Word Sense Disambiguation of Word-in-Context Data

Michela Proietti - 1739846
Sapienza University of Rome
proietti.1739846@studenti.uniroma1.it

1 Introduction

Word-in-context (WiC) disambiguation is a very common task in natural language processing (NLP), which aims at identifying the meaning of a word in a given context. More specifically, our purpose is to determine if two words in two different contexts are used with the same meaning. In this paper, we address this problem in an indirect way, by solving a different task, namely Word Sense Disambiguation (WSD). The latter consists in associating to a specific target word in a sentence its most suitable entry in a pre-defined sense inventory. In order to do this, we use BERT pre-trained model that is fine-tuned using additional linear layers.

2 Pre-Trained Contextualized Word Representation

Contextualized word embeddings can provide different representations for the same word used in different contexts. Christian Hadiwinoto and Gan (2019) showed how pre-trained contextualized word embeddings could outperform previous state-of-the-art approaches, obtaining much higher accuracies on multiple WSD datasets. In our approach, we use BERT contextualized embeddings (Jacob Devlin and Toutanova, 2019). BERT has been released in two sizes, namely *bert-base* and *bert-large*. However, due to the limited available resources, we focused on doing experiments with the former one, because the training stopped too early using *bert-large*, leading to worse performances. In particular, we have used *bert-base-uncased* model, which is made up by 12 layers with 12 self-attention heads with a hidden dimension of 768 and 110M parameters (see figure 1).

3 Preprocessing

First of all, as suggested by Yap et al. (2020), we have inserted a special $[TGT]$ token right before and after the target word to be disambiguated, in order to make it stand out from the context. Moreover, as previously suggested also by Luyao Huang and Huang (2019), we have created *context-gloss* pairs by retrieving through WordNet (Miller, 1995) all the possible senses that the target word can take, and for each of them the corresponding gloss sentence. Then, in order to provide more information to the WSD system, we have done further experiments using also the examples provided by WordNet, and this led to an improvement in the results in both WSD and WiC tasks. In order to obtain the inputs to be given to BERT, we have used *BertTokenizer*, which splits the text into tokens, adding the $[CLS]$ and $[SEP]$ tokens, and returns the *input_ids*, the *attention_mask*, and the *token_type_ids* for each *context-gloss* pair. In particular, each correctly tokenized *context-gloss* pair will have the following asset:

$$[CLS] + left\_context\_tokens + [TGT] +$$
$$target\_token + [TGT] + right\_context\_tokens$$
$$+[SEP] + gloss\_tokens + [SEP]$$
$$+examples\_tokens + [SEP]$$

Because of the limited available resources, we have pre-computed BERT inputs and stored them to decrease the amount of computations to be done at training time, as well as the amount of memory. In fact, without this simple trick the training stopped at 1% of the first epoch, and this led to very bad performances. After the preprocessing step, for each input sentence containing a word to be disambiguated, we will have as many samples as the number of possible senses of the word itself. Therefore, each batch will be constituted by sub-batches of variable length in which just the sample corresponding to the actual meaning of the ambiguous word has label 1, while all the others have label 0.

4 Methodology

In order to train and evaluate our WSD model, we have used the evaluation framework proposed by Alessandro Raganato and Navigli (2017). In particular, we have used SemCor as training set, SemEval2015 as validation set, and the concatenation of SemEval and Senseval datasets for testing. Our WSD system is made up by pre-trained BERT, which is fine-tuned thanks to the addition of three linear layers with respectively 768, 256, and 128 input features. The last linear layer has just one neuron in the output layer, because it will have to predict the most probable sense for each context-gloss pair. Much fine-tuning has been done on important hyperparameters, such as the learning rate or the dropout probability. This led to the addition of a dropout layer right after BERT, and other dropout layers after each of the linear layers, exception made for the output one. As optimizer, we have chosen *AdamW* (Loshchilov and Hutter, 2017), which decouples the weight decay from the optimization steps done with respect to the loss function. Moreover, as a common practice in the literature, we did not apply any weight decay to some parameters, namely bias, beta and gamma, and this allowed us to speed up the training, managing to get to 10% of the first epoch, thus reaching better results. The batch size has been set to 4, otherwise the training was unable to start due to the lack of memory. Finally, the number of epochs has been set to 4, even if the training actually stops before the end of the first epoch, because with better resources it could be possible to reach even higher performances. For the same reason, early stopping with a patience of 2 has been implemented to prevent overfitting. Table 2 contains information about the structure of our final model and the chosen values for all the hyperparameters. After obtaining a sufficiently high performance on the WSD task, we have addressed the WiC task by simply retrieving the WSD predictions for both input sentences using our pre-trained model, and comparing them to see if the predicted senses were the same or not.

5 Experiments

First, we have done several experiments in which we trained our WSD model using just gloss sentences, without adding the examples. The model's architecture presented in section 4 has been obtained by starting with a smaller network with just one linear layer, and increasing its size in order to have a system which was powerful enough to give better predictions. The different results obtained changing the structure of the network and the value of several hyperparameters are shown in table 3. Then, we tried to use the additional information provided by WordNet examples, and this led to an immediate improvement in the network performance. In fact, using the configuration that had given the highest accuracy using glosses, we managed to have an improvement of almost 1.8% on the WiC task, despite having just a 0.6% improvement on the WSD task. Knowing that using examples allowed us to improve our WSD system, we have done further changes to the network and hyperparameter tuning, thus getting the results shown in table 4. As can be easily noticed, the model that gave the best results on the WSD task is the one with the structure reported in table 2. The best WiC accuracy is actually obtained with a different configuration, and this means that the corresponding WSD model predicts the same incorrect meaning for words that are used with the same sense in different contexts, while it predicts different senses when they are used with different meanings. Figure 4 shows an example of the output of our WSD model, which provides a score for each possible sense key of the word to disambiguate, and predicts the one with the highest value. For each WiC pair, the two predicted sense keys are simply compared to see if they are the same or not.

6 Comparisons with previous work

In our previous work, we addressed the WiC disambiguation task in a direct way, obtaining the best results using GloVe pre-trained embeddings and 2 additional layers to perform the classification. In this way, we managed to obtain 67.9% of accuracy on the WiC test set. Using this indirect approach, instead, we are able to reach much higher performances, with more than a 10% improvement, as shown in table 5.

7 Conclusions

We showed that by creating an accurate WSD system, we can address WiC disambiguation without the need of training a different network or using WiC annotated data, but simply exploiting the huge available WSD evaluation framework. Moreover, the additional information provided by WordNet examples has turned out to be crucial to significantly improve our performances.

References

Jose Camacho-Collados Alessandro Raganato and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 99—-110.

Hwee Tou Ng Christian Hadiwinoto and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Jiaju Du, Fanchao Qi, and Maosong Sun. 2019. Using BERT for word sense disambiguation. *CoRR*, abs/1909.08358.

Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171—4186.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *CoRR*, abs/1711.05101.

Xipeng Qiu Luyao Huang, Chi Sun and Xuanjing Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge. *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.

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

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69.

Boon Peng Yap, Andrew Koh, and Eng Siong Chng. 2020. Adapting BERT for word sense disambiguation with gloss selection objective and example sentences. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 41–46, Online. Association for Computational Linguistics.

A Figures and Tables

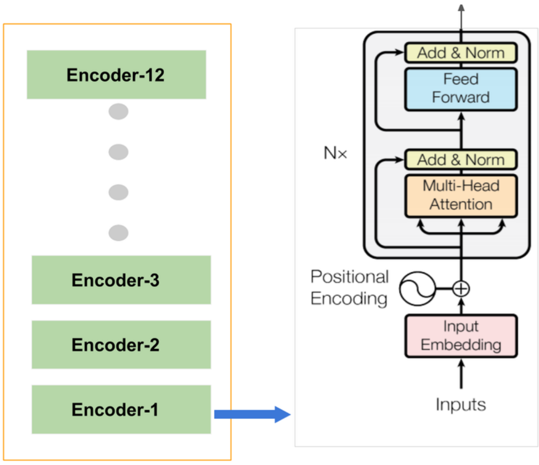


Figure 1: BERT's architecture. It is a multilayered bidirectional transformer encoder, and this figure shows the bert-base version, with 12 encoder layers, 12 attention heads and 768 hidden sized representations.

Hyperparameter	**Value**
Batch size	4
Dropout BERT	0.1
Dropout linear layers	0.4
Number of linear layers	3
Input features	768, 256, 128
Optimizer	AdamW
Learning rate	2e-5
Number of epochs	4
Early stopping patience	2

Table 1: Values of the hyperparameters used in the final model.

Hyperparameter	**Value**
Batch size	16
Activation function	PReLU
Dropout	0.2
Number of layers	2
Hidden units	64
Optimizer	Adam
Learning rate	0.0001
Beta 1	0.85
Beta 2	0.999 (default)
Early stopping patience	5

Table 2: Values of the hyperparameters used in the final model in the case of the direct approach presented in the previous paper.

N° linear layers	Dropout	Learning Rate	WSD	WSD of WiC	WiC
1 (768)	-	2e-5	0.6806	0.6321	0.7481
2 (768, 64)	-	2e-5	0.6851	0.6299	0.7647
2 (768, 128)	-	2e-5	0.6843	0.6398	0.7658
2 (768, 256)	-	2e-5	0.6862	0.6238	0.7680
2 (768, 128)*	0.2	2e-5	0.6875	0.6443	**0.7769**
2 (768, 128)	0.3	2e-5	0.6777	0.6371	0.7336
2 (768, 128)**	0.4	2e-5	**0.6949**	**0.6609**	0.7603
2 (768, 128)	0.5	2e-5	0.6880	0.6260	0.7469
2 (768, 128)	0.4	1e-5	0.6854	0.6421	0.7736

Table 3: This table shows the different results obtained using only the glosses, by changing the configuration of the network and the value of the learning rate and the dropout probability. In particular, we report the accuracy values obtained respectively on the SemEval/Senseval datasets (WSD) and on the WiC dataset for both the WSD (WSD of WiC) and WiC tasks (WiC). The best results for the WSD task were obtained by model **, while we obtained the highest accuracy for the WiC disambiguation task with a different asset (*). This means that the WSD system predicts the same wrong sense key for words used with the same meaning in different contexts, while it predicts different sense keys if they are used with different meanings.

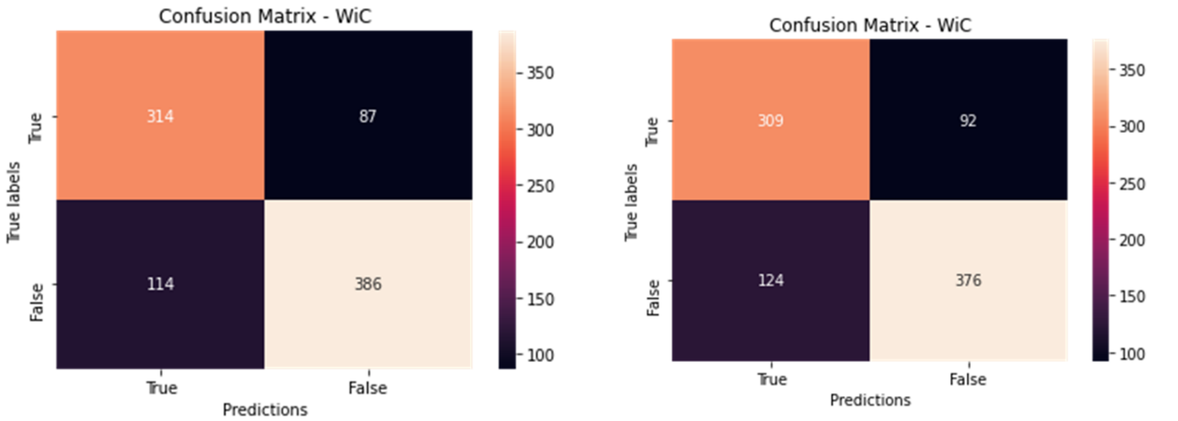


Figure 2: The two confusion matrices refer to the performances for the WiC disambiguation task obtained respectively with models * and ** in table 3. The fact that the accuracy in the WiC disambiguation task is higher for model *, even if the WSD performance is poorer, means that it predicts the same incorrect sense key for words used with the same meaning, which is reflected in a higher number of true positive in the confusion matrix on the left, while it outputs different sense keys if the words are used with different meanings, leading also to a higher number of true negative.

N° linear layers	Dropout	Learning Rate	WSD	WSD of WiC	WiC
2 (768, 128)	0.4	2e-5	0.7066	0.6615	0.7780
2 (768, 128)	0.4	3e-5	0.6153	0.6387	0.7714
2 (768, 256)	0.5	2e-5	0.7090	0.6565	0.7769
2 (768, 256)	0.4	2e-5	**0.7189**	0.6576	0.7880
2 (768, 256)*	0.3	2e-5	0.7117	0.6559	**0.8002**
2 (768, 256)	0.4	1e-5	0.7082	0.6593	0.7902
3 (768, 128, 64)	0.4	1e-5	0.7130	0.6426	0.7825
3 (768, 256, 128)**	0.4	2e-5	**0.7189**	**0.6709**	0.7869

Table 4: This table shows the different results obtained using both glosses and examples, by doing further changes to the network's architecture and some hyperparameter tuning. In particular, we report the accuracy values obtained respectively on the SemEval/Senseval datasets (WSD) and on the WiC dataset for both the WSD (WSD of WiC) and WiC tasks (WiC). The best results for WSD of WiC data were obtained with the model having the structure presented in section 4, setting the hyperparameters to the values reported in table 3. Again, there is another model (*) that gave better results on the WiC disambiguation task, but the poor accuracy in the WSD task on the same dataset shows that this high value is related to the way in which the network makes mistakes, so this model is less reliable than the chosen one.

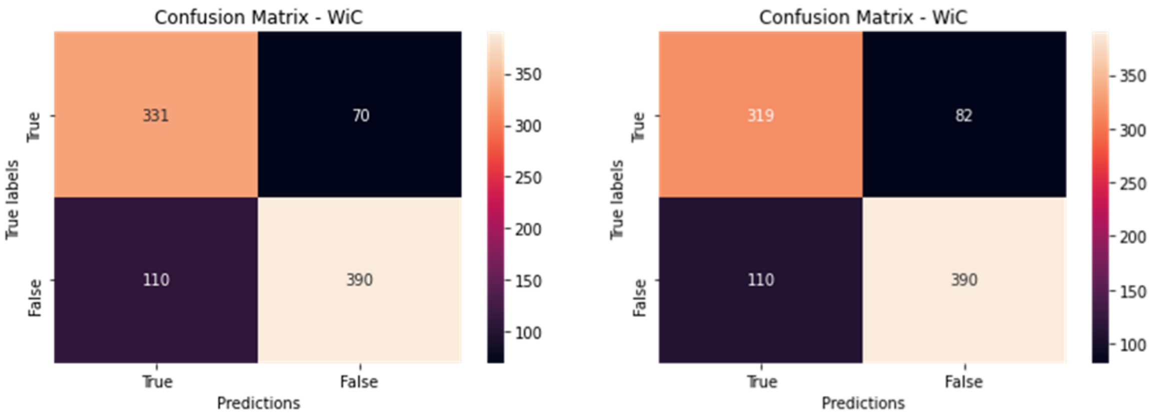
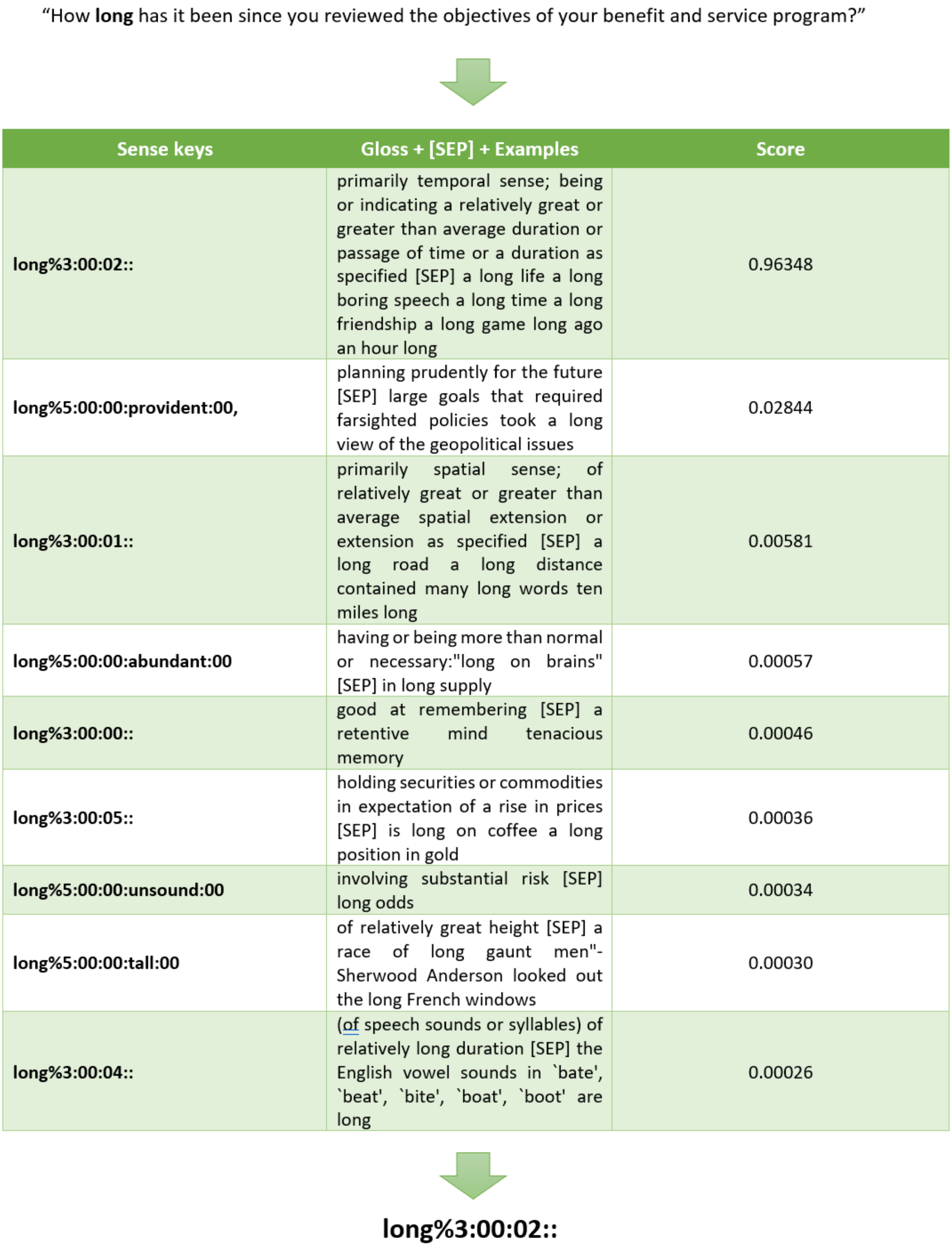


Figure 3: The two confusion matrices refer to the performances for the WiC disambiguation task obtained respectively with models * and ** in table 3. The fact that the accuracy in the WiC disambiguation task is higher for model *, even if the WSD performance is poorer, means that it predicts the same incorrect sense key for words used with the same meaning, which is reflected in a higher number of true positive in the confusion matrix on the left, while it outputs different sense keys if the words are used with different meanings. In this case, the number of true negative is the same for both models.

Metric	Direct approach	Indirect approach
Precision	0.6829	0.7861
Recall	0.6790	0.7890
F1 score	0.6773	0.7867
Accuracy	0.6790	0.7880
WSD Accuracy	-	0.6781

Table 5: Performance obtained on the WiC test set using the direct and indirect approaches.



“How **long** has it been since you reviewed the objectives of your benefit and service program?”

Sense keys	Gloss + [SEP] + Examples	Score
long%3:00:02::	primarily temporal sense; being or indicating a relatively great or greater than average duration or passage of time or a duration as specified [SEP] a long life a long boring speech a long time a long friendship a long game long ago an hour long	0.96348
long%5:00:00:provident:00,	planning prudently for the future [SEP] large goals that required farsighted policies took a long view of the geopolitical issues	0.02844
long%3:00:01::	primarily spatial sense; of relatively great or greater than average spatial extension or extension as specified [SEP] a long road a long distance contained many long words ten miles long	0.00581
long%5:00:00:abundant:00	having or being more than normal or necessary:"long on brains" [SEP] in long supply	0.00057
long%3:00:00::	good at remembering [SEP] a retentive mind tenacious memory	0.00046
long%3:00:05::	holding securities or commodities in expectation of a rise in prices [SEP] is long on coffee a long position in gold	0.00036
long%5:00:00:unsound:00	involving substantial risk [SEP] long odds	0.00034
long%5:00:00:tall:00	of relatively great height [SEP] a race of long gaunt men"-Sherwood Anderson looked out the long French windows	0.00030
long%3:00:04::	(of speech sounds or syllables) of relatively long duration [SEP] the English vowel sounds in `bate', `beat', `bite', `boat', `boot' are long	0.00026

long%3:00:02::

Figure 4: This image shows an example of the output of our best WSD model. We get a score for each of the sense keys corresponding to one of the possible meanings of the target word, and our model will output the sense key with the highest score.