consistent with the main experiment, and the information about the datasets is consistent with the setting method of this evaluation framework.

The experimental results are shown in Fig. 3.

- Comparing **Our**$_{original}$ with XLMR-Base, **Our**$_{original}$ is better than XLMR-Base on multiple datasets, which shows that our model has a certain robustness, and also shows that DR method has a wide range of applicability.
- Comparing **Our**$_{original}$ with **Our**$_{ablation}$, **Our**$_{original}$ is superior to **Our**$_{ablation}$, which shows that DR does play a role, and indirectly points out that DR is helpful for the representation and identification in the cross-lingual datasets. The excellent performance on these datasets also shows that employing English sense definitions for other languages easily allows the system to treat definitions in English as long-tail senses.

Analysis of the poor results: On the Spanish, Japanese, Estonian, Dutch, Croatian and Catalsn datasets, our model performs poorly, and the core reason is that their volume is relatively small, and most of them are labeled with high-frequency word senses. As the ablation study concluded, DR still requires a certain amount of data to back them up. Moreover, when the high-frequency word senses account for the vast majority, DR does not have room to play.

5 CONCLUSIONS

The long-tailed distribution of data leads to serious data imbalance and brings great challenges to data-driven models. Considering that DR does not require a complex feature extraction and integration process like the traditional neural network-based representation learning methods, this paper proposes to leverage DR to alleviate the dependence of the WSD model on the sample size during the long-tail sense training process. Our contribution is to propose a novel method to obtain DR through an independence constraint mechanism among features under the assumption of multivariate Gaussian distribution, which can enhance the strength of independence between features compared to β-VAE. The effectiveness of the model is validated on the English all-words WSD evaluation framework, the constructed long-tail word sense datasets and the latest cross-lingual datasets.

The significance of this paper is to rediscover the value of DR from the perspective of alleviating the dependence of data-driven models on training data. And this paper proposes a novel method to obtain DR, but the effectiveness of this method in other fields needs further verification. In future work, we will explore ways to extract fewer and more representative features under the premise of feature independence.

ACKNOWLEDGEMENTS

Our work is supported by the National Natural Science Foundation of China (61976154), the National Key R&D Program of China (2019YFC1521200) and the National Natural Science Foundation of China (No. 62106176).

[6] https://sapienzanlp.github.io/xl-wsd/
[7] https://babelnet.org/

REFERENCES

[1] Marwah Alian, Arafat A. Awajan, and Akram Al-Kouz. 2016. Arabic Word Sense Disambiguation Using Wikipedia. *International Journal of Computing* 12, 4 (2016), 61–66.

[2] Carmen Banea and Rada Mihalcea. 2011. Word Sense Disambiguation with Multilingual Features. In *IWCS*. 847–851.

[3] Saptarshi Bej, Narek Davtyan, Markus Wolfien, Mariam Nassar, and Olaf Wolkenhauer. 2021. LoRAS: An oversampling approach for imbalanced datasets. *Mach. Learn.* 110, 2 (2021), 279–301.

[4] Gábor Bella, Alessio Zamboni, and Fausto Giunchiglia. 2016. Domain-Based Sense Disambiguation in Multilingual Structured Data. In *ECAI*. 53–61.

[5] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.

[6] Michele Bevilacqua and R. Navigli. 2020. Breaking Through the 80% Glass Ceiling: Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information. In *ACL*. 2854–2864.

[7] Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent Trends in Word Sense Disambiguation: A Survey. In *IJCAI*. 4330–4338.

[8] Sudha Bhingardive and Pushpak Bhattacharyya. 2017. Word Sense Disambiguation Using IndoWordNet. In *The WordNet in Indian Languages*. 243–260.

[9] Terra Blevins and Luke Zettlemoyer. 2020. Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders. In *ACL*. 1006–1017.

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

[11] Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nicholas Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in β-VAE. *ArXiv* abs/1804.03599 (2018).

[12] Agisilaos Chartsias, Thomas Joyce, Giorgos Papanastasiou, Michelle Claire Williams, David E. Newby, Rohan Dharmakumar, and Sotirios A. Tsaftaris. 2019. Factorised Representation Learning in Cardiac Image Analysis. *Medical image analysis* 58 (2019), 101535.

[13] Junpeng Chen and Juan Liu. 2011. Combining ConceptNet and WordNet for Word Sense Disambiguation. In *IJCNLP*. 686–694.

[14] Tian Qi Chen, Xuechen Li, Roger B. Grosse, and David Kristjanson Duvenaud. 2018. Isolating Sources of Disentanglement in Variational Autoencoders. In *NeurIPS*. 2615–2625.

[15] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and P. Abbeel. 2016. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NeurIPS*. 2172–2180.

[16] Kenneth Ward Church. 2017. Word2Vec. *Natural Language Engineering* 23, 1 (2017), 155–162.

[17] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *ACL*. 8440–8451.

[18] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil Anthony Bharath. 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* 35, 1 (2018), 53–65.

[19] Bharath Dandala, Rada Mihalcea, and Razvan C. Bunescu. 2013. Multilingual Word Sense Disambiguation Using Wikipedia. In *IJCNLP*. 498–506.

[20] Bharath Dandala, Rada Mihalcea, and Razvan C. Bunescu. 2013. Word Sense Disambiguation Using Wikipedia. In *The People's Web Meets NLP*. 241–262.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*. 4171–4186.

[22] Yingjun Du, Nithin Holla, Xiantong Zhen, Cees G. M. Snoek, and Ekaterina Shutova. 2021. Meta-Learning with Variational Semantic Memory for Word Sense Disambiguation. In *ACL/IJCNLP*. 5254–5268.

[23] Erwin Fernandez-Ordonez, Rada Mihalcea, and Samer Hassan. 2012. Unsupervised Word Sense Disambiguation with Multilingual Representations. In *LREC*. 847–851.

[24] Ping Guo, Yue Hu, and Yunpeng Li. 2020. MG-BERT: A Multi-glosses BERT Model for Word Sense Disambiguation. In *KSEM*, Vol. 12275. 263–275.

[25] Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations. In *EMNLP*. 5296–5305.

[26] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*. 314–323.

[27] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*. 897–907.

[28] Nithin Holla, Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2020. Learning to Learn to Disambiguate: Meta-Learning for Few-Shot Word Sense Disambiguation. In *EMNLP*, Vol. 2020. 4517–4533.

[29] Zijian Hu, Fuli Luo, Yutong Tan, Wenxin Zeng, and Zhifang Sui. 2019. WSD-GAN: Word Sense Disambiguation Using Generative Adversarial Networks. In *AAAI*. 9943–9944.

[30] Zhiting Hu, Bowen Tan, Ruslan Salakhutdinov, Tom Michael Mitchell, and Eric P. Xing. 2019. Learning Data Manipulation for Augmentation and Weighting. In *NeurIPS*. 15738–15749.

[31] Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. In *EMNLP/IJCNLP*. 3507–3512.

[32] Ignacio Iacobacci, Mohammad Taher Pilehvar, and R. Navigli. 2016. Embeddings for Word Sense Disambiguation: An Evaluation Study. In *ACL*. 897–907.

[33] In S. Jeon, Wonkwang Lee, Myeongjang Pyeon, and Gunhee Kim. 2021. IB-GAN: Disentangled Representation Learning with Information Bottleneck Generative Adversarial Networks. In *AAAI*. 7926–7934.

[34] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems* 33 (2020), 18661–18673.

[35] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*. 99–110.

[36] Harsh Kohli. 2021. Transfer Learning and Augmentation for Word Sense Disambiguation. In *ECIR*, Vol. 12657. 303–311.

[37] Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Pratim Talukdar. 2019. Zero-shot Word Sense Disambiguation using Sense Definition Embeddings. In *ACL*. 5670–5681.

[38] Bohan Li, Yutai Hou, and Wanxiang Che. 2022. Data Augmentation Approaches in Natural Language Processing: A Survey. *AI Open* 3 (2022), 71–90.

[39] Decheng Liu, Xinbo Gao, Chunlei Peng, Nannan Wang, and Jie Li. 2021. Heterogeneous Face Interpretable Disentangled Representation for Joint Face Recognition and Synthesis. *IEEE transactions on neural networks and learning systems* 21 (2021), 1–15.

[40] Daniel Loureiro and Alípio Mário Jorge. 2019. Language Modelling Makes Sense: Propagating Representations through WordNet for Full-Coverage Word Sense Disambiguation. In *ACL*. 5682–5691.

[41] Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *CoNLL*. 51–61.

[42] George A. Miller. 1992. WordNet: A Lexical Database for English. *Commun. ACM* 38 (1992), 39–41.

[43] George A. Miller, Claudia Leacock, Randee Tengi, and Ross Bunker. 1993. A Semantic Concordance. In *Proceedings of the Workshop on Human Language Technology*. 303—-308.

[44] Andrea Moro and R. Navigli. 2015. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation*. 288–297.

[45] R. Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.* 41 (2009), 10:1–10:69.

[46] Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.* 41, 2 (2009), 10:1–10:69.

[47] R. Navigli, José Camacho-Collados, and Alessandro Raganato. 2017. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In *EACL*. 99–110.

[48] R. Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. In *Second Joint Conference on Lexical and Computational Semantics*. 222–231.

[49] Martha Palmer, Christiane D. Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. 2001. English Tasks: All-Words and Verb Lexical Sample. In *Proceedings of Second International Workshop on Evaluating Word Sense Disambiguation Systems*. 21–24.

[50] Tommaso Pasini, Alessandro Raganato, and R. Navigli. 2021. XL-WSD: An Extra-Large and Cross-Lingual Evaluation Framework for Word Sense Disambiguation. In *AAAI*. 13648–13656.

[51] Jiajia Peng and Kaixu Han. 2021. Survey of Pre-trained Models for Natural Language Processing. *ICEIB* (2021), 277–280.

[52] 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*. 87–92.

[53] William J. Reed. 2001. The Pareto, Zipf and other power laws. *Economics Letters* 74, 1 (2001), 15–19.

[54] Karl Ridgeway. 2016. A Survey of Inductive Biases for Factorial Representation-Learning. *ArXiv* abs/1612.05299 (2016).

[55] Terry Ruas, William I. Grosky, and Akiko Aizawa. 2019. Multi-sense embeddings through a word sense disambiguation process. *Expert Syst. Appl.* 136 (2019), 288–303.

[56] Eduardo Hugo Sanchez, Mathieu Serrurier, and Mathias Ortner. 2020. Learning Disentangled Representations via Mutual Information Estimation. In *ECCV*,

Vol. 12367. 205–221.

[57] Mhd Hasan Sarhan, Abouzar Eslami, Nassir Navab, and Shadi Albarqouni. 2019. Learning Interpretable Disentangled Representations using Adversarial VAEs. In *MICCAI*, Vol. 11795. 37–44.

[58] Bianca Scarlini, Tommaso Pasini, and R. Navigli. 2020. SensEmBERT: Context-Enhanced Sense Embeddings for Multilingual Word Sense Disambiguation. In *AAAI*. 8758–8765.

[59] Bianca Scarlini, Tommaso Pasini, and R. Navigli. 2020. With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All-Round Word Sense Disambiguation. In *EMNLP*. 3528–3539.

[60] Federico Scozzafava, Marco Maru, Fabrizio Brignone, Giovanni Torrisi, and R. Navigli. 2020. Personalized PageRank with Syntagmatic Information for Multilingual Word Sense Disambiguation. In *ACL*. 37–46.

[61] 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.* 41–43.

[62] Yang Song, Xin Cai Ong, Hwee Tou Ng, and Qian Lin. 2021. Improved Word Sense Disambiguation with Enhanced Sense Representations. In *EMNLP*. 4311–4320.

[63] Kaveh Taghipour and Hwee Tou Ng. 2015. Semi-Supervised Word Sense Disambiguation Using Word Embeddings in General and Specific Domains. In *NAACL*. 314–323.

[64] Sahar Tahvili, Leo Hatvani, Enislay Ramentol, Rita Pimentel, Wasif Afzal, and Francisco Herrera. 2020. A novel methodology to classify test cases using natural language processing and imbalanced learning. *Engineering applications of artificial intelligence* 95 (2020), 103878.

[65] Shichang Tang, Xueying Zhou, Xuming He, and Yi Ma. 2021. Disentangled Representation Learning for Controllable Image Synthesis: An Information-Theoretic Perspective. *ICPR* (2021), 10042–10049.

[66] Dilara Torunoglu-Selamet, Arda Inceoğlu, and G. Eryigit. 2020. Preliminary Investigation on Using Semi-Supervised Contextual Word Sense Disambiguation for Data Augmentation. *UBMK* (2020), 337–342.

[67] Ming Wang and Yinglin Wang. 2020. A Synset Relation-enhanced Framework with a Try-again Mechanism for Word Sense Disambiguation. In *EMNLP*. 6229–6240.

[68] Ming Wang and Yinglin Wang. 2021. Word Sense Disambiguation: Towards Interactive Context Exploitation from Both Word and Sense Perspectives. In *ACL/IJCNLP*. 5218–5229.

[69] Ming Wang, Jianzhang Zhang, and Yinglin Wang. 2021. Enhancing the Context Representation in Similarity-based Word Sense Disambiguation. In *EMNLP*. 8965–8973.

[70] Yuxiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to Model the Tail. In *NIPS*. 7029–7039.

[71] Guorui Zhao and Wan li Zuo. 2014. Semi-Supervised Word Sense Disambiguation via Context Weighting. *Advanced Materials Research* 1049-1050 (2014), 1327 – 1338.

[72] Jingbo Zhu and Eduard H. Hovy. 2007. Active Learning for Word Sense Disambiguation with Methods for Addressing the Class Imbalance Problem. In *EMNLP*. 783–790.