# NLP homework 2

**Amine Ahardane: 2050689**

## Abstract

Word Sense Disambiguation (WSD) is a task in natural language processing (NLP) that aims to determine the correct sense of a word within a given context taken form its sentence. This paper presents an approach to coarse-grained WSD problem and fine-grained WSD problem.

The approach used for coarse-grained WSD is a model that uses a pre-trained language model, GlossBERT and a linear classification layer. Since word senses are clustered into distinguishable homonymy clusters, coarse-grained model exploits the distant meanings of these clusters, significantly simplifying the disambiguation task.

In the fine-grained WSD scenario, the pre-trained coarse-grained model is held fixed, and a new linear classification layer is introduced. This additional layer is trained solely on the fine-grained senses, allowing the model to specialize in distinguishing subtle distinctions between closely related meanings. This approach improve the prediction of the specific meaning of the word since we have also information about its coarse-grained cluster.

## 1 Introduction

In Natural Language Processing (NLP),Word Sense Disambiguation (WSD) continues to challenge researchers due to its critical role in elevating the accuracy of language understanding systems. WSD is the task of finding the intended meaning of a word within a given context, a task that is far from trivial since the language often rapresents inherent ambiguities. This paper deal with WSD, particularly in the contexts of coarse-grained and fine-grained scenarios, where Transformer-based models, such as GlossBERT, come to the rescue since is able to address the complexities of polysemous and homonym words.

Coarse-grained WSD deals with the disambiguation of a word among a set of broadly categorized homonymy-cluster that are highly distinguishable: Imagine a scenario where a word has multiple meanings, but these meanings fall under a small number of distinct clusters. In this context, the task of selecting the appropriate cluster becomes less intricate, as the choice is limited to these distinct groups. In the fine-grained WSD takes the challenge to a deeper level. Here we want to find the gold meaning among the meanings of the word that often differ only by a shade of context. This fine-grained disambiguation demands a more sophisticated approach.

To address these challenges, Transformer-based models have emerged as revolutionary frameworks in the NLP domain. These models, initially introduced by Vaswani et al. (2017), has a remarkable capability to capture contextual information from sequences of text, thanks to their self-attention mechanisms. GlossBERT, uses the Transformer architecture and during its learning process, it adds short definitions of word meanings to its knowledge. This helps GlossBERT understand words even better.

So, in this paper, we're on a mission to see how helpful GlossBERT can be in both the easier situation and the harder one.

## 2 Text Processing

Text processing remains the same for both the coarse-grained and fine-grained models. GlossBERT is equipped with its own specialized tokenizer. The only customization we introduced is the term "unk" for senses or clusters that are not found in the training data, the model will be able to know that it lacks knowledge allowing the model to handle uncertainty, improving its generalization power, making also the model more robust.

When you input a sentence into glossBERT, it first tokenizes the text into word pieces (subword tokens), adds special tokens (e.g., [CLS] at the beginning and [SEP] between sentences), and con-

verts them into token embeddings. These token embeddings are then passed through multiple layers of the transformer model.

## 3 Model Architecture Coarse Grained

The model is pretty straigthforward. There is a pretrained GlossBERT and a classification layer . The training phase is consistent for coarse grained model and fine grained model. As input we have the sentence. For each target word, we utilize the corresponding gold label to compute the loss. For the loss function, we used cross-entropy loss because we are dealing with multi-class classification.

The optimizer used is Adam that uses adaptive learning rates for each parameter in the model allowing the model while training to adjust step sizes based on how steep the learning curve is.

## 4 Model Architecture Fine Grained

The Fine Grained model consists of two main components: the previously described coarse-grained model, which remains unchanged, and an additional classification layer designed for fine-grained distinctions.

Initially, an attempt was made to construct a fine-grained model mirroring the architecture of the coarse-grained model adjusting the final layer to the number of possible meanings. And then an approach was explored where the coarse-grained model's predictions were employed for every word. The outcome of this coarse-grained prediction was then utilized so that only the meaning listed by this output is used to select the meanings in the output layer of the fine grained model, taking the meaning with the highest probability.

However, the outcomes of this approach yielded unfavorable results.

Meanwhile, using the first approach specified above yielded to a better result.

## 5 Experimental Results

### 5.1 Fine-tuning Coarse Grained model

Many attempts were made to improve the model accuracy: Adding in input to glossBERT not only the sentence but also the candidates using <sep> keyword didn't improve the accuracy. the idea was to add important information that could improve the accuracy.Next i tried to not give it as input of glossBERT but to the linear layer. It's worth noting that GlossBERT's weights remain fixed throughout the process. This approach was chosen because freezing the weights improved accuracy, probably because the dataset is too small to fine tune it with my dataset. Through the process of fine-tuning, summing the last 4 hidden states and then using it in the classification layer, would take the f1-score of the model to 94.84%, the highest level of accuracy attained during experimentation.

Hidden states represent the contextual information about each word based on its position in the sentence and its relationship with the other words around it. Taking the last 4 we focus on the information that has been processed by the deepest layers of the model, leveraging the more abstract and high-level representations the model have learned.

It is also important to note that in inference phase, the candidates play a pivotal role in influencing the selection of the output and determining the maximum probability. There exist other intriguing techniques(not implemented) that can be used in the training phase, like a technique that involves enhancing error signals specifically for outputs that do not correspond to the candidate senses.

### 5.2 Fine-tuning fine Grained model

The attempts made to improve coarse grained model, improved also the fine grained model. As mentioned in the preceding section, our initial attempt involved utilizing the coarse-grained model's output to generate potential candidates.Then taken the result, we did an set-based intersection between the candidates available in the dataset. The resulting candidates, are then used to select the output layer of fine grained model accordingly, computing the probability and taking the max. This technique gave an f1-score of 80%, worse than using only the fine grained model alone (82%).

Instead, maintaining the coarse-grained model in a fixed state and introducing an additional linear layer for fine grained classification, we improved the score to 85.08% which stands as the highest accuracy achieved in our experimentation.

### 5.3 Possible future developments

An intriguing direction lies in investigating the recursive utilization of both coarse-grained and fine-grained models. This could involve training a coarse-grained model initially, then training a fine-grained model that leverages insights from the coarse-grained model as prescribed before. This refined fine-grained model could be augmented with

an additional layer for conducting coarse-grained model classification.

From here, question emerges: how many iterations of this recursive approach would result in noticeable enhancements? Could this recursive process truly lead to improvements?

### 5.4 References

## References

Ashish Vaswani. 2017. Attention is all you need.