CNRL at SemEval-2020 Task 5: Modelling Causal Reasoning in Language with Multi-Head Self-Attention Weights based Counterfactual Detection

Rajaswa Patil[1,2] and **Veeky Baths**[1,3]
[1]Cognitive Neuroscience Lab, BITS Goa
[2]Department of Electrical & Electronics Engineering
[3]Department of Biological Sciences
BITS Pilani K. K. Birla Goa Campus, India
{f20170334, veeky}@goa.bits-pilani.ac.in

Abstract

In this paper, we describe an approach for modelling causal reasoning in natural language by detecting counterfactuals in text using multi-head self-attention weights. We use pre-trained transformer models to extract contextual embeddings and self-attention weights from the text. We show the use of convolutional layers to extract task-specific features from these self-attention weights. Further, we describe a fine-tuning approach with a common base model for knowledge sharing between the two closely related sub-tasks for counterfactual detection. We analyze and compare the performance of various transformer models in our experiments. Finally, we perform a qualitative analysis with the multi-head self-attention weights to interpret our models' dynamics.

1 Introduction

Causal reasoning is a process of detecting cause-effect relationships and is increasingly being used in artificial intelligence for improving generalization and interpretability. Modelling causal reasoning in language involves detecting such cause-effect relationships from natural language texts. A cause-effect relationship can be modelled as: *Event A causes Event B*. Counterfactuals describe events counter to facts and hence naturally involve common sense and causal reasoning. A counterfactual can be modelled as a cause-effect relationship of the form: *Event A could have caused Event B (Event A did not occur)*.

SemEval-2020 Task-5 (Yang et al., 2020) consists of two independent sub-tasks: a binary-classification task for detecting counterfactual statements and a span-detection task for detecting antecedent (*cause*) - consequent (*effect*) spans of given counterfactual statements (Table 1). In this work, we use multi-head self-attention weights from pre-trained transformer models (Vaswani et al., 2017) to capture the semantic interactions between the tokens of given text with respect to causal relations. We use a fine-tuning approach with a common base model for knowledge sharing between these two closely related sub-tasks. The code for this work is made publicly available as a GitHub repository.[1]

2 Background

Early work on causal reasoning and related tasks in natural language was based on various statistical and linguistic approaches (Asghar, 2016). Recent work for causal reasoning related tasks involves deep learning based approaches. Causal reasoning can be achieved through extraction of cause-effect relations with CRF and LSTM based sequence labelling tasks (Dasgupta et al., 2018). Counterfactuals can contain implicit causal relations (Table 1). Using multi-head self-attention at word level can help capture such implicit causal relations effectively (Liang et al., 2019). Current benchmarks for modelling causal reasoning involves question-answering tasks (Gordon et al., 2012). Using pre-trained transformer models have been effective on such tasks (Sap et al., 2019). Self-attention weights from such transformer models are usually structured in a 3-dimensional matrix. Using convolutional neural networks with these self-attention weight matrices can be helpful for extracting semantic features for downstream NLP tasks (Fang et al., 2019). Similar approaches can be used to extract features for detecting counterfactuals in natural language texts. Further, the multi-head self-attention attention weights from these models can also be used in interpretative qualitative analysis (Voita et al., 2019).

[1]https://github.com/rajaswa/counterfactual-detection-semeval-2020

Counterfactual Statement	Antecedent	Consequent
"If I had 10 pharmacists who worked with me, I could reach 100 people more effectively."	*If I had 10 pharmacists who worked with me*	*I could reach 100 people more effectively*
"Thanks for the article on this new term that fits me so well, wish all your articles were worthy of praise."	*wish all your articles were worthy of praise*	-

Table 1: Example counterfactual statements from the task dataset

3 Methodology

3.1 Base Architecture

We use the knowledge learned during the binary-classification counterfactual detection sub-task for the antecedent-consequent span-detection sub-task by defining a base architecture, common to both the sub-tasks (Figure 1). The base architecture is used to extract task-specific features, which are further passed on to task-specific modules. We first train the base architecture with a binary-classification module for the first sub-task. Then, we replace the binary-classification module with a regression-module and fine-tune the already trained base architecture for the more complex second sub-task.

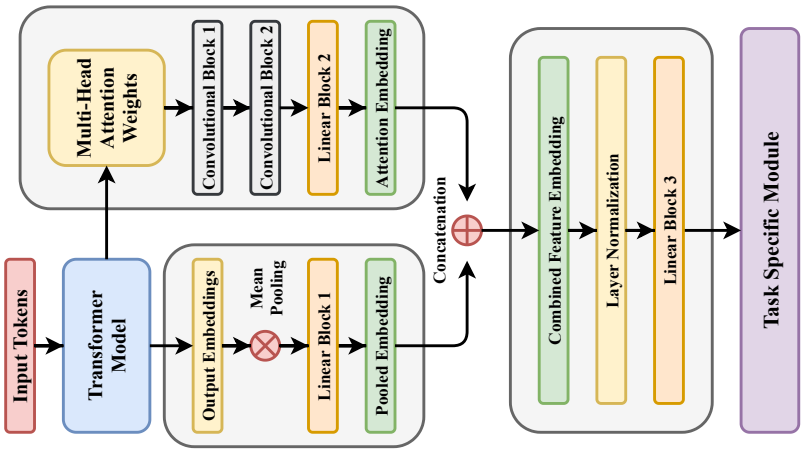

Figure 1: Base Architecture

For the base architecture, we use pre-trained transformer models to extract contextual output embeddings and multi-head self-attention weights from the tokenized input text. The output embeddings are passed through a pooling layer to get a pooled embedding. The multi-head self-attention weights are structured in a 3-dimensional matrix with the following dimensions: *(number of attention heads, number of tokens in the text, number of tokens in the text)*. This matrix is passed through convolutional and linear blocks to get an attention embedding. The pooled embedding and the attention embedding are concatenated together to form a combined feature embedding. We apply a layer normalization operation on the combined feature embedding for better generalization and stability for knowledge sharing across the sub-tasks. The feature embedding is then passed through a linear block and fed to the task-specific module. A linear block is composed of a fully connected layer with ReLU activation and dropout regularization, and a convolutional block is composed of a 2D-convolution layer with batch normalization, ReLU activation and dropout regularization. Here, we experiment with various pre-trained transformer models which differ from each other in terms of pre-training approach, architecture and number of parameters: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) (robust pre-training), XLNet (Yang et al., 2019) (autoregressive

model) and DistilBERT (Sanh et al., 2019) (distilled model). Usually, the last layer of any transformer model gets quickly biased for any individually trained task. Here, we concatenate the output embeddings and multi-head self-attention weights from the last three layers of the transformer model so that more generalized features are learned by the common base architecture while training for each of the sub-tasks separately.

3.2 Task Specific Modules

For the counterfactual detection sub-task, we have binary labels for various statements to be *counterfactuals / non-counterfactuals*. We use a linear block with sigmoid activation as a binary-classification module for this task. For the second sub-task, we have character-level span locations (*start-id* and *end-id*) for the antecedent and consequent spans of the given counterfactual statements. This can be treated as a regression problem with 4 feature values. We use an another linear block with ReLU activation and 4 output neurons as a regression module for this task. The lengths of various counterfactual statements in the second sub-task vary considerably across the dataset. This induces a certain variance in the character-level span location features. To handle this variance, we scale each of these 4 span features by the length of the counterfactual statement. The span features are scaled down by the lengths of their respective counterfactual statements during training. Consequently, we scale up the predicted span features during inference to obtain the actual antecedent-consequent span locations. The functioning of the regression module for antecedent-consequent detection during training and inference is explained in Algorithm 1.

Algorithm 1 Counterfactual Antecedent-Consequent Detection

while Training **do**
  for all $counterfactual \in data$ **do**
    $spans \Leftarrow (antecedent\_start,\ antecedent\_end,\ consequent\_start,\ consequent\_end)$
    $spans \Leftarrow \frac{spans}{length(counterfactual)}$
    $output \Leftarrow$ **RegressionModule**$(counterfactual)$
    $loss \Leftarrow$ **SmoothL1Loss**$(output,\ spans)$
  end for
end while
while Inference **do**
  for all $counterfactual \in data$ **do**
    $output \Leftarrow$ **RegressionModule**$(counterfactual)$
    $spans \Leftarrow output * length(counterfactual)$
  end for
end while

4 Experiments

For all our experiments, we use Binary Cross Entropy loss (*for counterfactual detection*) and Smooth L1 loss (*for antecedent-consequent span regression*) with Adam optimizer (*with weight decay*) to train our models. We use the PyTorch implementations of the smallest base variants of pre-trained transformer models by Hugging Face[2] (Wolf et al., 2019) in our base architecture. We use a 90-10 data split for training and development purposes respectively. The data distribution across the splits is shown in Table 2. We validate our models on F1 score (*for counterfactual detection*) and Smooth L1 loss (*for antecedent-consequent detection*). The evaluation metrics for the counterfactual detection task are the binary precision, recall and F1 scores. For the antecedent-consequent span detection task, the precision, recall and F1 score are defined as sequence labelling metrics with respect to the overlap between the predicted and the ground truth spans[3].

[2] https://huggingface.co/

[3] https://github.com/arielsho/Task-5_Baseline/blob/master/subtask2_baseline.py

Task	Training	Development	Test
Counterfactual Detection	11700	1300	7000
Antecedent-Consequent Detection	3195	356	1950

Table 2: Data distribution across the splits (number of samples)

5 Discussion

For the final submission, we use RoBERTa and BERT as the transformer model in our base architecture for sub-task 1 and sub-task 2 respectively. On the final task leaderboard, our system ranks 13th (0.845 F1) for the counterfactual detection sub-task and 7th (0.688 F1) for the antecedent-consequent detection sub-task. Since we treat antecedent-consequent detection as a regression task, we do not monitor the *Exact Match score* between the predicted and ground truth spans, which is more significant for token-level sequence labelling based approaches. We analyse the performance of various transformer models with hyperparameter tuning post evaluation (Table 3). RoBERTa gives the best results for the counterfactual detection sub-task. Whereas, BERT gives the best results for the antecedent-consequent detection sub-task. DistilBERT, a considerably smaller model (65M parameters) shows comparable performance with the rest of the transformer models (110M+ parameters) for both the sub-tasks.

Transformer Model	Sub-Task 1			Sub-Task 2		
	F1	Recall	Precision	F1	Recall	Precision
BERT	0.824	0.787	0.863	**0.688**	**0.672**	0.74
RoBERTa	**0.863**	0.847	**0.879**	0.644	0.567	**0.822**
XLNet	0.823	**0.862**	0.788	0.634	0.554	0.816
DistilBERT	0.788	0.825	0.754	0.639	0.566	0.812

Table 3: Post evaluation analysis of various transformer models in base architecture

Since BERT performs marginally better than rest of the transformer models for antecedent-consequent detection sub-task, we consider BERT for the further qualitative analysis. We inspect the multi-head self-attention weights from the final layer of BERT (Vig, 2019) to interpret the model's dynamics. Overall, the model assigns more attention to certain parts of text which are related to the conditional nature of the counterfactual statements. Moreover, we see that some of the attention-heads learn to assign more attention to some specific parts of the text. Head^{1,6} assign maximum attention to punctuation and Head^{2,4,12} focus more on the auxiliary verbs. Head^{3,11} attend to conjunctions and verbs which act as causal connectives in the text. Head^{5} and Head^{7} attend to entities and numerical values respectively (if present). This property of linguistically selective-attention of the attention-heads can be observed in the following examples of antecedent-consequent spans detected by our system (rounded off to include partially covered words). Where, an underline indicates the detected antecedent and the detected consequent is made bold.

1. If^{3,11} only^{3} Trump had listened to Chris^{5} Christie^{5} **he wouldn't^{2,4,12} be in this mess.**
2. If^{3,11} this were an open seat^{5}, you **would^{2,4,12} have six, eight^{7}, maybe 12^{7} people running.**
3. 'I wish^{2,4,11} I was 40^{7} years old, but I'm not,' he told^{12} POLITICO^{5}.
4. I could^{2,4,11} have been you and you could^{3,12} have been me.
5. Of course **the company wouldn't^{2,4,7,12} have^{3} had to sell such a prized** asset if^{11} it had other options to raise^{5} capital.

The superscripts here represent the most attending attention-head(s) for the corresponding word. The same can be confirmed by a visualization (Figure 2) of the head-wise color coded self-attention weights. For example 3 and 4, we have no consequent part in the text. Our system detects (0,0) as consequent span start and end locations for such counterfactual statements, indicating the absence of the consequent. The conjunctions (*but / and*) in such counterfactual statements are ignored by the attention-heads. But the conjunctions (*if*) in counterfactual statements with a consequent part (Example 1,2 and 5) are highly attended by the attention-heads through the tokens from entire sequence. This shows the ability of the model to differentiate the causal connectives from the non-causal ones in the text. Punctuation play an important role here as they are usually present near the boundaries of antecedent-consequent spans (Example 2 and 3). Auxiliary verbs (would, wouldn't, could, have) are assigned maximum attention across all the examples as they directly correspond to the conditional nature of counterfactual statements.

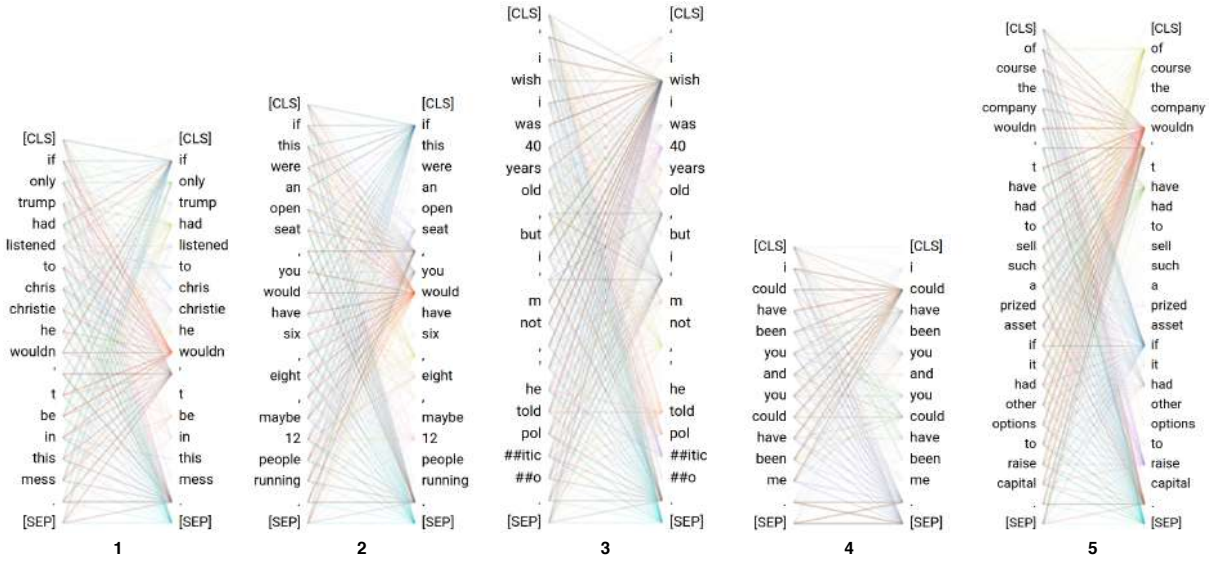

Figure 2: Head-wise self-attention weights visualization for BERT in base architecture

6 Conclusion

Our proposed approach uses multi-head self-attention weights from transformer models to detect causal relations for counterfactual detection in text. Through our experiments, we find that RoBERTa overall shows the best performance for counterfactual detection task and BERT performs the best for the antecedent-consequent detection task. We show that even smaller transformer models like DistilBERT perform counterfactual detection tasks effectively. With knowledge sharing between the two sub-tasks, our system detects antecedent-consequent spans in counterfactual statements with good efficiency by a simple regression over the spans. This can possibly be further improved by post-inference processing on the predicted spans or replacing the regression module with a token-level sequence labelling module. Further, we show that through our approach, the attention-heads attain a property of assigning linguistically selective-attention with respect to the conditional nature of the counterfactual statements.

Acknowledgements

This work was carried out at the Cognitive Neuroscience Lab at BITS Goa[4] through the funding from DST-CSRI SR/CSRI/50/2014(G).

[4] http://bitscogneuro.com/

References

Nabiha Asghar. 2016. Automatic extraction of causal relations from natural language texts: A comprehensive survey. *CoRR*, abs/1605.07895.

Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. 2018. Automatic extraction of causal relations from text using linguistically informed deep neural networks. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 306–316, Melbourne, Australia, July. 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. 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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Yong Fang, Jian Gao, Cheng Huang, Hua Peng, and Runpu Wu. 2019. Self multi-head attention-based convolutional neural networks for fake news detection. *PLOS ONE*, 14(9):1–13, 09.

Andrew Gordon, Zornitsa Kozareva, and Melissa Roemmele. 2012. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 394–398, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Shining Liang, Wanli Zuo, Zhenkun Shi, and Sen Wang. 2019. A multi-level neural network for implicit causality detection in web texts.

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. *CoRR*, abs/1907.11692.

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

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

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

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *CoRR*, abs/1905.09418.

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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

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.