Rahul Patil at SemEval-2023 Task 1: V-WSD: Visual Word Sense Disambiguation

Rahul Patil[†]**, Pinal Patel**[*]**, Charin Patel**[*]**, Mangal Verma**[*]
[*]RAAPID.ai
[†]HEALTHCARENLP SOFTECH LLP
[†]{rahul.p}@healthcarenlp.com, [*]{pinal,charin,mangal}@raapid.ai

Abstract

Semeval 2023 task 1: Visual-WSD, In this paper, we propose an ensemble of two Neural network systems that ranks 10 images given a word and limited textual context. We have used OpenAI CLIP based models for the English language and multilingual text-to-text translation models for Farsi-to-English and Italian-to-English. Additionally, we propose a system that learns from multilingual bert-base embeddings for text and ResNet101 embeddings for the image. Considering all three languages into account this system has achieved the fourth rank.

1 Introduction

Visual word sense disambiguation (Visual-WSD) (Raganato et al., 2023) is an important task for understanding and measuring machine's ability to find correct image from a set of candidate images given the word and the limited text context.

Word sense disambiguation (WSD) has long been formulated and studied as an important problem in natural language processing since 1940s. Visual Word Sense Disambiguation (Visual-WSD) is a combination of natural language processing and computer vision that involves identifying the correct visual image of an ambiguous word using limited context. For example, the word "bank" can refer to a financial institution or the side of a river. In the context of limited text data, such as "bank erosion", Visual-WSD aims to identify the image displaying the correct meaning of the word in its context. Visual-WSD is an important task in natural language processing and computer vision, as it helps to improve the accuracy of systems that use text and visual information together, such as image captioning, video analysis and visual question answering. Successful Visual-WSD can help improve the accuracy of natural language processing applications that rely on visual information.

We used an ensemble of two different approaches 1) Fine-tune CLIP (Radford et al., 2021) base architecture 2) Joint Embedding. The first approach involves finetuning CLIP based model with additional data and the second approach involves mapping image and text-based features into a common vector space, which allows direct comparison of the two features. This enables the model to learn the relationship between visual and text features and use this knowledge to determine the correct sense of a word.

The proposed system has achieved the 4th overall rank in this competition with the hit rate of 69.805 and mean reciprocal rank score of 78.230. The system has achieved the second rank in English language with the HIT rate of 83.153 and MRR rate of 88.802, first rank in Italian with the HIT rate of 84.262 and MRR rate of 89.048, lastly achieved 16th rank in Farsi language with the HIT rate of 42.000 and MRR rate of 56.841. The system lacked understanding of Farsi language and struggle with Farsi to English conversion.

2 Background

Three different types of datasets were released in this competition. Trial and train dataset contained words, limited context, 10 candidate images and correctly labeled gold data. Test dataset consists of word, limited context and 10 candidate images.

In the figure1: bank is an ambiguous word. It can have multiple meanings like river bank, erosion of bank, money bank, piggy bank, turning of airplane. In this figure, context is mentioned as an erosion bank hence the gold label.

The train and the trial data consists of English language only, whereas test data includes English, Italian and Farsi languages. Additionally we utilized the MS-COCO (Lin et al., 2015) dataset in conjunction with the Wikipedia Image Text (WIT) dataset (Srinivasan et al., 2021), a vast multimodal and multilingual dataset sourced from Wikipedia.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1271–1275
July 13-14, 2023 ©2023 Association for Computational Linguistics

Table 1: EDA: train-test-trial Dataset statistics

Dataset	Count of Data points	Contexts per unique word	Count of Word is present in context	Data coverage of Top 5 words	Data coverage of Top 10 words	Data coverage of Top 50 words	Data coverage of Top 100 words
Train+Trial	12885	1.03	12540	0.171	0.311	0.427	1.9351
Test [all]	968	1.56	621	2.273	4.339	6.405	27.789
Test [EN]	463	1.79	259	4.32	7.991	11.231	50.324
Test [IT]	305	1.46	208	5.574	10.492	15.41	64.59
Test [FA]	200	1.30	154	9.5	15.0	20.0	73.0

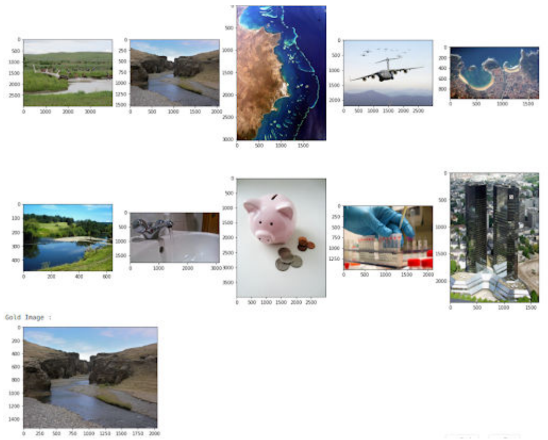

Figure 1: Example of candidate images and gold label; Word : bank; Limited Context : bank erosion.

As mentioned in table 1, the train + trial data distribution is slightly different from test dataset. In train + trial data we have 1.03 contexts per unique word whereas test data has 1.56. Percentage of data covered by top 5 words in train+trial dataset if 0.171% whereas for the test dataset, it is 2.273%.

3 System overview

3.1 Fine-tune CLIP

CLIP stands for "Contrastive Language-Image Pre-Training". It is a machine learning algorithm developed by OpenAI that is trained to learn the relationship between text and images by pre-training on a massive dataset of images and associated text captions. We also used data from MS-COCO dataset (English captions) and WIT dataset (English, Farsi and Italian dataset).

In the CLIP architecture (Figure 2), we take an image-text caption as input. During prediction, we provide a collection of candidate images (i) and a set of text captions (t). Model generates similarity score with i*t dimensional matrix. We used this output for identifying the best matching image for a given text caption. In our case we had 12,885 labeled image-text pairs. CLIP architecture consists of 3 main components:

3.1.1 Image Encoder

This component is responsible for encoding the input image into a set of feature vectors. These features are able to capture different levels of visual information, from low-level features such as edges and colors to high-level features such as object and scene semantics. We utilized ResNet50 as the backbone for our CLIP model.

3.1.2 Text Encoder

This component encodes the input text prompt or caption into a set of feature vectors that capture the semantic meaning of the text. The encoder is a transformer-based neural network and generally works well with sentence structure. To utilize the network at its best, we augmented 10 sentences from a given word-context pair and used them as captions for a specific image.

3.1.3 Contrastive Learning Objective

This component trains the image and text encoders to map similar images and text pairs close together in the shared feature space and dissimilar pairs far apart. This is done by maximizing the agreement between the similarity scores of positive pairs (i.e., images and text that belong together) and the similarity scores of negative pairs (i.e., images and text that do not belong together).

By jointly embedding images and text in a shared feature space, CLIP can perform a variety of tasks, such as zero-shot classification and image retrieval, by simply comparing the distances between the embeddings of different images and text prompts. This allows CLIP to generalize to new tasks and domains without the need for task-specific training data.

At the end, We optimize the CLIP model using symmetric cross entropy loss over these similarity scores. According to the CLIP paper, despite

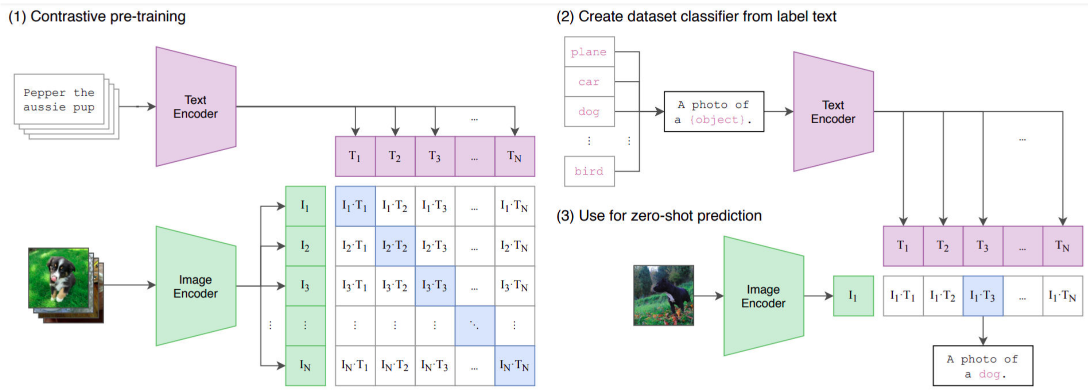

Figure 2: OpenAI-CLIP model architecture

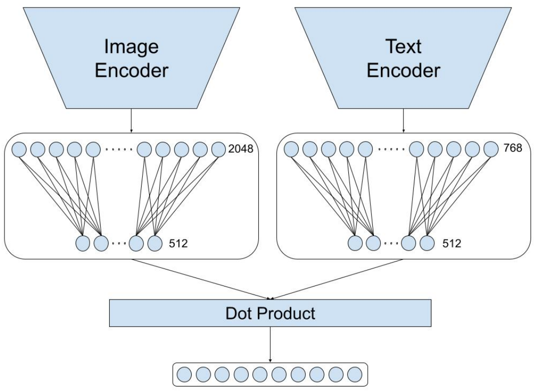

Figure 3: Joint Embedding [1 caption vs 10 images] Model Architecture

observing an increase in validation loss, we continued training the model for a few additional epochs, resulting in improved accuracy on the test dataset.

3.2 Joint Embedding [1 caption vs 10 images]

CLIP architecture tries to improve symmetric cross entropy loss which contains loss due to images and loss due to text captions. In our problem statement we have 10 images and one word-context pair and we have to rank candidate images which are best suitable for give text and context pair. Hence we propose a slight variation of CLIP architecture (Figure 3) and its components are:

3.2.1 Image Encoder

We used pretrained ResNet101 model from pytorch model hub (maintainers and contributors, 2016) and applied fully connected layer (FC layer) for dimensionality reduction followed by normalization layer and dropout layer. We extracted 512 dimensional vectors for all 10 images.

3.2.2 Text Encoder

We used pretrained multilingual SBERT architecture "sentence-transformers/distiluse-base-multilingual-cased-v2" (Reimers and Gurevych, 2019) from hugging face. We generated 768 dimensional vectors for each word-context pair. We added dropout layer and fully connected (FC) layer for dimensionality reduction followed by normalization layer and dropout layer to generate 512 dimentional vector.

3.2.3 Learning Objective

Finally, we utilized cosine similarity to compare the image embeddings and text embeddings, while adopting cross-entropy as the loss function.

Unlike fine-tuned clip, we did not apply preprocessing to the text or image data. However, we rotated the sequence of candidate images to prevent any single image class from dominating during training or prediction.

4 Experimental setup

4.1 Preprocessing

In our case, we did not used any image augmentations but we used sentences formed using words and limited context. For the example mentioned in figure 1, we generated following text augmentations:

this is a photo of a bank erosion
this is a photo of a bank erosion, a type of bank
a photo of a bank erosion
a photo of a bank erosion, a type of bank
this is a photo of a bank
this is a photo of a bank in context of bank
a photo of a bank in context of bank erosion
a photo of a bank in sense of bank erosion
a photo of a bank erosion in context of bank

a photo of a bank erosion in sense of bank

We merged the trial dataset, train dataset, MS-COCO, and WIT dataset to create a comprehensive training dataset. Prior to training, we performed basic preprocessing steps on the strings, such as removing extra spaces and replacing emojis with relevant words. Additionally, we leveraged the Helsinki model from HuggingFace (Tiedemann and Thottingal, 2020) to convert text and context words from Italian to English and persiannlp model from hugginface (Daniel Khashabi, 2020) and Farsi to English.

4.2 Evaluation metrics

In this competition organizers proposed the following evaluation metrics:

4.2.1 Hit Rate (HR)

HR is an abbreviation for hit rate, which denotes the proportion of word-context pairs in which the correct image is predicted with the highest score. As demonstrated, a higher hit ratio indicates a greater likelihood that the correct image is ranked in the top 1 predicted images.

4.2.2 Mean Reciprocal Rank (MRR)

The Mean Reciprocal Rank (MRR) is a statistic measure for evaluating any process that produces a list of possible responses to a sample of queries, ordered by probability of correctness. Similar to the HR metric, MRR is higher the better metric.

We Trained our system on a single Tesla T4 GPU for 4 days. We used Adam optimizer with learning rate of 5e-5, beta values of (0.9, 0.98), eps of 1e-6 and weight decay of 0.2. Here the learning rate is smaller compared to the one mentioned on paper. After experimentation, we determined that this learning rate was best suited for fine-tuning the entire dataset.

Although we attempted to use the ViT-B/32 model while fine-tuning CLIP model as an alternative to ResNet50, the training time required for ViT-B/32 proved to be impractical for our hardware configuration.

5 Results

While fine-tuning the CLIP model, we placed greater emphasis on loss values rather than HIT rate or MRR. Additionally, we followed the approach outlined in the OpenAI paper and overfitted our model, resulting in enhanced MRR and HIT scores, despite observing an increase in validation loss.

Unlike a fine-tuned CLIP model, we opted to stop training our Joint Embedding 1 vs 10 model once the validation loss ceased to improve. This approach helped us prevent overfitting and ensured that our model remained robust. Furthermore, we evaluated best-performing model using multiple metrics including MRR, HIT rate and validation loss.

Table 2: Results

Model	Hit Rate	MRR	Rank
Fine-tune CLIP [ALL]	69.80	78.23	4
Fine-tune CLIP [EN]	83.15	88.80	2
Fine-tune CLIP [IT]	84.26	89.04	1
Fine-tune CLIP [FA]	42.00	56.84	23
Joint Embed [1vs10][ALL]	69.56	76.57	5
Joint Embed [1vs10][EN]	82.93	87.21	3
Joint Embed [1vs10][IT]	84.26	87.49	2
Joint Embed [1vs10][FA]	41.50	55.02	25

6 Conclusion

Our proposed solution for task SemEval-2023 Task 1: V-WSD is able bridge the gap between computer vision and natural language processing for English and Italian language. As mentioned in table 2, we achieved 1st and 2nd rank in IT and EN language respectively. Improving performance for the Farsi language may necessitate the use of additional data and/or more sophisticated pretrained multilingual models. We believe that our solution has zero shot learning capability and can also work on unseen data.

References

Siamak Shakeri Pedram Hosseini Pouya Pezeshkpour Malihe Alikhani Moin Aminnaseri Marzieh Bitaab Faeze Brahman Sarik Ghazarian Mozhdeh Gheini Arman Kabiri Rabeeh Karimi Mahabadi Omid Memarrast Ahmadreza Mosallanezhad Erfan Noury Shahab Raji Mohammad Sadegh Rasooli Sepideh Sadeghi Erfan Sadeqi Azer Niloofar Safi Samghabadi Mahsa Shafaei Saber Sheybani Ali Tazarv Yadollah Yaghoobzadeh Daniel Khashabi, Arman Cohan. 2020. ParsiNLU: a suite of language understanding challenges for persian. *arXiv*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

TorchVision maintainers and contributors. 2016. Torchvision: Pytorch's computer vision library. https://github.com/pytorch/vision.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

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. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. WIT: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.