StFX NLP at SemEval-2023 Task 1: Multimodal Encoding-based Methods for Visual Word Sense Disambiguation

Yuchen Wei
Department of Computer Science
St. Francis Xavier University
x2020fct@stfx.ca

Milton King
Department of Computer Science
St. Francis Xavier University
mking@stfx.ca

Abstract

SemEval-2023's Task 1, Visual Word Sense Disambiguation, a task about text semantics and visual semantics, is about selecting the best-matched image to represent a target word in a limited context. We explored several methods, including image captioning methods and CLIP-based methods, and submitted our predictions in the competition for this task. This paper will focus on the methods we used and their performance, and provide an analysis and discussion of their performance.

1 Introduction

SemEval-2023's Task 1: Visual Word Sense Disambiguation (V-WSD) involves selecting an image from a list of candidates, that best exhibits a given target word in a small context. In this task (Raganato et al., 2023), each sample will contain one target word, a limited context, and ten candidate images. The ten candidate images contain one gold image - the image that best matches the sense of the target word in its context. In addition to this, the ten candidate images also contain images related to other senses (i.e., the meaning of the word) of the target word as well as images not related to the target word. The limited context contains two or three words (including the target word), and the vast majority of these contexts are strongly related to a sense of the target word. Figure 1 shows one example in the dataset. In this example, the target word is *andromeda* and the context is *andromeda tree*. The first image is the gold image (the tree with white flowers). The second image (whale) and the third image (snake) are some of the candidate images not related to the target word. The fourth image (stars) is the candidate image related to another sense of the word *andromeda*. The dataset for this task contains a trial set, a training set, and a testing set. The context and target words in the samples of the trial set and the training set are in English. The testing set contains samples in English, Italian, and Farsi.

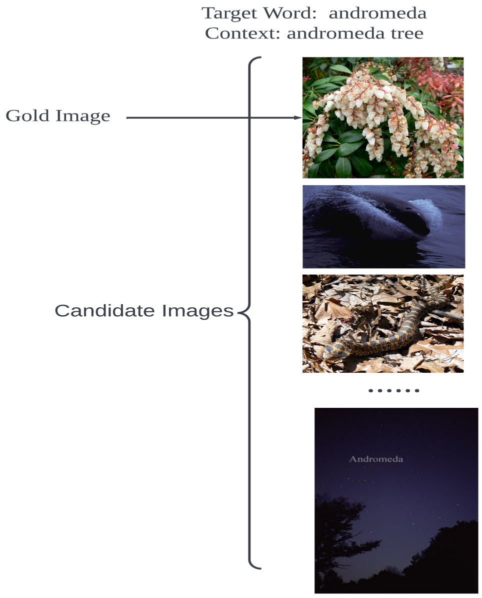


Figure 1: The sample *andromeda*.

.

This task involves both natural language processing and computer vision. This is because we need to acquire semantic features of both text and images. For this task, we chose to map text and images to the same vector space and then compare them using a similarity metric. To perform this feature mapping, we require the use of models that have been trained on other datasets. We approached this task with several pre-trained visual language models. This includes the image caption model (Wang et al., 2022) and Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) models.

2 Related Work

Word sense disambiguation (WSD), a task in natural language processing, is about identifying the

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 409–414
July 13-14, 2023 ©2023 Association for Computational Linguistics

sense of a target word in the context. The most commonly used sense inventory for word senses in WSD is WordNet (Miller, 1995). WordNet is a large lexical database of English senses, which organises word senses in the form of synsets(sets of synonyms). For each word sense(synset), WordNet provides a text definition to describe it.

In the WSD task, we can represent the usage of a word through embeddings, which are word embeddings, the most commonly used representations of words. Word embeddings represent words as dense vectors(low dimensional vectors). There are some commonly used word embeddings: Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), ELMo (Matthew et al., 1802), BERT (Devlin et al., 2018). Researchers such as Wiedemann et al. (2019) have shown that simply using word embeddings can achieve good performance on WSD tasks.

Similar to word embedding, sentence embeddings can represent sentences in a low-dimension vector. The sentence embedding technique used in this study is sentence BERT (Reimers and Gurevych, 2019), also known as the sentence transformer. It is trained on multiple corpora, such as Reddit comments (Henderson et al., 2019) and S2ORC (Lo et al., 2020), and used to generate sentence embeddings for sentence similarity.

In recent years, progress has been made in the development of multimodal models, which take multiple types of data(like images and text) as learning subjects. Multimodal models have been applied to image captioning tasks such as Xu et al. (2015) and Anderson et al. (2018). A state-of-the-art task-agnostic and modality-agnostic framework, called OFA (Wang et al., 2022) has outstanding results on a range of vision-language tasks. Microsoft COCO Captions (Chen et al., 2015) is one of the most commonly used datasets for image captioning tasks. It contains 330,000 images with roughly 1.5 million captions, making it the largest image caption dataset available, with captions generated by human annotation. OFA framework has achieved great performance on Microsoft COCO Captions.

In 2021, OpenAI released a pre-trained neural network model for matching images and text, named CLIP (Contrastive Language-Image Pre-Training) (Radford et al., 2021). The model is trained on over 400 million image text pairs collected by OpenAI on the internet to encode images and text. It then trains with the aim of improving the similarity between the encodings, resulting in a multimodal pre-trained model that generates vision-language embeddings.

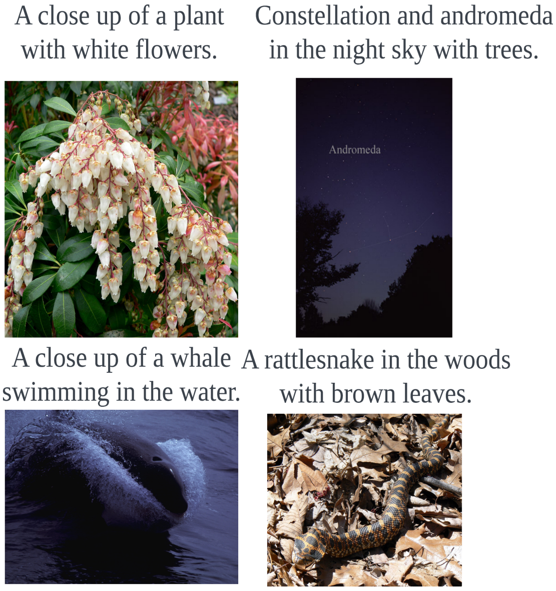
3 Methods

In this section, we will introduce the methods that we applied to the V-WSD task. A common method used in comparing the similarity of vector spaces is cosine, which we use in each of our models. The cosine similarity between numerical vectors that represent semantic features is first calculated, and then the similarity is ranked according to the cosine value. Although this method is relatively simple, it is very effective in comparing similarities.

3.1 Image Captioning Method

In this research, the first method we proposed is to generate captions for the images using a pre-trained image caption model, and then use these captions to compare the text similarity with the synset definitions of the target words' senses in WordNet.

We use the pre-trained OFA framework to generate a caption for each image in the samples. The OFA framework pre-trained on Microsoft COCO Captions comes from ModelScope [1], an open-source model-sharing platform. Figure 2 shows some generated captions for some images of the sample *andromeda tree*.


A close up of a plant with white flowers.

Constellation and andromeda in the night sky with trees.

A close up of a whale swimming in the water.

A rattlesnake in the woods with brown leaves.

Figure 2: Image captions generated by the pre-trained OFA framework.

[1] https://modelscope.cn/home

The image captioning method includes determining the sense of the target word in context. To find the word sense of the target word, we compared the cosine similarity of the word embedding of the target word and context words with the word embedding of each word in the defined text. For the word embeddings, we choose to use GloVe, which has 300 dimensions per word embedding and is obtained after training on Wikipedia 2014 and English Gigaword Fifth Edition. For the context text (including target word and context words), we generate word embeddings: c_1, c_2, ... Each c_i represents the word embedding of each token in the context and the context for the majority of the samples contain two or three tokens. For each synset of the target word, we generate word embeddings for each word in the definition text (i.e., the definition of the synset in WordNet): d_1, d_2, ..., d_n. For each synset, we calculate the cosine of each $\mathrm{d_i}$ with each $\mathrm{c_i}$ in the context text. The average of the three highest cosine values is then taken as the similarity score of the synset. Finally, the synset with the highest score is selected as the sense of the target word in the sample.

After this, we generate sentence embeddings s_d for the defined text of the synset by using the sentence BERT. For each candidate image caption, we also generate the corresponding sentence embeddings by using the sentence BERT: s_1, s_2, ..., s_{10}. Finally, by calculating the cosine of s_d and each s_i, the image with the highest cosine is selected as the predicted image.

3.2 CLIP Methods

In this method, we use the CLIP model to generate language-vision embeddings for the context text and candidate images. For the context text, we add "a photo of" as a prefix, as this leads to a better text feature vector (Radford et al., 2021). We calculate the cosine between each image's CLIP embedding and the context CLIP embedding as the similarity score. Finally, the image with the highest cosine value is chosen as the prediction image.

Regarding the pre-trained CLIP model, we tried two different versions. The difference between them is that they use two different vision transformers: ViT-B/32 and ViT-L/14. The CLIP model with ViT-B/32 generates a uniform 512-dimensional feature embedding for text and images, while the dimensionality of the CLIP model with ViT-L/14 is 768.

In addition to using cosine as a metric, we tried to use other methods similar to cosine similarity. Assume for each candidate image, the CLIP vector is v = [v_1, v_2, v_3, ..., v_n] and the context vector is t = [t_1, t_2, t_3, ..., t_n]. For each candidate image, we construct its feature vector f = [f_1, f_2, f_3,......, f_n] and $f_\mathrm{i} = \frac{v_\mathrm{i} \cdot t_\mathrm{i}}{|v| \cdot |t|}$. For the feature vector f, the sum of each element is the cosine between the corresponding candidate image CLIP vector and the corresponding context CLIP vector. We want to reassign the weights of each f_i in the sum to obtain the score for each candidate image better. For the method of assigning weights, we used either a neural network or a linear regression.

For the linear regression model, we use the binary classification method and take the candidate image with the highest prediction value as the predicted image (i.e., *1* for gold images' labels and *0* for non-gold images' labels). For the neural network model, we use the multiclassification approach, where the label of a sample is a vector containing one *1* (gold image) and nine *0*s (non-gold images). The neural network we designed contains one fully-connected layer with the number of units equal to 10 times the CLIP vector dimension, one dropout layer with the dropout rate equal to 0.5, and one fully-connected layer with the number of units equal to 10. The activation function of the first fully-connected layer is ReLU and the activation function of the second fully-connected layer is Softmax.

4 Experimental Results

This section presents the results of our experiments. We use the hit rate and mean reciprocal rank (MRR) as our evaluation metrics.

We submitted the prediction results on the English test set for two methods using the ViT-B/32 CLIP model. One is the method that uses cosine which achieved a hit rate of 58.3%, an MRR of 72.1, and placed 62nd in the task, and one is the method that uses linear regression which achieved a hit rate of 59.2%, an MRR of 73.0, and placed 57th in the task. They performed slightly below the organizer's baseline model (also using CLIP), which achieved a hit rate of 60.5%, an MRR of 73.9, and placed 54th on the English test set. Since we did not try the ViT-L/14 CLIP model before submission, we only submitted the method using the ViT-B/32 CLIP model. These results are shown in Table 1.

Method Name	Hit Rate	MRR	Rank
Cosine CLIP	58.3%	72.1	62nd
Linear Regression CLIP	59.2%	73.0	57th
Organizer's Baseline	60.5%	73.9	54th

Table 1: The prediction results we submitted and the prediction result of the organizer's baseline using CLIP.

In the remainder of this section, we will discuss the performance of our models on the training set, which contains 12,869 samples. The experimental results of all methods are shown in Table 2.

Method Name	Hit Rate	MRR
Image Captioning Method	56.8%	70.6
ViT-B/32 CLIP Model		
Cosine	74.9%	83.7
Linear Regression	79.3%	86.9
Neural Network	76.7%	85.1
ViT-L/14 CLIP Model		
Cosine	79.5%	86.5
Linear Regression	86.0%	91.2
Neural Network	84.1%	90.1

Table 2: Experimental results of the image captioning method and CLIP methods. The results of the image captioning method and cosine CLIP methods come from the test of the entire training set, and the results of the linear regression and neural network CLIP methods come from the test of K-fold with K=5 on the training set.

4.1 Image Captioning Method

The method using the image captioning model introduced in Section 3 has a hit rate of 56.8% and an MMR of 70.6 on the training set, which is much smaller than methods using the CLIP model. We think there are three main reasons for this result.

First of all, this prediction method needs to predict word senses first. And once the word senses are predicted incorrectly, it could directly lead to predicting the wrong images. For example, when predicting the word sense of *bank* in *bank erosion*, a wrong prediction, such as predicting a financial institution, will directly give the remaining part of the method incorrect information.

Secondly, many WordNet definitions of word senses do not contain visual aspects. For example, the definitions of the *anteater*'s word senses mainly include the geographical location of different types of anteaters and do not include much about the appearance features of different kinds of anteaters. In addition to this, some word senses are often abstract concepts, like the sense of *administration* in *administration minister*, which makes the captions of the images difficult to relate to the definitions.

Finally, it is the over-simplicity of the captions we generate on the images by the OFA framework trained on Microsoft COCO Captions. Although they give a good overview of what is in the images, they do not include too many visual details of the images. This is because each image caption in Microsoft COCO Captions usually contains only one sentence about the objects in the image. In the V-WSD task, more complex visual features are required to predict some samples. For example, in predicting the *anteater* in the context *marsupial anteater*, this method needs to know not only that it is a species of mammal, but also its appearance features, such as the striped fur. Otherwise, this method may take other species of anteater as the prediction result.

4.2 CLIP Methods

The results of all experiments of the CLIP methods are shown in Table 2. In Table 2, the performance of ViT-L/14 CLIP is better than ViT-B/32 CLIP. Among them, the CLIP methods, which simply use cosine as the score, achieved 74.9% (ViT-B/32) and 79.5% (ViT-L/14) hit rates. They also have very high MMRs, which are equal to 83.7 (ViT-B/32) and 86.5 (ViT-L/14), compared with our image captioning method. They are better than our image captioning method by about 20% on the hit rate and 15 on the MMR. This is could due to the fact that the text and image vectors generated by CLIP models trained on very large vision-language data could do better at representing semantic features.

Using linear regression and neural networks to assign weights requires training, and we used K-fold with K=5 to cross-validate each method. In each validation, 20 % of the dataset is used for validation and 80 % for training. Finally, the averages of the five hit rates and MRRs are used as the performance score and shown in Table 2.

Regarding the neural network model, the final hit rates are higher than the method that simply uses cosine by about 3% on the hit rate and 3 on the MRR. Figure 3 and Figure 4 show the training history of the neural network method using the ViT-B/32 CLIP model and the neural network method using the ViT-L/14 CLIP model, respectively.

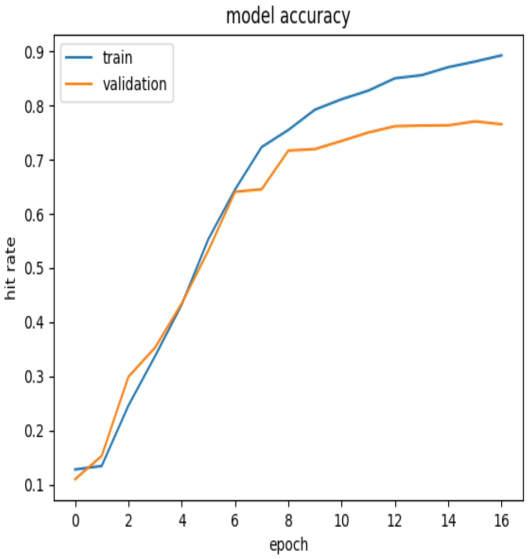


Figure 3: Training history of the neural network method using ViT-B/32 CLIP model.

The hit rates of the linear regression model are better than the method simply using cosine, about 5% to 6% higher. This shows that using linear regression to reassign the weights in the summation when computing the vector cosine is useful in measuring similarity. Table 1 shows that the prediction results, which we submitted, on the test set, are slightly below the task organizer's CLIP baseline on both hit rate (by 1.3%) and MRR (by 0.9). The difference between the organizer's CLIP-based model and ours is that they tried other textual prefixes when generating the context vector, such as "Example of an image caption that explains".

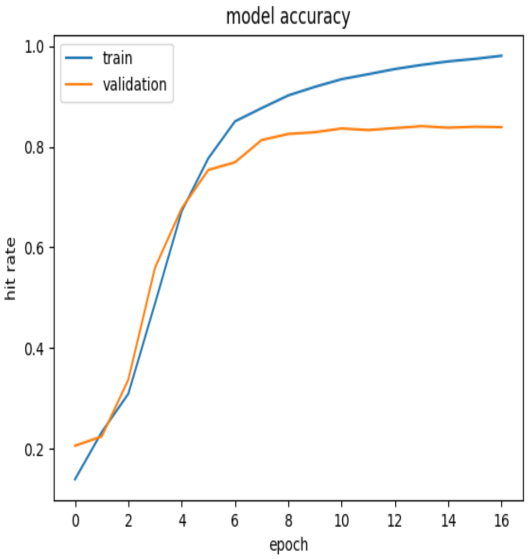


Figure 4: Training history of the neural network method using ViT-L/14 CLIP model.

We think that one reason why the CLIP model can achieve such good results on the training set is that many of the images in the training set originate from the Internet, which is where CLIP gathers its dataset for training over the image-text pairs. When we use the search engine to search for images by entering context and target word, the images we get happen to contain the gold image. As to why the test results of our method on the training set differ so much from the test results on the test set, it may be because the images in the test set could have been gathered after CLIP was trained, which might be why we see a decrease in performance between the training and test set.

5 Conclusion

In this study, we used several methods for the V-WSD task in SemEval-2023. The main approaches include the one using the image caption model and the one using the CLIP model. The method using the image caption model does not perform well, probably because the generated image captions are too simple and some word definitions are difficult to associate with image captions. The methods using the CLIP model are better, reflecting the powerful ability of the CLIP model to generate uniform vision-language embeddings and its effectiveness in comparing image and text similarity. We also used other methods, including a neural network and a linear regression, to reassign the weights of the elements in the summation phase when calculating the cosine similarity. They all have better performance than simply using cosine similarity.

6 Future Work

In the future, we will explore constructing a model similar to CLIP that can generate visual features for word senses. We explore this proposed model on tasks such as V-WSD.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on* *computer vision and pattern recognition*, pages 6077–6086.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Matthew Henderson, Pawel Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrksic, Georgios Spithourakis, Pei-Hao Su, Ivan Vulic, and Tsung-Hsien Wen. 2019. A repository of conversational datasets. *CoRR*, abs/1904.06472.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

E Peters Matthew, N Mark, I Mohit, G Matt, C Christopher, and L Kenton. 1802. Deep contextualized word representations (2018). *arXiv preprint arXiv:1802.05365*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

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

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

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)*, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings. *arXiv preprint arXiv:1909.10430*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.