Word Sense Disambiguation with Transformer Models

Pierre-Yves Vandenbussche
Elsevier Labs
Radarweg 29,
Amsterdam 1043 NX,
Netherlands
p.vandenbussche@elsevier.com

Tony Scerri
Elsevier Labs
1 Appold Street,
London EC2A 2UT, UK
a.scerri@elsevier.com

Ron Daniel Jr.
Elsevier Labs
230 Park Avenue,
New York, NY, 10169,
USA
r.daniel@elsevier.com

Abstract

In this paper, we tackle the task of Word Sense Disambiguation (WSD). We present our system submitted to the Word-in-Context Target Sense Verification challenge, part of the SemDeep workshop at IJCAI 2020 (Breit et al., 2020). That challenge asks participants to predict if a specific mention of a word in a text matches a pre-defined sense. Our approach uses pre-trained transformer models such as BERT that are fine-tuned on the task using different architecture strategies. Our model achieves the best accuracy and precision on Subtask 1 – make use of definitions for deciding whether the target word in context corresponds to the given sense or not. We believe the strategies we explored in the context of this challenge can be useful to other Natural Language Processing tasks.

1 Introduction

Word Sense Disambiguation (WSD) is a fundamental and long-standing problem in Natural Language Processing (NLP) (Navigli, 2009). It aims at clearly identifying which specific sense of a word is being used in a text. As illustrated in Table 1, in the sentence *I spent my spring holidays in Morocco.*, the word *spring* is used in the sense of *the season of growth*, and not in other senses involving coils of metal, sources of water, the act of jumping, etc.

The Word-in-Context Target Sense Verification challenge (WiC-TSV) (Breit et al., 2020) structures WSD tasks in particular ways in order to make the competition feasible. In Subtask 1, the system is provided with a sentence, also known as the *context*, the *target* word, and a definition also known as *word sense*. The system is to decide if the use of the target word matches the sense given by the definition. Note that Table 1 contains a *Hypernym* column. In Subtask 2 system is to decide if the use of the target in the context is a hyponym of the given hypernym. In Subtask 3 the system can use both the sentence and the hypernym in making the decision.

The dataset provided with the WiC-TSV challenge has relatively few sense annotated examples ($< 4,000$) and with a single target sense per word. This makes pre-trained Transformer models well suited for the task since the small amount of data would limit the learning ability of a typical supervised model trained from scratch.

Thanks to the recent advances made in language models such as BERT (Devlin et al., 2018) or XL-Net (Yang et al., 2019) trained on large corpora, neural language models have established the state-of-the-art in many NLP tasks. Their ability to capture context-sensitive semantic information from text would seem to make them particularly well suited for this challenge. In this paper, we explore different fine-tuning architecture strategies to answer the challenge. Beyond the results of our system, our main contribution comes from the intuition and implementation around this set of strategies that can be applied to other NLP tasks.

2 Data Analysis

The Word-in-Context Target Sense Verification dataset consists of more than 3800 rows. As shown in Table 1, each row contains a target word, a context sentence containing the target, and both hypernym(s) and a definition giving a sense of the term. There are both positive and negative examples, the dataset provides a label to distinguish them.

Table 2 shows some statistics about the training, dev, and test splits within the dataset. Note the substantial differences between the test set and the training and dev sets. The longer length of the context sentences and definitions in the test set may have an impact on a model trained solely on the given training and dev sets. This is a known

Target Word	Pos.	Sentence	Hypernyms	Definition	Label
spring	3	I spent my spring holidays in Morocco .	season, time_of_year	the season of growth	T
integrity	1	the integrity of the nervous system is required for normal developments	honesty, honestness	moral soundness	F

Table 1: Examples of training data.

issue whose roots are explained in the dataset author's paper (Breit et al., 2020). The training and development sets come from WordNet and Wiktionary while the test set incorporates both general purpose sources WordNet and Wiktionary, and domain-specific examples from Cocktails, Medical Subjects and Computer Science. The difference in the distributions of the test set from the training and dev sets, the short length of the definitions and hypernyms, and the relatively small number of examples all combine to provide a good challenge for the language models.

3 System Description and Related Work

Word Sense Disambiguation is a long-standing task in NLP because of its difficulty and subtlety. One way the WiC-TSV challenge has simplified the problem is by reducing it to a binary yes/no decision over a single sense for a single pre-identified target. This is in contrast to most prior work that provides a pre-defined sense inventory, typically WordNet, and requires the system to both identify the terms and find the best matching sense from the inventory. WordNet provides extremely fine-grained senses which have been shown to be difficult for humans to accurately select (Hovy et al., 2006). Coupled with this is the task of even selecting the term in the presence of multi-word expressions and negations.

Since the introduction of the transformer self-attention-based neural architecture and its ability to capture complex linguistic knowledge (Vaswani et al., 2017), their use in resolving WSD has received considerable attention (Loureiro et al., 2020). A common approach consists in fine-tuning a single pre-trained transformer model to the WSD downstream task. The pre-trained model is provided with the task-specific inputs and further trained for several epochs with the task's objective and negative examples of the objective.

Our system is inspired from the work of Huang et al. (2019) where the WSD task can be seen as a binary classification problem. The system is given the target word in context (input sentence) and one

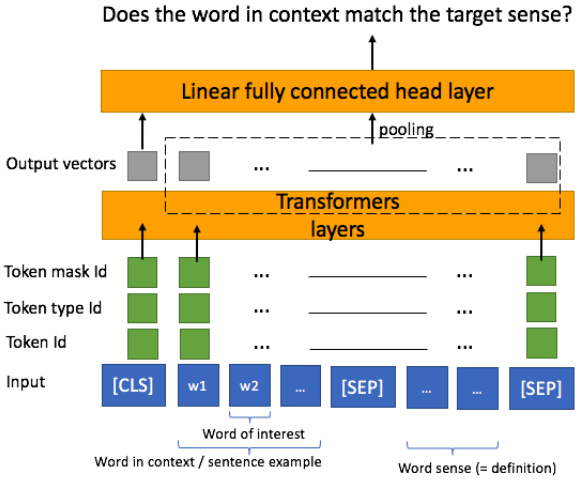

Figure 1: System overview

sense of the word separated by a special token (*[SEP]*). This configuration was originally used to predict whether two sentences follow each other in a text. But the learning power of the transformer architecture lets us learn this new task by simply changing the meaning of the fields in the input data while keeping the structure the same. We add a fully connected layer on top of the transformer model's layers with classification function to predict whether the target word in context matches the definition. This approach is particularly well suited for weak supervision and can generalise to word/sense pairs not previously seen in the training set. This overcomes the limitation of multi-class objective models, e.g. (Vial et al., 2019) that use a predefined sense inventory (as described above) and can't generalise to unseen word/sense pairs. An illustration of our system is provided in Figure 1.

4 Experiments and Results

The system described in the previous section was adapted in several ways as we tested alternatives. We first considered different transformer models, such as BERT v. XLNet. We then concentrated our efforts on one transformer model, BERT-base-uncased, and performed other experiments to improve performance.

All experiments were run five times with differ-

	Train Set	Dev Set	Test Set
Number Examples	2,137	389	1,305
Avg. Sentence Char Length	44 ± 27	44 ± 26	99 ± 87
Avg. Sentence Word Length	9 ± 5	9 ± 5	19 ± 16
Avg. Term Use	2.5 ± 2.7	1.0 ± 0.2	1.9 ± 4.4
Avg. Number Hypernyms	2.2 ± 1.5	2.2 ± 1.4	1.9 ± 1.3
Percentage of True Label	56%	51%	NA
Avg. Definition Char Length	54 ± 27	56 ± 27	157 ± 151
Avg. Definition Word Length	9.3 ± 4.7	9.6 ± 4.7	25.3 ± 23.9

Table 2: Dataset statistics. Values with ± are mean and SD

Model	Accuracy	F1
XLNet-base-cased	.522 ± .030	.666 ± .020
DistilBERT-base-uncased	.612 ± .017	.665 ± .017
RoBERTa-base	.635 ± .074	.717 ± .030
BERT-base-uncased	**.723 ± .023**	**.751 ± .023**

Table 3: Comparison of transformer models performance (lr=$5e^{-5}$; 3 epochs)

ent random seeds. We report the mean and standard deviation of the system's performance on the metrics of accuracy and F1. We believe this is more informative than a single 'best' number. All models in these experiments are trained on the training set and evaluated on the dev set.

In addition to the experiments whose results are reported here, we tried a variety of other things such as pooling methods (layers, ops), a Siamese network with shared encoders for two input sentences, and alternative loss calculations. None of them gave better results in the time available.

4.1 Alternative Transformer Models

We compared the following pre-trained transformer models from the HuggingFace transformers library: XLNet (Yang et al., 2019), BERT (Devlin et al., 2018), and derived models including RoBERTa (Liu et al., 2019) or DistilBERT (Sanh et al., 2019).

Following standard practice, those pretrained models were used as feature detectors for a fine-tuning pass using the fully-connected head layer. The results for those models are given in Table 3. The BERT-base-uncased model performed the best so it was the basis for further experiments described in the next section.

It is worth mentioning that no attempt was made to perform a hyperparameter optimization for each model. Instead, a single set of hyperparameters was used for all the models being compared.

Model	Accuracy	F1
BERT-base-uncased	.723 ± .023	.751 ± .023
+mask	.699 ± .011	.748 ± .009
+target emph	**.725 ± .013**	**.752 ± .012**
+mean-pooling	**.737 ± .017**	**.762 ± .015**
+freezing	**.734 ± .008**	**.761 ± .010**
+data augment.	**.749 ± .009**	**.752 ± .011**
+hypernyms	**.726 ± .012**	**.755 ± .011**

Table 4: Influence of strategies on model performance. We note in bold those that had a positive impact on the performance

4.2 Alternative BERT Strategies

Having selected the BERT-base-uncased pretrained transformer model, and staying with a single set of hyperparameters (learning rate = $5e^{-5}$ and 3 training epochs), there are still many different strategies that could be used to try to improve performance. The individual strategies are discussed below. The results for all the strategies are presented in Table 4

4.2.1 Masking the target word

We wondered if the context of the target word was sufficient for the model to predict whether the definition is correct. By masking the target word from the input sentence, we test the ability of the model to learn solely from the contextual words. We hoped this might improve its generalisation. Masking led to a small decrease in performance. This small delta indicates that the non-target words in the context have strong influence on the model's prediction of the correct sense.

4.2.2 Emphasising the word of interest

We wondered about the impact of taking the opposite tack and calling out the target word. As illustrated in Figure 1, some transformer models make use of *token type ids* (segment token indices) to indicate the first and second portion of the inputs.

We set the token(s) type of the target word in the input sentence to match that of the definition. Applying this strategy leads to a slight improvement in accuracy.

4.2.3 CLS vs. pooling over token sequence

The community has developed several common ways to select the input for the head binary classification layer. We compare the performance using the dedicated *[CLS]* token vector v. mean/max-pooling methods applied to the sequence hidden states of selected layers of the transformer model. Applying mean-pooling to the last layer gave the best accuracy and F1 of the configurations tested.

4.2.4 Weight Training vs. Freeze-then-Thaw

Another strategy centers on whether, and how, to update the pre-trained model parameters during fine-tuning, in addition to the training of the newly initialized fully connected head layer. Updating the pre-trained model would allow it to specialize on our downstream task but might lead to "catastrophic forgetting" where we destroy the benefit of the pre-trained model. One strategy the community has evolved (Bevilacqua and Navigli, 2020) first freezes the transformer model's parameters for several epochs while the head layer receives the updates. Later the pre-trained parameters are unfrozen and updated too. This strategy provides some improvements in accuracy and F1.

4.2.5 Data augmentation

Due to the small size of the training dataset, we experimented with data augmentation techniques while using only the data provided for the challenge. For each word in context/target sense pair, we generated:

- one positive example by replacing the target word with a random hypernym, if any exist.
- one negative example by associating the target word to a random definition.

This strategy triples the size of the training dataset. This strategy gave the greatest improvement (3.6%) of all those tested. Further work could test the effect of more negative examples.

4.2.6 Using Hypernyms (Subtask 3)

For the WiC-TSV challenge's Subtask 3 , the system can use the additional information of hypernyms of the target word. We simply concatenate the hypernyms to the definition. This strategy leads to a slight performance improvement, presumably because the hypernym indirectly emphasizes the intended sense of the target word.

Model	Acc.	Prec.	Recall	F1
Baseline (BERT)	.753	.717	**.849**	**.777**
Run1	.775	.804	.736	.769
Run2	**.778**	**.819**	.722	.768

Table 5: Model's Results on the Subtask 1 of the WiC-TSV challenge

5 Challenge Submission

The challenge allowed each participant to submit two results per task. However there was no clear winner from the strategies above; most led to a minimal improvement with a substantial standard deviation. We therefore selected our system for submitted results by a grid search over common hyperparameter values including the strategies mentioned previously. We use the train set for training and dev set to measure the performance of each model in the grid search. We chose accuracy as the main evaluation metric. For Subtask 1 we opted for the following parameters:

- Run1: BERT-base-uncased model trained for 3 epochs using the augmented dataset, with a learning rate of $7e^{-6}$ and emphasising the word of interest. Other parameters include: max sequence length of 256; train batch size of 32.
- Run2: we kept the parameters from the previous run, updating the learning rate to $1e^{-5}$.

The results on the private test set of the Subtask 1 are presented in Table 5. The Run2 of our system demonstrated a 3.3% accuracy and 14.2% precision improvements compared to the baseline.

For Subtask 3 we arrived at the following parameters:

- Run1: BERT-base-uncased model trained for 3 epochs using the original dataset, with a learning rate of $1e^{-5}$. Other parameters include: max sequence length of 256; train batch size of 32.
- Run2: we kept the parameters from the previous run, extending the number of training epochs to 5.

Model	Acc.	Prec.	Recall	F1
Baseline (BERT)	**.766**	**.741**	.828	**.782**
Run1	.694	.643	**.893**	.747
Run2	.719	.669	.885	.762

Table 6: Model's Results on the Subtask 3 of the WiC-TSV challenge

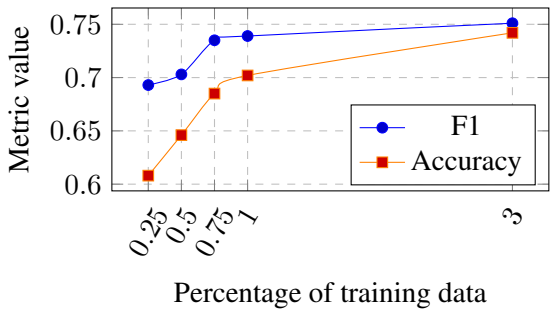

Figure 2: Influence of training data size on model performance. We used the augmented dataset to reach a proportion of 3. Parameters from Subtask 1 Run2 were used for this comparison.

The results on the private test set of the SubTask 3 are presented in Table 6. Compared to using the sentence and definition alone, our naive approach to handling hypernyms hurt performance.

6 Discussion and Future Work

We applied transformer models to tackle a Word Sense Disambiguation challenge. As in much of the current NLP research, pre-trained transformer models demonstrated a good ability to learn from few examples with high accuracy. Using different architecture modifications, and in particular the use of the token type id to flag the word of interest along with automatically augmented data, our system demonstrated the best accuracy and precision in the competition and third-best F1. There is still a noticeable gap to human performance on this dataset (85.3 acc.), but the level of effort required to create these kinds of systems is easily within reach of small groups or individuals. Despite the test set having a very different distribution than the training/development sets, our system demonstrated similar performance on both the development and test sets.

An analysis of the errors produced by our best performing model on the dev set (Subtask 1, Run2) is presented in Table 7. It shows a mix of obvious errors and more ambiguous ones where it has been difficult for the model to draw conclusions from the limited context provided by the sentence. For instance, the short sentence *it's my go* could very well correspond to the associated definition *a usually brief attempt* of the target word *go*.

As motivated by the construction of an augmented dataset, we believe that increasing the size of the training dataset would probably lead to improved performance, even without system changes. To test this hypothesis we measured the performance of our best model with increasing fractions of the training data. The results in Figure 2 show improvement as the fraction of the training dataset grows.

As a counterbalance to the positive note above, we must note that this challenge set up WSD as a binary classification problem. This is a considerable simplification from the more general sense inventory approach. Further work will be needed to obtain similar accuracy in that regime.

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.

Anna Breit, Artem Revenko, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2020. Wic-tsv: An evaluation benchmark for target sense verification of words in context. *arXiv preprint*, arXiv:2004.15016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

E. Hovy, M. Marcus, Martha Palmer, L. Ramshaw, and R. Weischedel. 2006. Ontonotes: The 90 In *HLT-NAACL*.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. Glossbert: Bert for word sense disambiguation with gloss knowledge. *arXiv preprint arXiv:1908.07245*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Daniel Loureiro, Kiamehr Rezaee, Mohammad Taher Pilehvar, and Jose Camacho-Collados. 2020. Language models and word sense disambiguation: An overview and analysis. *arXiv preprint arXiv:2008.11608*.

Target Word	Pos.	Sentence	Definition	Label	Pred
criticism	4	the senator received severe criticism from his opponent	a serious examination and judgment of something	F	T
go	3	it 's my go	a usually brief attempt	F	T
reappearance	1	the reappearance of Halley 's comet	the act of someone appearing again	F	T
continent	5	pioneers had to cross the continent on foot	one of the large landmasses of the earth	T	F
rail	4	he was concerned with rail safety	short for railway	T	F

Table 7: Examples of errors in the development set for the model used in Subtask 1 Run2.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of wordnet for neural word sense disambiguation. *arXiv preprint arXiv:1905.05677*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.