Word Sense Disambiguation (WSD) of Word-in-Context (WiC) data
Natural Language Processing course 2021 – Homework 3

Andrea Gasparini
Sapienza University of Rome
gasparini.1813486@studenti.uniroma1.it

1 Introduction

Word Sense Disambiguation (WSD) is a fundamental NLP task (Navigli, 2009), which aim is to identify the meaning of ambiguous words in a given context by assigning sense identifiers from a pre-defined inventory, e.g. WordNet (Miller, 1994).

As an extension of the first homework on Word-in-Context (WiC) disambiguation (Gasparini, 2021), in this report are described two different approaches that leverage both WSD and WiC, in order to investigate if tackling these two tasks together can lead to some improvement.

2 Datasets

In addition to the provided WiC data, we make use of the WSD Evaluation Framework proposed by Raganato et al. (2017)[1], selecting SemCor as the main training corpus, and SemEval-2007 as development set. We also include the corpus of annotated glosses from Bevilacqua and Navigli (2020)[2], which is then exploited for the second approach (subsection 3.2).

All the corpora are tagged with sense ids from WordNet 3.0, which is therefore used for any additional information retrieval, e.g. retrieving glosses not available in the training corpus.

3 Methodology

In this section are described the two implemented models to address the task, both of which are based on BERT (Devlin et al., 2019), a bidirectional transformer-based encoder that provides contextual representations as a result of the pre-training on huge corpora over the two tasks of Next Sentence Prediction and Language Modelling.

In both cases we always take into account the hidden states of the last BERT's layer, averaging eventual WordPiece sub-words representations.

3.1 BERT$_{frozen}$ + WordNet

Initially, the task has been formulated as a token level classification task, in which for each sentence we compute its BERT's contextualized embedding and then for each token to be disambiguated we extract its hidden state and we label it with the corresponding sense id.

All the embeddings are computed using BERT as a frozen encoder (*feature-based* approach), and then feeded into a model composed by a Linear layer with Dropout before and after it, and a ReLU activation function followed by a Linear classification head (final architecture is shown in Figure 7).

WordNet integration In this setting, the model outputs a probability distribution over all the possible sense ids of a pre-computed vocabulary V, even though the number of possible senses for each token is always much smaller than all the ones in V. In order to solve this issue and improve the performance, we extract from WordNet the actual set of possible sense candidates for each token, and we then use the model's scores to select the most probable sense only among the extracted ones that are also contained in V. The addition of lexical knowledge from WordNet also makes possible to have a straightforward way to handle out-of-vocabulary senses (i.e. when none of the candidates is in V), by selecting the most frequent/common one (MFS).

3.2 BERT$_{finetuned}$ with context-gloss pairs

Following Huang et al. (2019) we develop a second model that leverages gloss information by treating the task as a sentence pair binary classification task, where each pair is composed by a context sentence and the gloss (definition) of the target sense,

[1] http://lcl.uniroma1.it/wsdeval
[2] http://github.com/SapienzaNLP/ewiser/tree/master/res/corpora/training/orig

and it is labeled with a boolean value determining whether the gloss, and therefore the corresponding sense, is correct for the given context sentence.

The context-gloss pairs are produced by once again exploiting WordNet; for each sentence and token to be disambiguated we retrieve the possible sense candidates and concatenate the given sentence to every candidate's corresponding gloss, following a BERT suitable format: `[CLS] sentence [SEP] gloss [SEP]`.

We make use of BERT by *fine-tuning* all its parameters during training to optimize its performance on this task, adding a Linear classification head. After fine-tuning, the hidden state of the first token `[CLS]` can be considered as a good representation and it is so used in place of the target word's one (final architecture is shown in Figure 8).

Furthermore we add a weak supervision in the pairs, highlighting the target word by prefixing its lemma to the glosses, as shown in Figure 6.

Finally, at inference time, for a given sentence we output the sense id corresponding to the most probable context-gloss pair.

4 Experimental setup

In the first approach (subsection 3.1) we precompute all the BERT's embeddings only once to directly use them as input of the simpler model, so as to save a lot of computation time.

In the BERT$_{finetuned}$ approach (subsection 3.2) instead, due to the huge number of trainable parameters (108M) and to the number of samples that substantially increases after the context-gloss pairs generation, the training corpus has been reduced by randomly sampling the 15% of SemCor to fit both computational power and training time constraints. Additionally, we also experiment on a greater fraction (50%) by replacing BERT with its lighter version: DistilBERT (Sanh et al., 2019).

The final employed hyperparameters are shown in Table 3 and 4, all tuned on the development set by also applying early stopping. The random seed has been always set to 42 for reproducibility.

Evaluation Since our models always give an answer, the F1-score would be equal to the accuracy, so we only take the latter into account. We use the given WiC dev data after training as test set to evaluate the performance of the two models both on WSD and WiC. The WiC evaluation has been performed by simply comparing the WSD outputs of two WiC sentences and producing a boolean value, i.e. when the sense ids are equal, the word in context has the same meaning in both sentences.

5 Experimental results

As we can see from Figure 1, the use of WordNet leads to a boost of performance despite the loss essentially stays the same. This behaviour can be explained by the fact that we exploit lexical knowledge only at inference time while the computation of the loss is done over the whole vocabulary. Moreover, from Table 2 we observe that there is not any relevant improvement on WiC, and we believe this may also be due to the dependency on a fixed vocabulary, which frequently results in predicting the same sense id for the same target word.

When it comes to the fine-tuning approach, we can easily see how it fully exploits BERT potential; at the cost of a less feasible training phase, after just few epochs (Figure 2) we already obtain satisfactory results better or comparable to the previous ones (Table 1 and 2). More interestingly, in this case we observe a notable improvement on WiC with respect to WSD (Table 2), which means that even though the model doesn't always predict correct sense ids, it is still able to identify when two senses are the same one and when they are not (Figure 5).

Lastly, despite the greater fraction of train data, the experimentation with DistilBERT (Figure 3) did not further improve test performance (Table 2).

6 Conclusions

In this work we developed two considerably different approaches to tackle WSD and compared their performances on WiC as well. We showed the limitations of a straightforward approach to WSD and its similar performance on WiC disambiguation; then how a more complex architecture that integrate gloss knowledge yields a notable improvement on the latter task despite the considerable reduction of the training data, reaching an accuracy of **60,10%** for WSD and **68,04%** for WiC.

Possible future works include re-training the BERT$_{finetuned}$ model over the whole SemCor corpus to further improve WSD performance (Huang et al., 2019), and, more importantly, we would consequently expect even better results in WiC.

Moreover, the first approach could be improved by injecting into the network relatedness knowledge among senses and use it both during training and inference (Bevilacqua and Navigli, 2020).

References

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Andrea Gasparini. 2021. NLP 2021 – Homework 1: Word-in-Context disambiguation.

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.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

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

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: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*.

A Tables, plots and images

Model architecture	WSD Accuracy	Epoch	Batch size
BERT$_{frozen}$	36,48	14	8
BERT$_{frozen}$ + WordNet	**65,61**	14	8
Context-gloss BERT$_{finetuned}$ (15% SemCor)	60,66	2	8
Context-gloss DistilBERT$_{finetuned}$ (50% SemCor)	60,00	5	16

Table 1: Validation performances computed on SemEval-2007, where "epoch" and "batch size" refers to the best performing checkpoint (at testing time). The other hyperparameters are in Table 3 and 4.

Model architecture	WSD Accuracy	WiC Accuracy	Epoch	Batch size
Word-level MLP (HW1) (Gasparini, 2021)	-	67,70	7	32
BERT$_{frozen}$ + WordNet	58,10	61,77	14	8
Context-gloss BERT$_{finetuned}$ (15% SemCor)	**60,10**	**68,04**	2	8
Context-gloss DistilBERT$_{finetuned}$ (50% SemCor)	58,38	68,04	5	16

Table 2: Test performances computed on the WiC dev data (`dev.jsonl` + `dev_wsd.txt`), where "epoch" and "batch size" refers to the best performing checkpoint. The other hyperparameters are in Table 3 and 4.

Hyperparameter	Value
Max epoch	100
Early stopping patience	5
Random seed	42
Input size	768
Hidden size	100
Vocabulary size (num. classes)	34.074
Dropout probability	0,2
Learning rate	1e-3
Optimizer	Adam
Loss function	cross-entropy
BERT model	bert-base-cased
BERT layer pooling	last
BERT WordPiece pooling	mean
BERT fine-tuning	false

Table 3: BERT$_{frozen}$ + WordNet approach (subsection 3.1) final hyperparameters. The vocabulary size results from the senses contained in all the corpora together (SemCor + "ALL" WSD evaluation sets + WiC dev data).

Hyperparameter	Value
Max epoch	10
Early stopping patience	5
Random seed	42
Learning rate	2e-5
Optimizer	Adam
Loss function	binary cross-entropy
BERT model	bert-base-cased distilbert-base-cased
BERT layer pooling	last
BERT WordPiece pooling	mean
BERT fine-tuning	true

Table 4: BERT$_{finetuned}$ with context-gloss pairs approach (subsection 3.2) final hyperparameters. "bert-base-cased" and "distilbert-base-cased" have been separately employed to respectively train the model with 15% of SemCor and the one with its 50%.

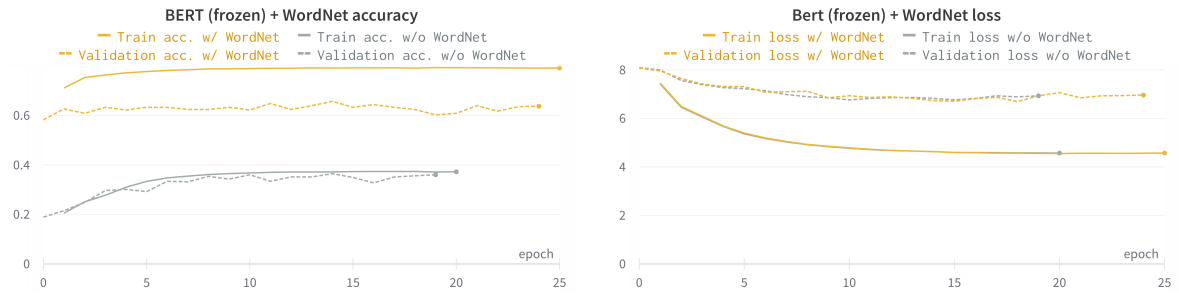

Figure 1: BERT$_{frozen}$ + WordNet approach (subsection 3.1) performance and loss histories, comparing them with (w/) and without (w/o) the integration of WordNet. The scores are computed on both the training (SemCor) and the validation (SemEval-2007) sets.

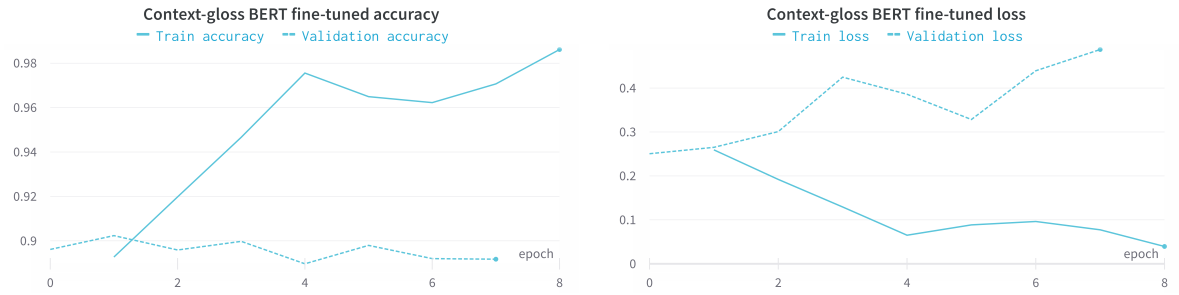

Figure 2: BERT$_{finetuned}$ (15% SemCor) with context-gloss pairs (subsection 3.2) performance and loss histories. The scores are computed on both the training (15% SemCor) and the validation (SemEval-2007) sets. Note that here the metrics refer to the binary classification of the context-gloss pairs performed to fine-tune BERT. The WSD performance at the best epoch are reported in Table 1 and 2.

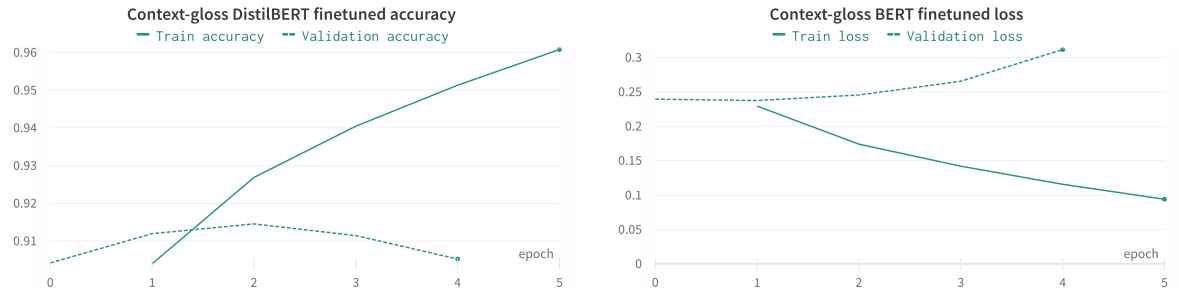

Figure 3: DistilBERT$_{finetuned}$ (50% SemCor) with context-gloss pairs (subsection 3.2) performance and loss histories. The scores are computed on both the training (50% SemCor) and the validation (SemEval-2007) sets. Note that here the metrics refer to the binary classification of the context-gloss pairs performed to fine-tune DistilBERT. The WSD performance at the best epoch are reported in Table 1 and 2.

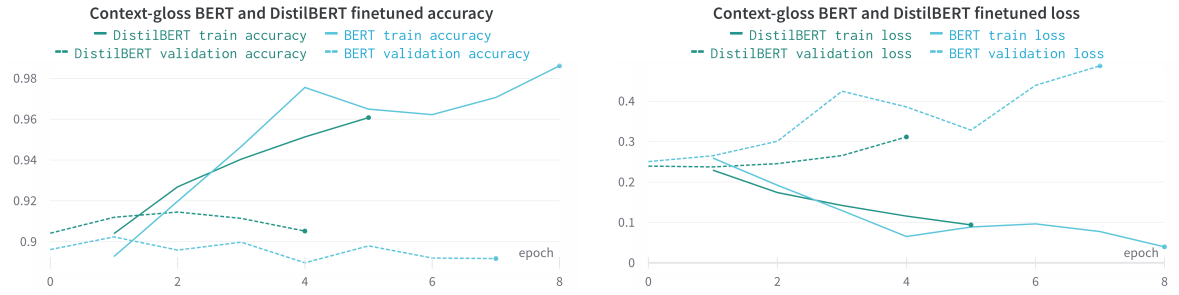

Figure 4: Performance and loss histories comparison of BERT$_{finetuned}$ with 15% SemCor (Figure 2) and DistilBERT$_{finetuned}$ with 50% SemCor (Figure 3).

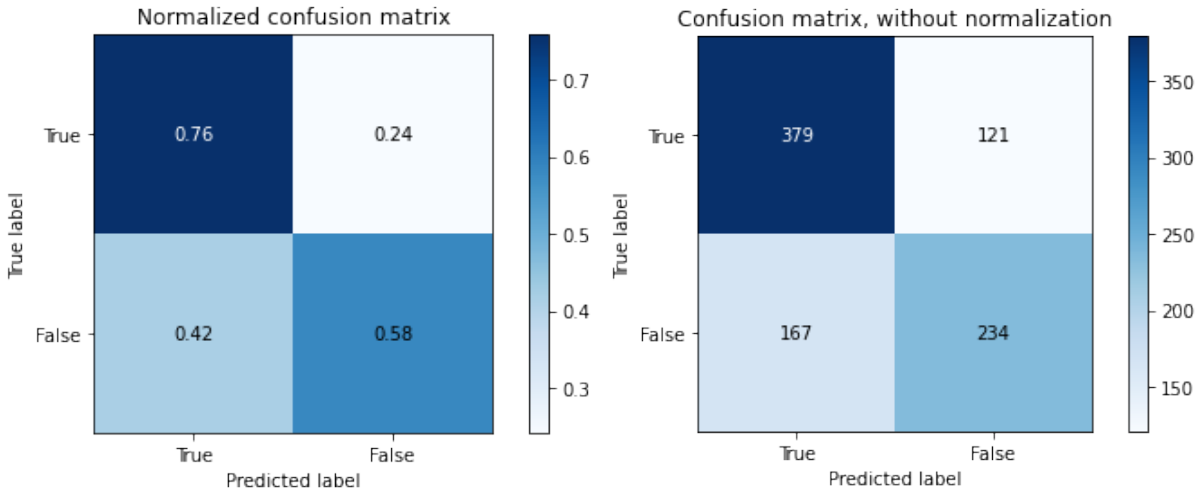

Figure 5: Normalized and non-normalized confusion matrices of the "context-gloss BERT$_{finetuned}$ (15% SemCor)" predictions on the WiC disambiguation task of the given dev data (`dev.jsonl` + `dev_wsd.txt`).

Sentence with 2 target words to be disambiguated:

We have made no such statement.

Context-gloss pairs of the target word "statement" (with weak supervision on the gloss)	Label	Sense id
[CLS] We have made no such statement [SEP] statement: a message that is stated or ... [SEP]	No	statement%1:10:00::
[CLS] We have made no such statement [SEP] statement: a fact or assertion offered as ... [SEP]	No	statement%1:10:02::
[CLS] We have made no such statement [SEP] statement: (music) the presentation of a ... [SEP]	No	statement%1:10:04::
[CLS] We have made no such statement [SEP] statement: the act of affirming or asserting or ... [SEP]	Yes	statement%1:10:06::
...		

Context-gloss pairs of the target word "made" (with weak supervision on the gloss)	Label	Sense id
[CLS] We have made no such statement [SEP] make: engage in [SEP]	No	make%2:41:00::
[CLS] We have made no such statement [SEP] make: give certain properties to something [SEP]	No	make%2:30:00::
[CLS] We have made no such statement [SEP] make: create or manufacture a man-made ... [SEP]	No	make%2:36:01::
[CLS] We have made no such statement [SEP] make: perform or carry out [SEP]	Yes	make%2:36:12::
...		

Figure 6: Example of the context-gloss pairs generation (subsection 3.2) given a sentence with target words to be disambiguated. The number of pairs is limited at 4 for brevity ("make" and "statement" respectively have 51 and 7 possible senses). The sentence is taken from SemEval-2007.

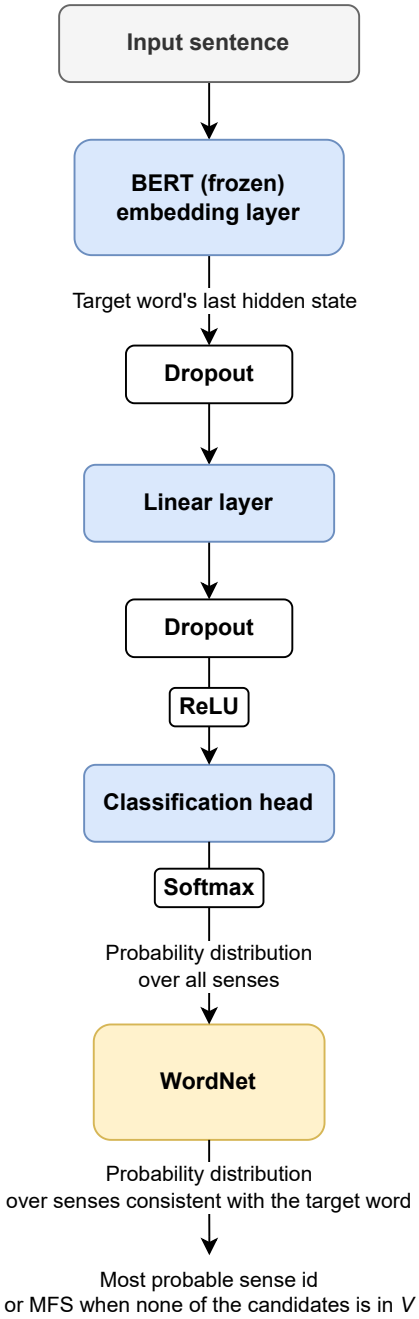

Figure 7: Architecture of the BERT$_{frozen}$ + WordNet approach (subsection 3.1). The BERT contextualized embeddings are pre-computed and directly used as input of the model. The WordNet integration is only employed at inference time and does not contribute to updating the network's weights. When none of the candidates (consistent senses) extracted by WordNet are contained in the vocabulary V, the output of the model is the MFS (most frequent/common sense).

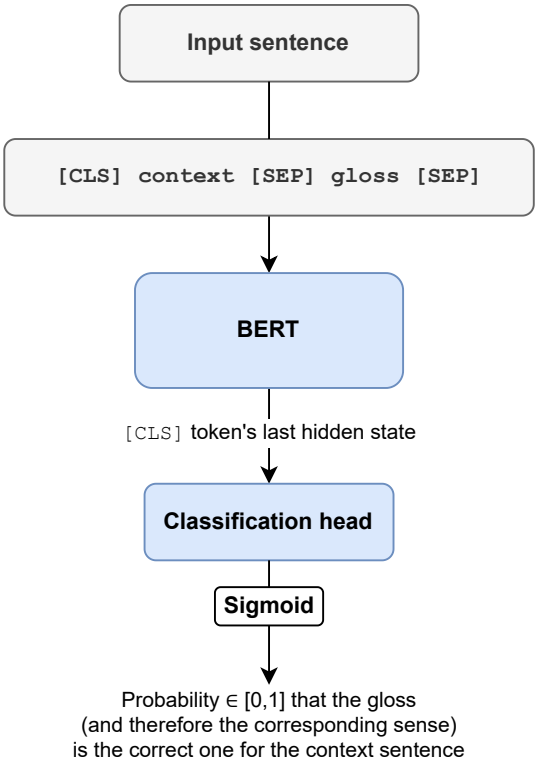

Figure 8: Architecture of the BERT$_{finetuned}$ with context-gloss pairs approach (subsection 3.2). At inference time, for a given input sentence, all the possible context-gloss pairs are generated and feeded into the model, to finally output the sense id corresponding to the gloss of the most probable pair.