Word Sense Disambiguation

Multilingual Natural Language Processing HW2

Lorenzo Ciarpaglini

August 8, 2025

Abstract

This report presents an approach to Word Sense Disambiguation (WSD), a critical task in natural language processing that involves identifying the correct meaning of a word within a specific context. My focus is on both coarse-grained and fine-grained datasets. The core idea of my proposed model lies in the utilization of sense-definition embeddings, which involves embedding the definitions of word senses relative to candidate clusters.

1 Introduction

In this report, I address the task of Word Sense Disambiguation (WSD), focusing primarily on the coarse-grained aspect of the task. My main objective is to accurately assign a given word in a sentence to the appropriate semantic cluster from a set of candidates. Each cluster encapsulates a range of senses, which can be used to solve both the main task on coarse-grained dataset, but also for a more difficult task on the fine-grained dataset. In the subsequent sections of this report, I will explore various approaches to tackling the coarse-grained and fine-grained WSD task and I will analyze the effectiveness of these methods.

2 Methods

In this section, I outline the various approaches I have adopted for WSD, focusing on both coarse-grained and fine-grained tasks.

2.1 Baseline Coarse-Grained

My baseline method involves creating embeddings for lemmas and POS tags of words, which are then concatenated with embeddings generated by a BERT transformer for each word. This concatenated representation is passed through a linear layer that serves as a classifier. In figure 2.a is reported the complete Baseline architecture.

2.2 Glossary Coarse-Grained

For each sense in the candidate clusters, I calculate the sentence embedding of the sense definition using a BERT model, focusing on the embedding corresponding to the [CLS] token. Simultaneously, I compute an embedding for the input sentence, selecting the embedding of a given target word to be disambiguated using again BERT. I then calculate the cosine similarity between the embedding of the target word and the sentence embedding of every sense definition across the clusters, choosing the one with the highest cosine similarity. Figure 1 provides a visual representation that elucidates this step, while in figure 2.b is reported the complete Glossary architecture. This approach was inspired by the Terra Blevins, Luke Zettlemoyer work explained in [1].

This selected embedding is added to the embedding of the target word. The complete new embedding of the sentence to be disambiguated is subsequently passed through two linear layers to reduce its dimension before being fed into a classifier. This process is repeated for each word to be disambiguated in the sentence. During the inference phase I select the candidate with the highest cosine similarity value.

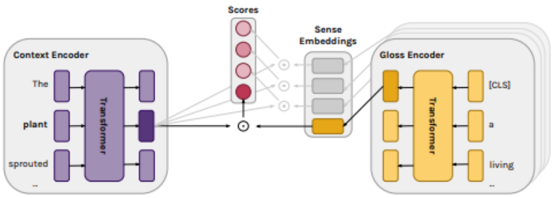


Figure 1: This figure illustrates the process employed for calculating the sentence embedding of each sense definition and its corresponding cosine similarity with the word to be disambiguated.

2.3 Glossary for Fine-Grained

This method is similar to the Glossary approach. Here, I create a sentence embedding for all candidate senses using BERT, selecting the [CLS] token embedding for representing the sentence. For each of these sentences, I calculate a dot product with the embedding of the word to be disambiguated, resulting in an embedding vector where each element corresponds to the dot product between the disambiguation word and a candidate sense's definition. Finally I select the candidate with the highest dot product value.

3 Setup

In preparing the dataset I conducted an initial analysis to determine the average length of sentences. Based on this analysis, I decided to remove all samples that were significantly longer than this average length. This decision was driven by two main considerations: first of all, to exclude outliers that could skew the model's performance, and secondly, to manage computational resources more efficiently, since in a batch, all sentences are padded to match the length of the longest sentence and this practice often led to memory overflow.

4 Experiments

I conducted several tests to evaluate the effectiveness of the proposed models.

4.1 Coarse-Grained base experiments

Initially, I trained the Baseline and GlossaryCoarse models on the coarse-grained dataset. To assess their performance, I monitored the accuracy and losses during training and validation. The results of these measurements are comprehensively depicted in Figures 3, and 4, with each figure corresponding to the performance metrics of both models. From these figures, we can discern that the Glossary-based model outperforms the baseline. It achieves an accuracy of 0.96 on the training set and 0.92 on the validation set. In contrast, the baseline model reaches an accuracy of 0.92 on the training set and 0.90 on the validation set.

4.2 Fine-Grained base experiment

In parallel, I trained a model specifically for the fine-grained WSD task(GlossaryFine model). The accuracy and loss results from this training are detailed in Figures 5.

From the Figure 5.a, it's evident that after two epochs of training, the model begins to deteriorate, with its accuracy dropping from 0.85 to 0.50 on the training set. A similar pattern is observed on the validation set. This led me to halt the training process and select the best parameters obtained after the initial two epochs.

4.3 Coarse-Grained Task using Fine-Grained Model

After training the models on their respective datasets, I embarked on an another experiment. I attempted to resolve the coarse-grained task using the GlossaryFine model trained on the fine-grained dataset. This involved selecting the sense output by the GlossaryFine model and then mapping it back to its corresponding cluster using a dictionary that linked each fine sense to its cluster. This method reached an accuracy of 0.918 on the test set.

4.4 Fine-Grained Task filtering with Coarse-Grained Model

The Fourth experiment was focused on addressing the fine-grained task, but with a twist. I first applied the GlossaryCoarse model to narrow down the candidate senses to those belonging to the correct cluster. Following this preliminary step, I then applied the GlossaryFine model, but with the candidates limited to those identified by the coarse-grained model. This approach gets an accuracy of 0.80 which shows no significant improvement over the only GlossaryFine model.

5 Conclusion

The results of the previous experiments suggest that the GlossaryFine model excels at determining the senses belonging to the correct cluster, but it is less effective in pinpointing the exact sense within that cluster. When used for resolving the coarse-grained task, this model achieves an impressive accuracy of 0.91. However, when the correct cluster is filtered out, the model shows a diminished capability in accurately identifying the precise sense within the cluster. In conclusion the model that gets the highest accuracy on the Coarse-grained task is the GlossaryCoarse model alone which reaches an accuracy of 0.92 on the test set.

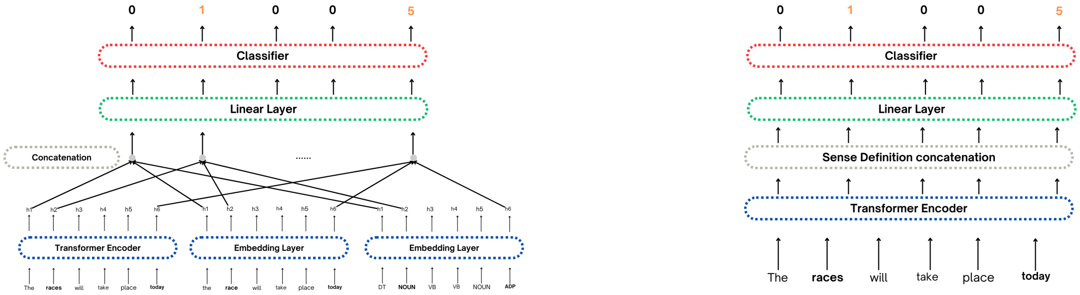
Finally, all the results obtained on the test sets for each model and each experiment are comprehensively documented in Table 1.

References

[1] Luke Zettlemoyer Terra Blevins. *Moving Down the Long Tail of Word Sense Disambiguation with Gloss Informed Bi-encoders.* 2020.

Experiment	**Coarse-Grained test set**	**Fine-Grained test set**
Baseline	0.89	...
GlossaryCoarse	**0.92**	...
GlossaryFine	0.91	0.79
GlossaryCoarse + GlossaryFine	...	**0.80**

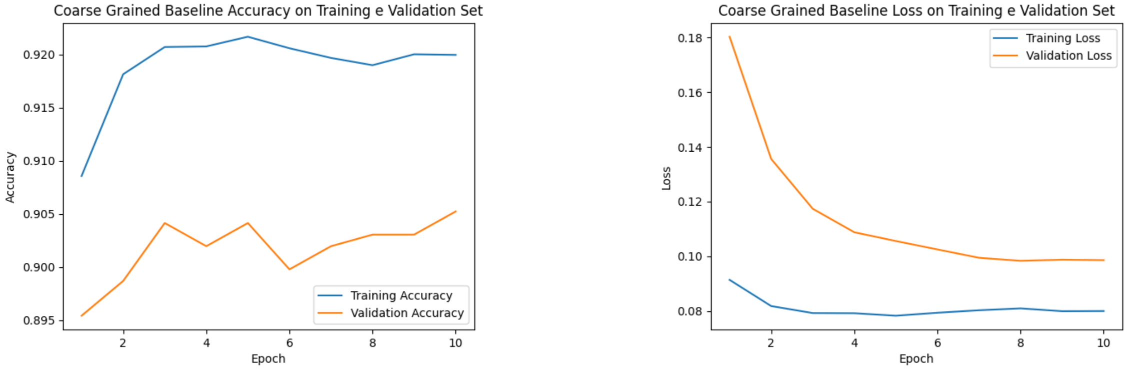
Table 1: Results for each experiments conducted on both Coarse-Grained and Fine-Grained test sets



(a) Baseline Architecture

(b) Glossary Complete Architecture

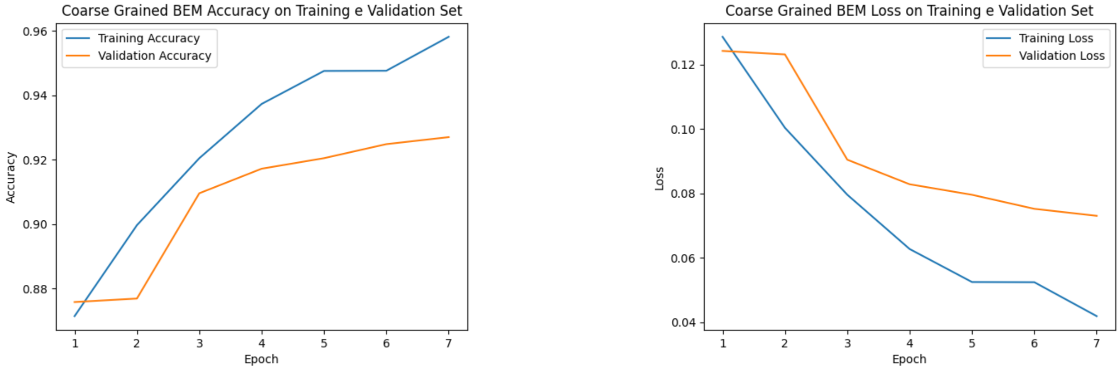
Figure 2



(a) Baseline Coarse-Grained Accuracy on Training and Validation set

(b) Baseline Coarse-Grained Loss on Training and Validation set

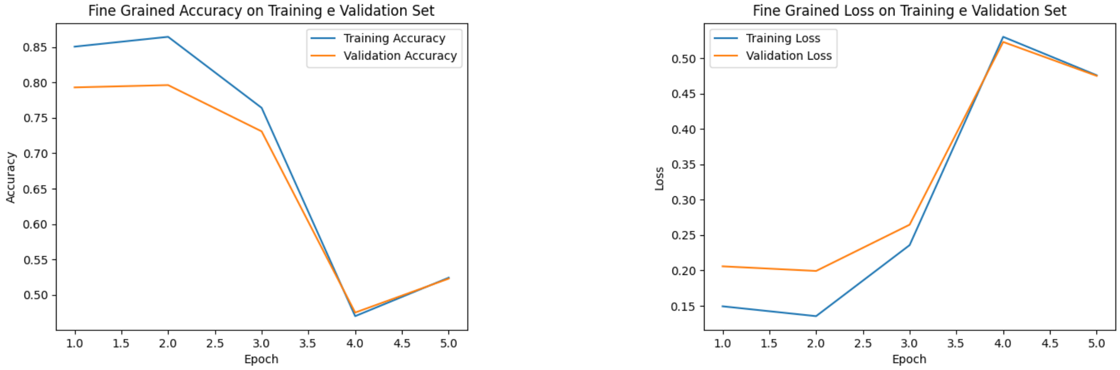
Figure 3



(a) Glossary Coarse-Grained Accuracy on Training and Validation set

(b) Glossary Coarse-Grained Loss on Training and Validation set

Figure 4



(a) Glossary Fine-Grained Accuracy on Training and Validation set

(b) Glossary Fine-Grained Loss on Training and Validation set

Figure 5