Pheonix at SemEval-2020 Task 5: Masking the Labels Lubricates Models for Sequence Labeling

Pouria Babvey, Dario Borrelli, Yutong Zhao, Carlo Lipizzi
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, United States
pbabvey,darioborrelli,yzhao102,
clipizzi@stevens.edu

Abstract

This paper presents the deep-learning model that is submitted to the SemEval-2020 Task 5 competition: "Detecting Counterfactuals". We participated in both Subtask1 and Subtask2. The model proposed in this paper ranked 2nd in Subtask2: "Detecting antecedent and consequence". Our model approaches the task as a sequence labeling. The architecture is built on top of BERT; and a multi-head attention layer with label masking is used to benefit from the mutual information between nearby labels. Also, for prediction, a multi-stage algorithm is used in which the model finalize some predictions with higher certainty in each step and use them in the following. Our results show that masking the labels not only is an efficient regularization method but also improves the accuracy of the model compared with other alternatives like CRF. Label masking can be used as a regularization method in sequence labeling. Also, it improves the performance of the model by learning the specific patterns in the target variable.

1 Introduction

Counterfactual statements describe events that did not actually happen or cannot happen, as well as the possible consequence. For example, in the sentence *"If I was tall just like a giraffe, I was eating the leafs of those charming trees."*, the first clause is a counterfactual *antecedent* and the second clause is the *consequent*. To model counterfactual semantics and reasoning in natural language, SemEval-2020 Task 5 (Yang et al., 2020) aims to provide a benchmark for two basic problems: (1) detecting counterfactual statements as a binary classification task (2) detecting antecedent and consequent as a sequence labeling task. The introduced model in this paper is designed for the 2nd task. The model approaches the antecedent (ANT) and consequent (CONS) detection as a sequence labeling.

Sequence labeling is the assignment of a categorical label to each member of a sequence. Common examples of a sequence labeling are part-of-speech tagging and named entity recognition. Although, sequence labeling can be treated as a set of independent classification tasks, considering the dependency of choices for nearby elements improves the accuracy to choose the globally best set of labels for the whole sequence. It is specially the case for some sequence labeling tasks like SQuAD (Rajpurkar et al., 2016), that the task is to find a set of consecutive tokens from a reading passage that respond to the specific question. Likewise, in counterfactual detection task, individual tokens in the same noun phrase, clause, or sentence are likely to get the same labels.

In order to benefit from the mutual information between the labels of nearby tokens our model is built on top of BERT (Devlin et al., 2018). BERT benefits from two main features: (1) Pretraining: different forms of information about the semantic and syntactic of the language is encoded in the model (Jawahar et al., 2019), and this let fine-tuning the model on a much smaller dataset than it would be required for a model that is built from the ground up. (2) multi-head attention layers: attention mechanism was first introduced in (Bahdanau et al., 2014) to focus on the most pertinent parts of the text for language translation and found applications in a wide range of NLP tasks (He et al., 2017; Babvey et al., 2019). Then, multi-head

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 677–682
Barcelona, Spain (Online), December 12, 2020.

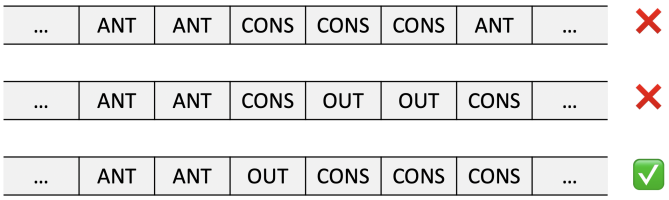

Figure 1: Consistency check for some sample outputs

attention mechanism (Vaswani et al., 2017) initiated a new epoch in NLP by allowing attention heads to focus on different aspects of the language concurrently. Some of these heads correspond well to linguistic notions of syntax, like direct objects of verbs, determiners of nouns, objects of prepositions, and coreference mentions (Clark et al., 2019; Vig, 2019).

Although BERT generates a vector representation for each token as a function of syntactic and semantic features of all tokens in the text segment (Devlin et al., 2018), using a typical linear layer with Softmax function on top of BERT may not be the best fit for some specific sequence labeling like counterfactual detection. To better understand the possible deficiencies of such models consider a simple case where a model has to predict a sequence of two elements where 'AB' and 'ab' are both plausible outcomes but 'Ab' and 'aB' outcomes are highly unlikely. Then, in case one element is clearly 'A' or 'a' and the other is uncertain, a reasonable sequence classification architecture like BERT can ensure that the information flows to the uncertain element to get aligned to the correct value. However, if both elements are very uncertain around 50% probability, then BERT will sometimes generates 'aB' or 'Ab' in the output.

To address this issue, conditional random fields (CRF) were used typically to filter out inconsistent labeling patterns. CRF is a standard model for prediction where contextual information or state of the neighbors affect the current prediction (Lafferty et al., 2001). CRF found its application in sequence labeling especially as a complement for LSTMs (Huang et al., 2015). However, the application of CRF became limited after the advent of BERT and other transformers. For some tasks adding CRF on top of BERT shows no improvement and in the original BERT paper a Softmax was used for sequence labeling. And since then, few works observed improvement using CRF on top of BERT (Souza et al., 2019). Our experiment results show very limited improvement for using CRF with BERT.

As an alternative we tried a multi-head attention layer on top of the BERT while part of the labels are revealed to the layer during training, a technique we call "label masking". Inspired by the Cloze task (Taylor, 1953) masking was used as a preliminary task for pretraining the BERT model. The mask language model randomly masks some of the tokens from the input, and then the model has to predict the masked tokens (Devlin et al., 2018). In this study, however, we mask part of the labels and then the model has to predict the remaining labels based on the text and the revealed labels. We observe a notable improvement in the results while using label masking. Our results suggest label masking as an efficient method for sequence labeling task both for regularization and accuracy improvement for sequence labeling.

The contribution in the model architecture are as follows: (1) a multi-head attention layer is added on the top of BERT. The input of the layer are the sequence of hidden-states from BERT and part of labels, then, the model has to predict the missing labels in the output. (2) for prediction, a multi-stage algorithm is used, in which during an iterative process, the model finalize some predictions with higher certainty, and in the next iteration it can use the knowledge from the finalized labels to predict the remaining labels.

The code is available at `http://github.com/pbabvey/label_masking`.

In the following, section 2 describes the model architecture and the experiment results are reported in Section 3.

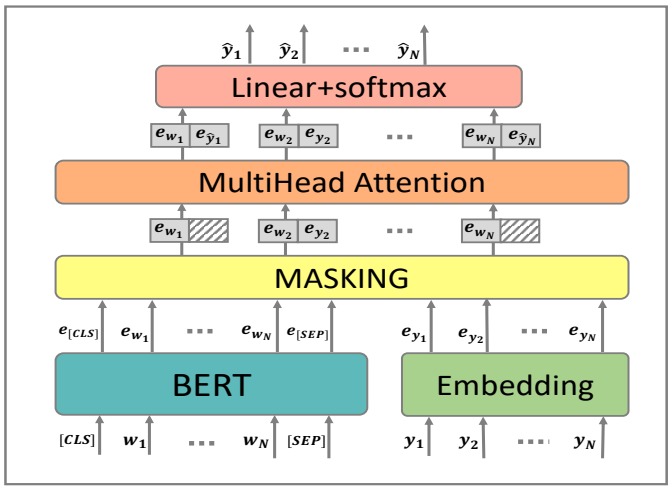

Figure 2: The architecture of the model

2 Model Description

2.1 model architecture

Our model approach the counterfactual detection task as a sequence labeling. Although a few cases exist in training set in which the antecedent and consequent are overlapping (in 0.3% of all training samples), our model is set to predict non-overlapping antecedent and consequent. Then, the task becomes a sequence labeling where each token can be labeled as ANT (as part of antecedent), or CONS (as part of consequent), and OUT (as part of none).

The model is built on top of BERT. Moreover, as the labels follow a limited set of patterns in counterfactual detection we used a new technique, called label masking, to filter out inconsistent patterns in the output. Figure 1 shows some examples of outputs with inconsistent patterns. In the first example, discontinuous antecedent is invalid, and in the second example, discontinuous consequent is invalid. To adjust such inconsistencies in the output of our model a multi-head attention layer is added on top of the architecture. During the training, the layer, gets all the token outputs from BERT and part of the labels, and has to predict the missing labels accordingly.

Figure 2 shows the model architecture. As can be seen in the figure, the masking layer receives the labels embeddings and the token outputs of BERT model (sequence of hidden-states at the output of the last layer of BERT), and then, mask a subset of labels. Then, multi-head attention layer predicts the missing labels based on the partially labeled sequence. For the multi-head attention layer the hidden_size is set to $H = 768$ and the number of self-attention heads is $A = 12$. Thus, the size of each self-attention head is $d_k = 768/12 = 64$. Although the original version of multi-head attention model (Vaswani et al., 2017) use the same $(key, value)$ pairs, in this study the token hidden states and label embeddings are used as key and values. Thus, the attention from token w_i with an unknown label to token w_j with label y_j is calculated as follows:

$$Attention(e_{w_i}, e_{w_j}, e_{y_j}) = Softmax(\frac{e_{w_i} \cdot e_{w_j}^T}{\sqrt{d_k}})e_{y_j} \quad (1)$$

With this strategy the model learns to predict an individual label not only based on the token hidden states from BERT, but also based on the predicted label for nearby tokens.

In the test step, the model use a multi-stage process to predict all the labels. The process is similar to decoding step in Transformers for language translation (Vaswani et al., 2017), in which the decoder predict the first word in the target language, and then, predicts the subsequent words one-by-one based on the previous predicted words. However, here as the length of the target sequence is fixed (the number of labels is equal to the number of tokens) we used a more efficient approach: In each round, the model finalize a subset of labels with confidence higher than a threshold, then, the model use the finalized labels

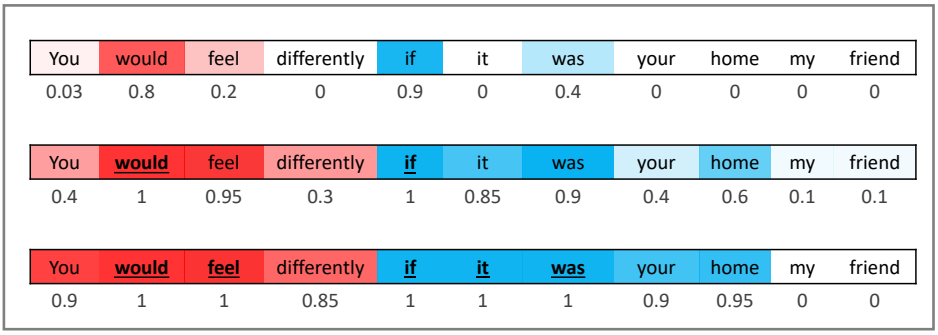

Figure 3: An example of multi-stage prediction with certainty threshold 0.8; red tokens are likely to be part of consequent while blue ones are part of antecedent. The numbers show the certainty of each individual prediction. The inderlined words are finalized at the end of the previous round

to predict the remaining labels in the following rounds. In this study we simply used a linear decremental threshold to determine the finalized labels, however non-linear alternative may fit better as inquired in the other scenarios (Xie et al., 2019). Figure 3 shows an example to demonstrate how the model predicts all the labels in some steps.

2.2 Post-processing

In the post-processing step we use a backtracking algorithm to find the antecedent and consequent spans based on the sequence of logit values in the output of the linear layer (the output of the linear layer is in the form of $(\alpha_0, \alpha_1, \alpha_2)$ in which α_0 represent the possibility that the token belongs to OUT, and α_1 and α_2 respectively show the possibility that the token is part of the ANT and CONS). While with a hard classification, the label for each token is simply $max\_index(\alpha_0, \alpha_1, \alpha_2)$, our model aggregates the knowledge from individual tokens by summing up the logit values for each possible span. We define the fitness of a span assignment $\mathcal{S}$ as follows:

$$F(\mathcal{S}) = \sum_{i \in \mathcal{A}} \alpha_{i,1} + \sum_{i \in \mathcal{C}} \alpha_{i,2} + \sum_{i \in \mathcal{O}} \alpha_{i,0} \tag{2}$$

Where $\mathcal{A}, \mathcal{C}, \mathcal{O}$ are the set of token indices with ANT, CONS, and OUT labels in the span assignment $\mathcal{S}$. Then, the backtracking algorithm finds the optimized span assignment by maximizing the fitness function $F(\mathcal{S})$ based on a sequence of logits. In each step, it maximize the fitness function for the first k tokens of the sequence based on the optimized assignments for the first $k-1$ tokens.

3 Experiment Results

The model specifications are as follows: The $BERT_{BASE}$ module is borrowed from Transformers library (Wolf et al., 2019) in PyTorch. For the multi-head attention layer the hidden_size is set to $H = 768$, and the number of self-attention heads is $A = 12$. In the masking layer, between 30% to 100% of all the tokens are selected at random for masking.

The learning rate of multi-head attention layer is set to $2 \cdot 10^{-3}$ to expedite its convergence, while the learning rate of other layers is set to $2 \cdot 10^{-5}$ to maintain the accumulated knowledge in BERT during training.

The train-validation ratio is set to 90%-10%. Table 1 compares the results for the three alternatives. As the access to the test labels was limited during the competition, the results are based on the average of 5 run for each model. As can be seen, the improvement using the multi-head attention layer is noticeable. The improvement by multi-head attention layer for the exact matches (0.53 instead of 0.49), suggest it allows the model to better find the boundaries.

	F-1	Recall	Precision	Exact match
BERT	87.05	87.8	88.9	0.49
BERT+CRF	87.09	87.9	88.9	0.5
BERT+MultiHead	87.7	87.5	91.1	0.53

Table 1: Test results for $BERT_{BASE}$ model and two other alternatives, the improvement using multi-head attention and label masking is notable

4 Conclusion

In this paper we introduced a deep-learning model to solve the counterfactual detection task. The model approaches the task as a sequence labeling. We used label masking with a multi-head attention layer on top of the BERT to exploit the mutual information between nearby labels. Moreover, we used a multi-stage procedure for label prediction in which the model learns to use the knowledge from predicted labels with high certainty in labeling the remaining tokens. Our model achieved considerable results in SemEval-2020 Task 5 (Yang et al., 2020). Our results suggest label masking plus a multi-head attention layer as an efficient method both for regularization and accuracy improvement. The improvement with label masking was higher than the CRF, however, further studies is needed to evaluate the advantage of the introduced method in other sequence labeling tasks.

Acknowledgements

Part of the research performed by Carlo Lipizzi and Dario Borrelli leading to these results have received funding from the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(RE)) under Contract [HQ0034-19-D-0003, TO0150].

References

Pouria Babvey, Carlo Lipizzi, and Jose Emmanuel Ramirez-Marquez. 2019. Dissecting twitter discussion threads with topic-aware network visualization. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1359–1364. IEEE.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

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*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language?

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using bert-crf. *arXiv preprint arXiv:1909.10649*.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

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.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training.

Xiaoyu Yang, Stephen Obadinma, Huasha Zhao, Qiong Zhang, Stan Matwin, and Xiaodan Zhu. 2020. SemEval-2020 task 5: Counterfactual recognition. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain.