UoR-NCL at SemEval-2023 Task 1: Learning Word-Sense and Image Embeddings for Word Sense Disambiguation

Thanet Markchom[1] and **Huizhi Liang**[2] and **Joyce Gitau**[2] and
Zehao Liu[2] and **Varun Ojha**[2] and **Lee Taylor**[2] and
Jake Bonnici[2] and **Abdullah Alshadadi**[2]
[1]Department of Computer Science, University of Reading, Reading, UK
[2]School of Computing, Newcastle University, Newcastle upon Tyne, UK
t.markchom@pgr.reading.ac.uk, {huizhi.liang, j.m.gitau2, z.liu83,
varun.ojha, l.taylor22, j.bonnici3, a.t.h.alshadadi2}@newcastle.ac.uk

Abstract

In SemEval-2023 Task 1, a task of applying Word Sense Disambiguation in an image retrieval system was introduced. To resolve this task, this work proposes three approaches: (1) an unsupervised approach considering similarities between word senses and image captions, (2) a supervised approach using a Siamese neural network, and (3) a self-supervised approach using a Bayesian personalized ranking framework. According to the results, both supervised and self-supervised approaches outperformed the unsupervised approach. They can effectively identify correct images of ambiguous words in the dataset provided in this task.

1 Introduction

In several natural languages, there are ambiguous words that denote different meanings depending on their contexts. For example, the word "bank" in English may have different senses including "financial institution" and "riverside". To identify the exact sense of an ambiguous word for a particular context, a Natural Language Processing (NLP) task called Word Sense Disambiguation (WSD) was introduced (Navigli, 2009; Bevilacqua et al., 2021). Despite the fact that WSD has been a long-standing task, it is still challenging to apply WSD methods to downstream NLP applications (Lopez de Lacalle and Agirre, 2015; Raganato et al., 2017). Therefore, the organizers of SemEval-2023 Task 1 proposed a task of using WSD in an information retrieval system (Raganato et al., 2023). Specifically, given a word and some limited textual context, the task is to select among a set of candidate images the one corresponding to the given word's intended meaning.

According to the goal, two challenges are raised. The first challenge is how to disambiguate a given word based on a given context. Normally, a human brain differentiates multiple senses of an ambiguous word based on prior knowledge and experience. Similarly, in the case of machines, external knowledge is required to map an ambiguous word to its actual meaning. Many knowledge-based WSD methods have been relying on external lexical resources such as WordNet (Miller, 1995). These resources contain word senses which can be extracted for disambiguation (Huang et al., 2019). The second challenge is how to match a word with an image. In many image retrieval systems, word representations/embeddings and image representations/embeddings are learned in a way that allows word-image association (Bengio et al., 2013). This work adopts this approach to learn word-sense and image embeddings in order to associate word senses with their corresponding images. We propose three approaches utilizing WordNet to disambiguate words and learn word-sense embeddings and image embeddings for image retrieval. These approaches are (1) an unsupervised approach using word senses and image captions, (2) a supervised approach using a Siamese neural network to learn word-sense and image embeddings, and (3) a self-supervised contrastive learning approach using a Bayesian personalized ranking framework.

2 Related Work

There are a number of methods that have been researched in WSD. The first method is Knowledge based method that relies on lexical resources and dictionaries (Yarowsky, 1993). The second approach is supervised machine learning methods that use manually sense-annotated corpora, e.g, Train-O-Matic (Pasini and Navigli, 2017), which is a language-independent that generates sense-annotated training instances for senses of words. The third approach involves semi-supervised methods that use both labeled and unlabeled data and therefore the amount of annotated data need not to be large. Taghipour and Ng (2015) proposed a method that uses word embeddings such that words are in a continuous space. These word em-

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 16–22
July 13-14, 2023 ©2023 Association for Computational Linguistics

beddings come from unlabeled data and therefore this method becomes semi-supervised. The fourth approach is unsupervised machine learning methods that use clusters to identify the different occurrences of word senses. For instance, Yarowsky (1995) presented an unsupervised learning algorithm trained on unannotated text. Supervised methods have outperformed all the other methods. However, due to the lack of a flexible knowledge base, they might be replaced by neural networks that use sequence learning for word disambiguation. In this work, we use WordNet, a lexical database, to disambiguate the context phrases to get the different senses of words that describe the gold image.

3 System Description

3.1 Sense-Caption Similarity Approach

Our aim of using WSD is to associate a word w in a context phrase to its most appropriate sense using WordNet[1] based on their synonymy, hyponymy, hypernymy, and antonymy. The word senses for the individual words in the complex phrase containing two tokens are obtained using a series of steps as illustrated in figure1. We will consider an ambiguous context phrase “leucaena genus” the phrase contains two words w_1 and w_2. We lemmatize these words to get the two lemmas and assign a POS tag per lemma, e.g., [(‘leucaena’, ‘NN’), (‘genus’, ‘NN’)]. After that, we query WordNet to find the word senses of the two tokens. All word senses obtained from WordNet are then concatenated by “or” and used as final word senses of w.

To match an image with a word sense, we compare the caption of each image with the word sense of a given word. The model used for generating image captions is called the Neural Image Caption (NIC) model[2] (Vinyals et al., 2015). The NIC model is a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The CNN component of the model provides the capability to extract and encode important visual features from the image, while the RNN component of the model generates a natural language caption based on the encoded image features. Assuming that the NIC model should contain prior knowledge of image understanding, in this work, we adopt this model to generate image captions and match them with word senses. Image captions are generated by first loading the training dataset and constructing a vocabulary. This vocabulary is then utilized to convert words into numerical representations that the NIC model can understand. The best model weights are loaded into a greedy inference model, a specific type of NIC model. The model processes an image into a feature vector and produces the final caption for the image through the decoder function.

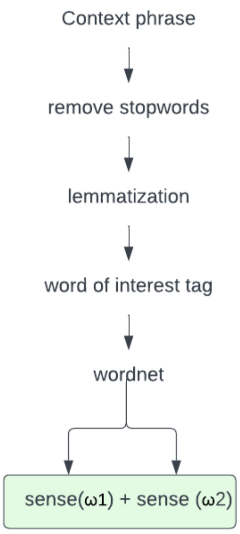


Figure 1: The step-by-step representation of obtaining word senses

An NLP-based approach for determining the similarity between sentences is implemented. This approach starts with loading a dataset including word senses and image captions generated as aforementioned. Each word sense/image caption is preprocessed by using spaCy’s NLP functions[3]. This includes removing stop words, lemmatizing the words, and creating a new text string from the lemmatized words. Then, for each word, the cosine similarity between its concatenated word senses and a caption of each candidate image is computed. The image with the highest similarity is finally selected as an answer.

3.2 Binary Classification Approach

Let w denote a word and $\mathcal{I}$ denote a set of candidate images, this approach determines a pair of w and an image $i \in \mathcal{I}$ whether i is the correct image for w. First, the original dataset is converted to a binary classification dataset. For each word w and each candidate image i, (w, i) is labeled as 1

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

[2] https://github.com/soloist97/Show-And-Tell-Keras

[3] https://spacy.io/

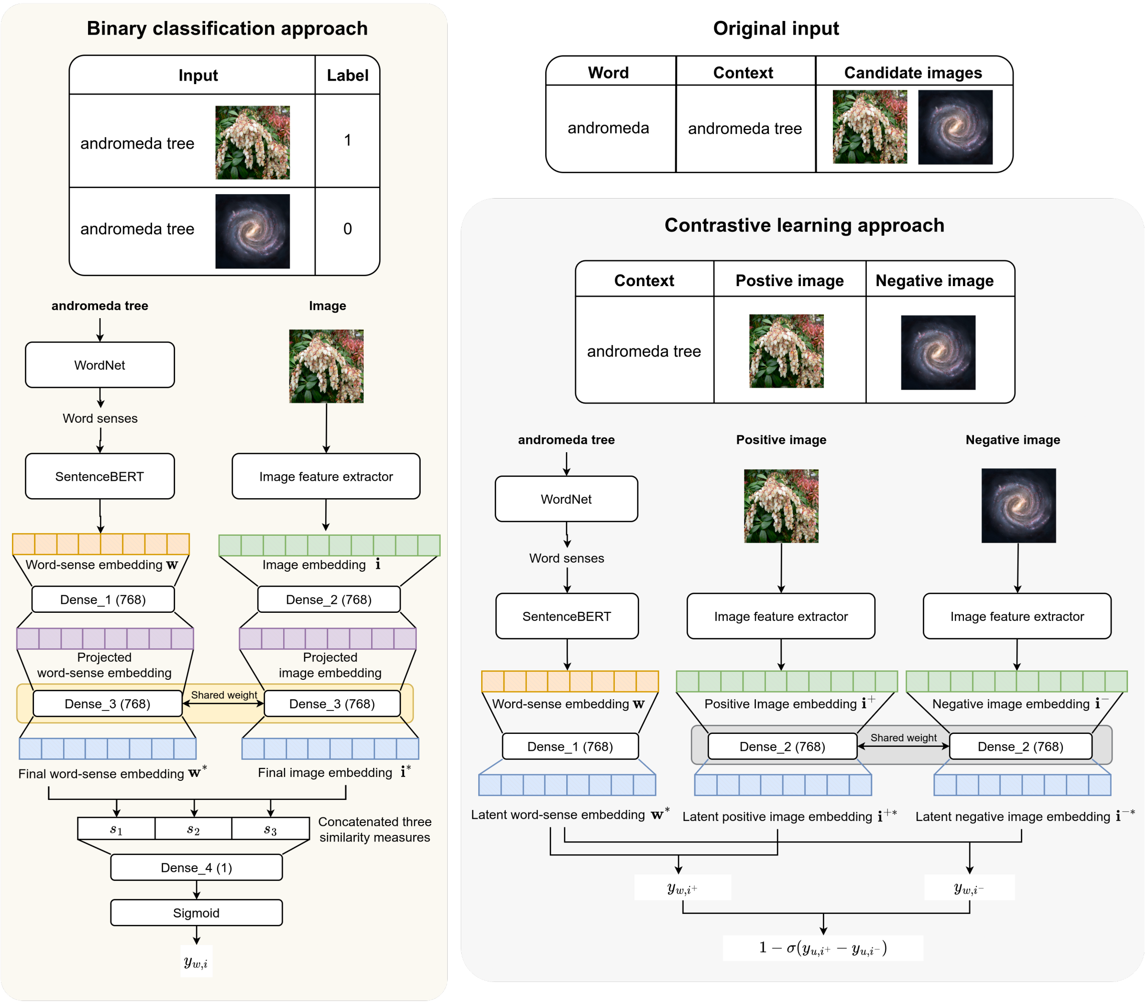

Binary classification approach
Input
Label
andromeda tree
1
andromeda tree
0
andromeda tree
Image
WordNet
Word senses
SentenceBERT
Image feature extractor
Word-sense embedding $\mathbf{w}$
Image embedding $\mathbf{i}$
Dense_1 (768)
Dense_2 (768)
Projected word-sense embedding
Projected image embedding
Dense_3 (768)
Shared weight
Dense_3 (768)
Final word-sense embedding $\mathbf{w}^*$
Final image embedding $\mathbf{i}^*$
s_1
s_2
s_3
Concatenated three similarity measures
Dense_4 (1)
Sigmoid
$y_{w,i}$
Original input
Word
Context
Candidate images
andromeda
andromeda tree
Contrastive learning approach
Context
Postive image
Negative image
andromeda tree
andromeda tree
Positive image
Negative image
WordNet
Word senses
SentenceBERT
Image feature extractor
Image feature extractor
Word-sense embedding $\mathbf{w}$
Positive Image embedding $\mathbf{i}^+$
Negative image embedding $\mathbf{i}^-$
Dense_1 (768)
Dense_2 (768)
Shared weight
Dense_2 (768)
Latent word-sense embedding $\mathbf{w}^*$
Latent positive image embedding $\mathbf{i}^{+*}$
Latent negative image embedding $\mathbf{i}^{-*}$
y_{w,i^+}
y_{w,i^-}
$1 - \sigma(y_{u,i^+} - y_{u,i^-})$

Figure 2: The proposed supervised binary classification approach and semi-supervised contrastive learning approach

if i corresponds to the intended meaning of w (a positive image) and 0 otherwise (a negative image).

For each sample (w, i), a word-sense embedding and an image embedding are used as input. For each w, to obtain a word-sense embedding, its word senses are first generated as in Section 3.1. Then, the pre-trained Sentence-BERT[4] model (Reimers and Gurevych, 2019) is used to generate an embedding of these senses. As for images, their embeddings can be extracted from various methods. In this work, we use ORB (Rublee et al., 2011), KAZE (Alcantarilla et al., 2012), and the pre-trained image classification model VGG16 (Simonyan and Zisserman, 2014) to generate image embeddings. Given an image, ORB and KAZE extract multiple feature vectors depending on the number of key points in that image. A mean pooling method is applied to these feature vectors to obtain an image embedding. For VGG16, an image embedding is extracted from the first fully-connected layer of this model.

We propose a binary classifier based on a Siamese neural network to predict the probability of an image i being a positive image. The idea is to let a classifier learn the similarity between a given word-sense embedding and image embedding as in a Siamese neural network. Siamese neural networks are exceptionally effective for identifying the similarity between two comparable objects such as sentences (Neculoiu et al., 2016) and images (Koch et al., 2015). Therefore, a Siamese neural network is adopted in this work to learn the similarity between word-sense and image embeddings. However, since input word-sense and image embeddings have different sizes, we first project these embeddings into the same D-dimensional latent space so that they have the same dimension. Then, these projected embeddings are passed through a Siamese neural network module to produce final

[4] https://www.sbert.net/

embeddings as follows:

$$\mathbf{w}^* = (\mathbf{W}_3^T(\mathbf{W}_1^T\mathbf{w} + b_1) + b_3) \tag{1}$$

and

$$\mathbf{i}^* = (\mathbf{W}_3^T(\mathbf{W}_2^T\mathbf{i} + b_2) + b_3) \tag{2}$$

where $\mathbf{W}_1$ denotes a weight matrix projecting $\mathbf{w}$ into a D-dimensional latent space, $\mathbf{W}_2$ denotes a weight matrix projecting $\mathbf{i}$ into a D-dimensional latent space, $\mathbf{W}_3$ denotes a shared weight matrix of a Siamese neural network module for learning a final embedding, and b_1, b_2, and b_3 are biases. Based on final embeddings $\mathbf{w}^*$ and $\mathbf{i}^*$, three similarity measures are then computed as follows: $s_1 = |\mathbf{w}^* - \mathbf{i}^*|^2$, $s_2 = |\mathbf{w}^*|^2 - |\mathbf{i}^*|^2$, and $s_3 = \frac{\mathbf{w}^* \cdot \mathbf{i}^*}{||\mathbf{w}^*||||\mathbf{i}^*||}$ where s_1 denotes the square of the difference, s_2 denotes the difference of two squares and s_3 denotes the cosine similarity. These similarity measures are then concatenated and used as input of an output layer, a fully-connected layer with a sigmoid activation function as follows:

$$y_{w,i} = \sigma(\mathbf{W}^T[s_1; s_2; s_3] + b) \tag{3}$$

where $y_{w,i}$ indicates the probability of image i being classified as 1, i.e., image i is corresponding with an intended meaning of w and $\mathbf{W}$ denotes a weight matrix of the output layer. After training, for each w and a given set of candidate images $\mathcal{I}$, an image $i \in \mathcal{I}$ that yields the highest $y_{w,i}$ among other images is selected as an answer.

3.3 Contrastive Learning Approach

This approach also utilizes word-sense embeddings obtained from Sentence-BERT and image features extracted via image processing techniques or pre-trained models (see Section 3.2). Given a word w, this approach learns to rank each image $i \in \mathcal{I}$ based on a score defined as follows:

$$y_{w,i} = (\mathbf{w}^*)^T\mathbf{i}^* + \beta_w + \beta_i \tag{4}$$

where $\mathbf{w}^* = (\mathbf{W}_1^T\mathbf{w} + b_1)$ is a latent embedding of w, $\mathbf{i}^* = (\mathbf{W}_2^T\mathbf{i} + b_2)$ is a latent embedding of i, $\beta_w = (\mathbf{V}_1^T\mathbf{w} + b_1)$ and $\beta_i = (\mathbf{V}_1^T\mathbf{i} + b_1)$ are biases of w and i respectively where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{V}_1$, and $\mathbf{V}_2$ are parameters. To learn these parameters, we adopt a Bayesian personalized ranking (BPR) framework (Rendle et al., 2009) which is a popularly used contrastive learning framework. Let i^+ be a positive image that corresponds to the intended meaning of w and i^- be a negative image that does not correspond to the intended meaning of w. For each word w, a positive image is identified from the original dataset and the rest of the candidate images can be treated as negative images. For each negative image in the candidate set, (w, i^+, i^-) is added to the BPR training set $\mathcal{D}$. Based on this training set, the loss function is defined as

$$\sum_{(u,i^+,i^-)\in\mathcal{D}} 1 - \sigma(y_{u,i^+} - y_{u,i^-}) \tag{5}$$

A stochastic gradient-descent algorithm is adopted for training with this optimization criterion. Similar to the previous approach, an image $i \in \mathcal{I}$ that yields the highest score $y_{w,i}$ is selected as a final output.

4 Experiments

4.1 Experimental Setup

We split the training set provided by the organizers into a training set (80%), a validation set (10%), and a test set (10%). The organizers also provided a trial set during the trial phase and a test set during the evaluation phase. Our approaches were compared with the baselines as follows:

- **Sense-caption**: the proposed unsupervised approach in Section 3.1. All parameters in the NIC model were set as in this repository[5].

- **MLP**: a multi-layer perceptron (MLP) binary classification model using a concatenation of word-sense embedding and image embedding as input. This model consists of a fully-connected layer with an output size of 1024 and a ReLU activation function followed by an output layer with a sigmoid function. An L2-norm regularizer was applied on the fully-connected layer with a regularization factor of 10^{-4}. The model was trained for 500 epochs using an Adam optimizer and binary cross-entropy as a loss function.

- **Siamese**: the proposed binary classification approach based on a Siamese neural network described in Section 3.2. The projected embedding size (D) and the final embedding size were set to 768. An L2 Norm regularization term was added with a regularization factor of 10^{-4} to avoid over-fitting. The model was trained for 500 epochs using an Adam optimizer with a learning rate of 10^{-4}. Binary cross-entropy was used as a loss function.

[5] https://github.com/soloist97/Show-And-Tell-Keras

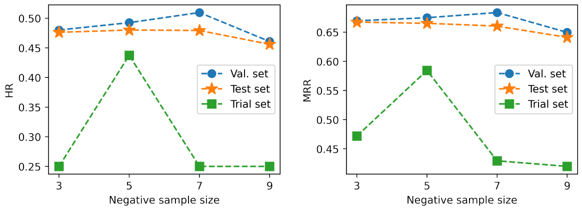
- **BPR**: the proposed approach based on a BPR framework described in Section 3.3. The size of latent embeddings was set to 768. The model was trained for 500 epochs using an Adam optimizer with a learning rate of 10^{-4}.

In the training set, there are 10 images in each set of candidate images, one positive image and 9 negative images. We randomly selected 5 negative images to train **MLP**, **Siamese**, and **BPR** to avoid an imbalanced data problem (i.e., the negative sample size is 5). Hit Ratio (HR) and Mean Reciprocal Rank (MRR) were used for evaluation.

4.2 Results and Discussions

The results are shown in Table 1. From this table, **Sense-caption** was able to obtain HR of 12.14% in the trial set with the largest limitation being spaCy library's large English model, en_core_web_lg, not recognized around 41% of the words. This resulted in **Sense-caption** underperforming the other approaches. Both **Siamese** and **BPR** using VGG16 outperformed the others including the baseline model **MLP**. This indicates that they can learn effective embeddings which can be used to associate word senses with corresponding images. Comparing our proposed approaches, **BPR** using VGG16 only performed slightly better than **Siamese** using VGG16 in terms of HR on the trial set. This suggests that **Siamese** using VGG16, generally performed better than **BPR**. Comparing all three types of image features adopted, **MLP**, **Siamese** and **BPR** using VGG16 performed better than those using ORB and KAZE on the split validation set and the split test set. However, the results on the trial set are the opposite. Using ORB and KAZE gave higher HR and MRR than using VGG16. In fact, by using ORB and KAZE, **MLP** and **BPR** achieved 100% HR and MRR on the trial set. However, since the trial set only consists of 16 samples, results on this set may not entirely reflect the actual model performances. Thus, we decided to submit the predictions from **Siamese** using VGG16 in the evaluation phase because it showed consistent overall performance on the validation set, the split test set, and the trial set. After the evaluation phase, we examined the performances of these models on the test set. Both of our approaches performed better than the other approaches. Comparing our approaches, **BPR** using VGG16 slightly outperformed **Siamese** using VGG16.

We also examined the effect of different negative sample sizes (n) used in **Siamese** using VGG16. Figure 3 shows a comparison of HR and MRR results on the split validation set, the split test set, and the trial set when n was varied among 3, 5, 7, and 9. From this figure, both HR and MRR on the validation set increased as we increased n and then dropped when $n = 9$. For the test set, HR slightly increased when we increased n and dropped after $n = 5$. Meanwhile, MRR continuously decreased as we increased n. On the trial set, HR and MRR peaked at $n = 5$ and declined afterward. These results indicate that using all negative images of each sample may result in lower accuracy than sampling a subset of negative images. Using $n = 5$ achieved better overall performance across various sets compared to the others.



(a) HR

(b) MRR

Figure 3: Comparison of (a) HR and (b) MRR results when varying the negative sample size

5 Conclusions

To solve the task of using WSD in an image retrieval system, this work proposes three approaches: (1) an unsupervised approach using word senses obtained from WordNet and image captions generated from the pre-trained model called NIC, (2) a supervised approach based on a Siamese neural network, and (3) a self-supervised contrastive learning approach using a BPR framework. We conducted experiments on the dataset provided. According to the results, the unsupervised approach underperformed the other approaches due to the limitation of the adopted knowledge base WordNet. The supervised and self-supervised approaches outperformed the baselines in terms of both HR and MRR. We also found that the accuracy of the proposed Siamese model decreased when using all negative images. This indicates an imbalanced data problem resulting in degraded performance.

Model	Image feature	Val. set (split)		Test set (split)		Trial set		Test set	
		HR	MRR	HR	MRR	HR	MRR	HR	MRR
Sense-caption	-	-	-	-	-	12.14	-	-	-
MLP	ORB	24.16	46.10	24.94	46.78	93.75	94.38	11.66	30.83
MLP	KAZE	31.47	53.57	33.88	55.11	**100.00**	**100.00**	9.50	28.63
MLP	VGG16	45.45	64.62	47.40	65.50	12.50	33.67	17.71	38.37
Siamese	ORB	18.49	33.07	19.35	34.98	50.00	55.00	9.07	28.40
Siamese	KAZE	25.72	48.55	28.21	50.22	93.75	94.38	10.37	29.49
Siamese	VGG16	**49.26**	**67.48**	48.02	**66.52**	43.75	58.44	20.52	41.48
BPR	ORB	19.81	43.80	21.52	45.37	**100.00**	**100.00**	11.23	30.45
BPR	KAZE	25.33	48.46	28.21	50.51	**100.00**	**100.00**	11.23	30.45
BPR	VGG16	47.63	66.09	**48.17**	66.04	18.75	43.87	**20.73**	**41.63**

Table 1: Hit Ratio (HR) and Mean Reciprocal Rank (MRR) results on the split validation (val.) set, the split test set, the trial set, and the test set provided by the organizers during the evaluation phase. The highest value in each column is in bold while the second highest one is underlined.

References

Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. 2012. Kaze features. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 214–227. Springer.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Michele Bevilacqua, Tommaso Pasini, Alessandro Raganato, and Roberto Navigli. 2021. Recent trends in word sense disambiguation: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4330–4338. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. 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, Hong Kong, China. Association for Computational Linguistics.

Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*.

Oier Lopez de Lacalle and Eneko Agirre. 2015. A methodology for word sense disambiguation at 90% based on large-scale CrowdSourcing. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 61–70, Denver, Colorado. Association for Computational Linguistics.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2).

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Rep4NLP@ACL*.

Tommaso Pasini and Roberto Navigli. 2017. Train-O-Matic: Large-scale supervised word sense disambiguation in multiple languages without manual training data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 78–88, Copenhagen, Denmark. Association for Computational Linguistics.

Alessandro Raganato, Iacer Calixto, Asahi Ushio, Jose Camacho-Collados, and Mohammad Taher Pilehvar. 2023. SemEval-2023 Task 1: Visual Word Sense Disambiguation. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*.

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*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 452–461, Arlington, Virginia, USA. AUAI Press.

Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. 2011. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Kaveh Taghipour and Hwee Tou Ng. 2015. Semi-supervised word sense disambiguation using word embeddings in general and specific domains. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 314–323.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. 2015. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164, Los Alamitos, CA, USA. IEEE Computer Society.

David Yarowsky. 1993. One sense per collocation. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, page 266–271, USA. Association for Computational Linguistics.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.