CTLR@WiC-TSV: Target Sense Verification using Marked Inputs and Pre-trained Models

Jose G. Moreno
University of Toulouse
IRIT, UMR 5505 CNRS
F-31000, Toulouse, France
jose.moreno@irit.fr

Elvys Linhares Pontes
University of La Rochelle
L3i
F-17000, La Rochelle, France
elvys.linhares_pontes@univ-lr.fr

Gaël Dias
University of Caen
GREYC, UMR 6072 CNRS
F-14000, Caen, France
gael.dias@unicaen.fr

Abstract

This paper describes the CTRL participation in the Target Sense Verification of the Words in Context challenge (WiC-TSV) at SemDeep-6. Our strategy is based on a simplistic annotation scheme of the target words to later be classified by well-known pre-trained neural models. In particular, the marker allows to include position information to help models to correctly identify the word to disambiguate. Results on the challenge show that our strategy outperforms other participants ($+11,4$ Accuracy points) and strong baselines ($+1,7$ Accuracy points).

1 Introduction

This paper describes the CTLR[1] participation at the Word in Context challenge on the Target Sense Verification (WiC-TSV) task at SemDeep-6. In this challenge, given a target word w within its context participants are asked to solve a binary task organised in three sub-tasks:

- *Sub-task 1* consists in predicting if the target word matches with a given *definition*,
- *Sub-task 2* consists in predicting if the target word matches with a given *set of hypernyms*, and
- *Sub-task 3* consists in predicting if the target word matches with a given couple *definition* and *set of hypernyms*.

Our system is based on a masked neural language model with position information for Word Sense Disambiguation (WSD). Neural language models are recent and powerful resources useful for multiple Natural Language Processing (NLP) tasks (Devlin et al., 2018). However, little effort has been made to perform tasks, where positions represent meaningful information. Regarding this line of research, Baldini Soares et al. (2019) include markers into the learning inputs for the task of relation classification and Boualili et al. (2020) into an information retrieval model. In both cases, the tokens allow the model to carefully identify the targets and to make an informed prediction. Besides these works, we are not aware of any other text-based tasks that have been tackled with this kind of information included into the models. To cover this gap, we propose to use markers to deal with target sense verification task.

The remainder of this paper presents a brief background knowledge in Section 2. Details of our strategy, including input modification and prediction mixing is presented in Section 3. Then, unofficial and official results are presented in Section 4. Finally, conclusions are drawn in Section 5.

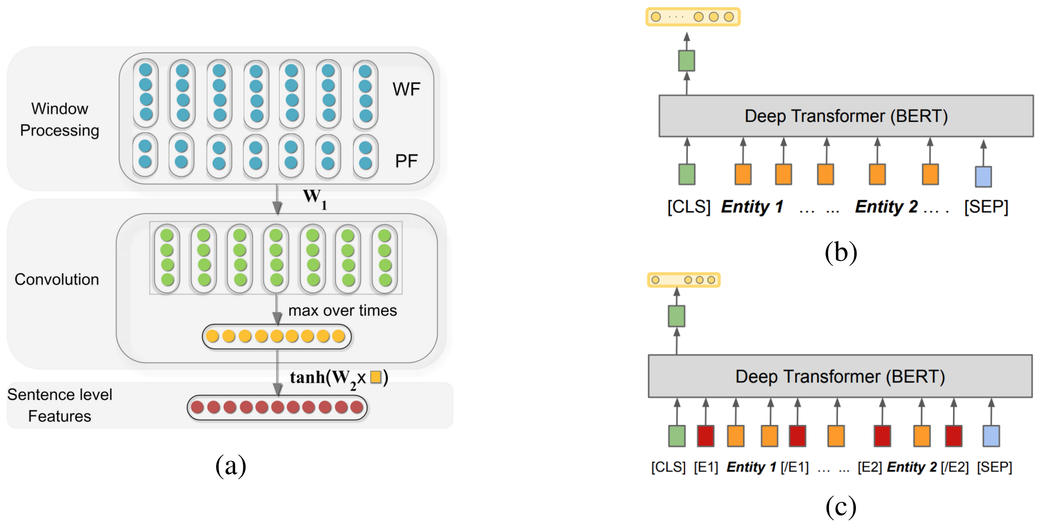
2 Background

NLP research has recently been boosted by new ways to use neural networks. Two main groups of neural networks can be distinguished[2] on NLP based on the training model and feature modification.

- First, *classical neural networks* usually use pre-trained embeddings as input and models learn their own weights during training time. Those weights are calculated directly on the target task and integration of new features or resources is intuitive. As an example, please refer to the Figure 1(a) which depicts the model from Zeng et al. (2014) for relation classification. Note that this model uses

[1] University of **C**aen Normandie, University of **T**oulouse, and University of **L**a **R**ochelle team.

[2] We are aware that our classification is arguable. Although this is not an established classification in the field, it seems important for us to make a difference between them as this work tries to introduce well-established concepts from the first group into the second one.



(a)

(b)

(c)

Figure 1: Representation examples for the relation classification problem proposed by Zeng et al. (2014) (a) and Baldini Soares et al. (2019) (b and c).

the positional features (PF in the figure) that enrich the word embeddings (WF in the figure) to better represent the target words in the sentence. In this first group, models tend to use few parameters because embeddings are not fine-tuned. This characteristic does not dramatically impact the model performances.

- The second group of models deals with *neural language models*[3] such as BERT (Devlin et al., 2018). The main difference, w.r.t. the first group, is that the weights of the models are not calculated during the training step of the target task. Instead, they are pre-calculated in an elegant but expensive fashion by using generic tasks that deal with strong initialised models. Then, these models are fine-tuned to adapt their weights to the target task.[4] Figure 1(b) depicts the model from Devlin et al. (2018) for the sentence classification based on BERT. Within the context of neural language models, adding extra features like PF demands re-train of the full model, which is highly expensive and eventually prohibitive. Similarly, re-train is needed if one opt for adding external information as made for recent works such as KNOW+E+E (Peters et al., 2019) or SenseBERT (Levine et al., 2019).

We propose an alternative to mix the best of both worlds by including extra tokens into the input in order to improve prediction without re-training it. To do so, we base our strategy on the introduction of signals to the neural language models as depicted in Figure 1(c) and done by Baldini Soares et al. (2019). Note that in this case the input is modified by introducing extra tokens ([E1], [/E1], [E2], and [/E2] are added based on target words (Baldini Soares et al., 2019)) that help the system to point out the target words. In this work, we mark the target word by modifying the sentence in order to improve performance of BERT for the task of target sense verification.

3 Target Sense Verification

3.1 Problem definition

Given a first sentence with a known target word, a second sentence with a definition, and a set of hypernyms, the target sense verification task consists in defining whether or not the target word in the first sentence corresponds to the definition or/and the set of hypernyms. Note that two sub-problems may be set if only the second sentence or the hypernyms are used. These sub-problems are presented as sub-tasks in the WiC-TSV challenge.

3.2 CTLR method

We implemented a target sense verification system as a simplified version[5] of the architecture proposed by Baldini Soares et al. (2019), namely $BERT_{EM}$. It is based on BERT (Devlin et al., 2018), where an extra layer is added to make the

[3] Some subcategories may exist.

[4] We can image a combination of both, but models that use BERT as embeddings and do not fine-tune BERT weights may be classified in the first group.

[5] We used the *EntityMarkers[CLS]* version.

classification of the sentence representation, i.e. classification is performed using as input the [CLS] token. As reported by Baldini Soares et al. (2019), an important component is the use of mark symbols to identify the entities to classify. In our case, we mark the target word in its context to let the system know where to focus on.

3.3 Pointing-out the target words

Learning the similarities between a couple of sentences (sub-task 1) can easily be addressed with BERT-based models by concatenating the two inputs one after the other one as presented in Equation 1, where S_1 and S_2 are two sentences given as inputs, $t^1_i (i = 1..n)$ are the tokens in S_1, and $t^2_j (j = 1..m)$ are the tokens in S_2. In this case, the model must learn to discriminate the correct definition and also to which of the words in S_1 the definition relates to.

$$\begin{array}{c} \mathrm{input}(\mathrm{S_1}, \mathrm{S_2}) = \\ \text{[CLS]} \quad t^1_1 \quad t^1_2 \quad ... \quad t^1_n \\ \text{[SEP]} \quad t^2_1 \quad t^2_2 \quad ... \quad t^2_m \end{array} \tag{1}$$

To avoid the extra effort by the model to evidence the target word, we propose to introduce this information into the learning input. Thus, we mark the target word in S_t by using a special token before and after the target word[6]. The input used when two sentences are compared is presented in Equation 2. S_t is the first sentence with the target word t_i, S_d is the definition sentence, and t^k_x are their respective tokens.

$$\begin{array}{c} \mathrm{input}^{\mathrm{sp1}}(\mathrm{S_t}, \mathrm{S_d}) = \\ \text{[CLS]} \quad t^t_1 \quad t^t_2 \quad ... \quad \$ \quad t^t_i \quad \$ \quad ... \quad t^t_n \\ \text{[SEP]} \quad t^d_1 \quad t^d_2 \quad ... \quad t^d_m \end{array} \tag{2}$$

In the case of hypernyms (sub-task 2), the input on the left side is kept as in Equation 2, but the right side includes the tagging of each hypernym as presented in Equation 3.

$$\begin{array}{c} \mathrm{input}^{\mathrm{sp2}}(\mathrm{S_t}, \mathrm{S_h}) = \\ \text{[CLS]} \quad t^t_1 \quad t^t_2 \quad ... \quad \$ \quad t^t_i \quad \$ \quad ... \quad t^t_n \\ \text{[SEP]} \quad s^h_1 \quad \$ \quad s^h_2 \quad \$ \quad ... \quad \$ \quad s^h_l \end{array} \tag{3}$$

3.4 Verifying the senses

We trained two separated models, one for each subproblem using the architecture defined in Section 3.2. The output predictions of both models are used to solve the two-tasks problem. So, our overall prediction for the main problem is calculated by combining both prediction scores. First, we normalise the scores by applying a $softmax$ function to each model output, and then we select the prediction with the maximum probability as shown in Equation 5.

$$pred(x) = \begin{cases} 1, & \text{if } m^{sp1}_1(x) + m^{sp2}_1(x) \\ & \quad > m^{sp1}_0(x) + m^{sp2}_0(x). \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where
$$m^{spk}_i = \frac{exp(p^{spk}_i)}{\sum_{j=\{0,1\}} exp(p^{spk}_j)} \tag{5}$$

and p^{spk}_i is the prediction value for the model k for the class i (m^{spk}_i).

4 Experiments and Results

4.1 Data Sets

The data set was manually created by the task organisers and some basic statistics are presented in Table 1. Detailed information can be found in the task description paper (Breit et al., 2020). No extra-annotated data was used for training.

	train	development	test
Positive	1206	198	-
Negative	931	191	-
Total	2137	389	1324

Table 1: WiC-TSV data set examples per class. Positive examples are identified as 'T' and negative as 'F' in the data set.

4.2 Implementation details

We implemented $BERT_{EM}$ of Baldini Soares et al. (2019) using the huggingface library (Wolf et al., 2019), and trained two models with each training set. We selected the model with best performance on the development set. Parameters were fixed as follows: 20 was used as maximum epochs, *Cross Entropy* as loss function, Adam as optimiser, *bert-base-uncased*[7] as pre-trained model, and other parameters were assigned following the library recommendations (Wolf et al., 2019). The final layer is composed of two neurons (negative or positive).

[6] We used '$' but any other special token may be used.

[7] https://github.com/google-research/bert

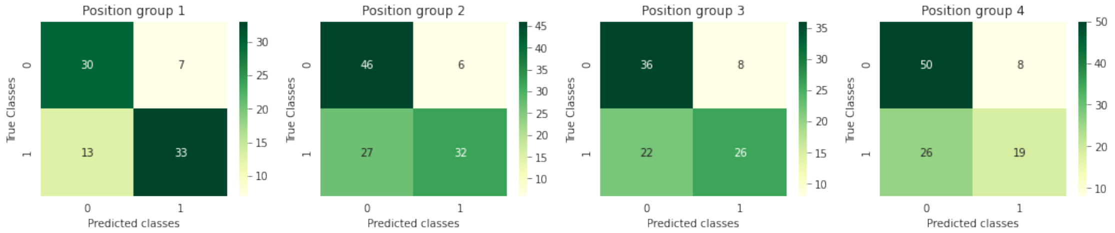


Figure 2: Confusion matrices for different position groups. Group 1 (resp. 2, 3, and 4) includes all sentences for which the target word appears in the first (resp. second, third, and fourth) quarter of the sentence.

4.3 Results

As the test labels are not publicly available, our following analysis is performed exclusively on the development set. Results on the test set were calculated by the task organisers.

We analyse confusion matrices depending on the position of the target word in the sentence as our strategy is based on marking the target word. These matrices are presented in Figure 2. The confusion matrix labelled as position group 1 shows our results when the target word is in the first 25% positions of the S_t sentence. Other matrices show the results of the remaining parts of the sentence (second, third, and fourth 25%, for respectively group 2, 3, and 4).

Confusion matrices show that the easiest cases are when the target word is located in the first 25%. Other parts are harder mainly because the system considers positive examples as negatives (high false negative rate). However, the system behaves correctly for negative examples independently of the position of the target word. To better understand this wrong classification of the positive examples, we calculated the true label distribution depending on the normalised prediction score as in Figure 3. Note that positive examples are mainly located on the right side but a bulk of them are located around the middle of the figure. It means that models m^{sp1} and m^{sp2} where in conflict and average results were slightly better for the negative class. In the development set, it seems important to correctly define a threshold strategy to better define which examples are marked as positive.

In our experiments, we implicitly used 0.5 as threshold[8] to define either the example belongs to the 'T' or 'F' class. When comparing Figures 3 and 4, we can clearly see that small changes in the threshold parameter would affect our results with a larger impact in recall than in precision. This is mainly given to the fact that our two models contradict for some examples.

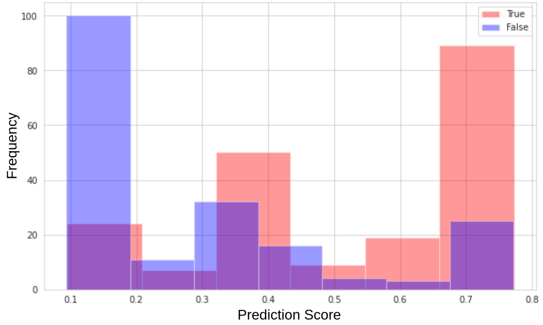


Figure 3: Histograms of predicted values in the dev set.

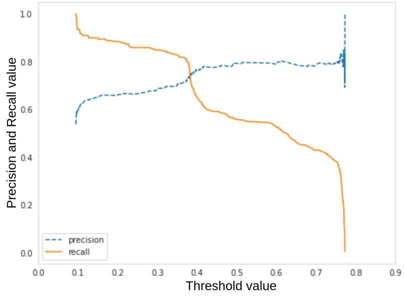


Figure 4: Precision/Recall curve for the development set for different threshold values.

We also considered the class distribution depending on a normalised distance between the target token and the beginning of the sentence. From Figure 5, we observe that both classes are less frequent at the beginning of the sentence with negative examples slightly less frequent than positive ones. It is interesting to remark that negative examples uniformly distribute after the first bin. On the contrary, the positive examples have a more unpredictable distribution indicating that a strategy based on only positions may fail. However, our strategy that combines markers to indicate the target word and a

[8] Because of the condition $m_1^{sp1}(x) + m_1^{sp2}(x) > m_0^{sp1}(x) + m_0^{sp2}(x)$.

		Global				WordNet/Wiktionary				Cocktails				Medical entities				Computer Science			
Run	User	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
run2	CTLR (ours)	**78,3**	**78,9**	78,0	**78,5**	72,1	75,8	70,7	73,2	87,5	82,4	90,3	86,2	85,9	86,7	85,8	86,3	83,3	78,4	88,5	83,1
szte	begab	66,9	61,6	92,5	73,9	70,2	66,5	89,6	76,4	55,1	48,9	96,8	65,0	65,4	60,5	95,3	74,0	70,2	61,3	97,4	75,2
szte2	begab	66,3	61,1	92,8	73,7	69,9	66,2	90,2	76,3	53,7	48,1	96,8	64,3	64,4	59,8	95,3	73,5	69,6	60,8	97,4	74,9
BERT	-	76,6	74,1	82,8	78,2	73,5	76,1	74,2	75,1	79,2	67,8	98,2	80,2	79,8	75,8	89,6	82,1	82,1	73,0	97,9	83,6
FastText	-	53,4	52,8	79,4	63,4	57,1	58,0	74,0	65,0	43,1	43,1	100,0	60,2	51,1	51,5	90,3	65,6	54,0	50,5	67,1	57,3
Baseline (true)		50,8	50,8	**100,0**	67,3	53,8	53,8	100,0	70,0	43,1	43,1	100,0	60,2	51,7	51,7	100,0	68,2	46,4	46,4	100,0	63,4
Human		85,3	80,2	96,2	87,4	82,1				92,0				89,1				86,5			

Table 2: Accuracy, Precision, Recall and F1 results of participants and baselines. Results where split by type. General results are included in column ‘Global’. All results were calculated by the task organisers (Breit et al., 2020) as participants have not access to test labels. Best performance for each global metric is marked in **bold** for automatic systems.

strong neural language model (BERT) successfully manage to classify the examples.

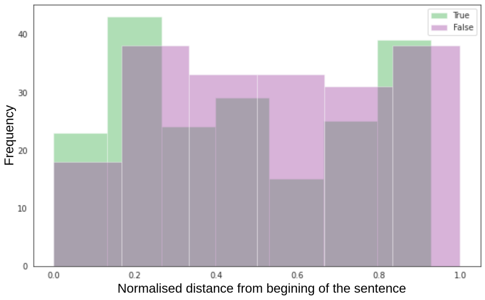


Figure 5: Position distribution based on the target token distances.

Finally, the main results calculated by the organisers are presented in Table 2. The global column presents the results for the global task, including definitions and hypernyms. Our submission is identified as run2-CTLR. In the global results, our strategy outperforms participants and baselines in terms of Accuracy, Precision, and F1. Best Recall performance is unsurprisingly obtained by the baseline (true) that corresponds to a system that predicts all examples as positives. Two strong baselines are included, FastText and BERT. Both baselines were calculated by the organisers with more details in (Breit et al., 2020). It is interesting to remark that the baseline BERT is very similar to our model but without the marked information. However, our model focuses more on improving Precision than Recall resulting with a clear improvement in terms of Accuracy but less important in terms of F1.

Organisers also provide results grouped by different types of examples. They included four types with three of them from domains that were not included in the training set[9]. From Table 2, we can also conclude that our system is able to adapt to out-of-domain topics as it is clearly shown for the *Cocktails* type in terms of F1, and also for the *Medical entities* type to a less extent. However, our system fails to provide better results than the standard BERT in terms of F1 for the *Computer Science* type. But, in terms of Accuracy, our strategy outperforms for a large margin the out-of-domain types (8.3, 6.1, and 1.2 improvements in absolute points for *Cocktails*, *Medical entities*, and *Computer Science* respectively). Surprisingly, it fails on both, F1 and Accuracy, for *WordNet/Wiktionary*.

5 Conclusion

This paper describes our participation in the WiC-TSV task. We proposed a simple but effective strategy for target sense verification. Our system is based on BERT and introduces markers around the target words to better drive the learned model. Our results are strong over an unseen collection used to verify senses. Indeed, our method (Acc=$78,3$) outperforms other participants (second best participant, Acc=$66,9$) and strong baselines (BERT, Acc=$76,6$) when compared in terms of Accuracy, the official metric. This margin is even larger when the results are compared for the out-of-domain examples of the test collection. Thus, the results suggest that the extra information provided to the BERT model through the markers clearly boost performance.

As future work, we plan to complete the evaluation of our system with the WiC dataset (Pilehvar and Camacho-Collados, 2019) as well as the integration of the model into a recent multi-lingual entity linking system (Linhares Pontes et al., 2020) by marking the anchor texts.

[9] More details in (Breit et al., 2020).

Acknowledgements

This work has been partly supported by the European Union's Horizon 2020 research and innovation programme under grant 825153 (EMBEDDIA).

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *ACL*. 2895–2905.

Lila Boualili, Jose G. Moreno, and Mohand Boughanem. 2020. MarkedBERT: Integrating Traditional IR Cues in Pre-Trained Language Models for Passage Retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20)*. Association for Computing Machinery, 1977–1980.

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:2004.15016* (2020).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805* (2018).

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2019. Sensebert: Driving some Sense into Bert. *arXiv:1908.05646* (2019).

Elvys Linhares Pontes, Jose G. Moreno, and Antoine Doucet. 2020. Linking Named Entities across Languages Using Multilingual Word Embeddings. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020 (JCDL '20)*. Association for Computing Machinery, 329–332.

Matthew E. Peters, Mark Neumann, Robert L. Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *EMNLP*. 43–54.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1267–1273.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771* (2019).

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 2335–2344.