Modeling Causal Reasoning in Language: Detecting Counterfactuals
CS263 Project Report (Spring 2022) - Group 3

Dipti Ranjan Sahu (405526161, diptisahu11@g.ucla.edu)
Abirami Anbumani (005526158, ad14abirami@g.ucla.edu)
Shivam Patel (805626050, shivambpatel@g.ucla.edu)
Rohit Sunchu (505525335, sunchurohit@ucla.edu)

Abstract

This report summarizes the results and findings from our experiment with SemEval 2020 Task 5 - Modelling Causal Reasoning in Language: Detecting Counterfactuals. This task has two subtasks: counterfactual sentence classification, and detecting consequent and antecedents in counterfactual sentences. We experimented with various state-of-the-art language representation models (LRMs) and their tokenizers. We also implemented a novel Neuro-symbolic classification model for subtask-1 and analyzed the results. Our experimental results demonstrate that RoBERTa with class-weighted loss gives the best accuracy for subtask-1, and RoBERTa LRM performs the best for subtask-2. Our code is available at https://github.com/shivampatel712/counterfactual-detection.

1. Introduction

Counterfactual statements describe hypothetical possibilities of outcomes for actions that did not occur. Counterfactual sentences consist of two parts: consequent and antecedent; consequent describes potential outcomes, and antecedent describes the actions or circumstances that did not or can not occur. Inspired by (Yang et al., 2020), we worked on counterfactual recognition task provided by SemEval 2020 and used the dataset provided. The two subtasks in this task are: Recognizing Counterfactual Statements and Detecting Antecedents and Consequents. Our main contributions to this project are:

- Implementation of various transformer models for subtask-1 and comparing them
- A novel neuro-symbolic language model for subtask-1
- Implementation of state-of-the-art BERT model presented in (Lu et al., 2020b)
- Thorough analysis of various language models and comparison between them

Our work is organized into the following sections: Literature Survey, Data, Model/Approach, Evaluation, Results & Analysis, Conclusion, and Future Scope.

2. Literature Survey

There are several approaches implemented for this task since it is part of a competition. (Ding et al., 2020b) uses pretrained transformers along with pseudo-labeling for subtask-1. (Lu et al., 2020a) utilize a new word representation layer in transformers, whereas (Yabloko, 2020) implements a cloud-computing architecture along with transformer and dependency parsers. Based on the resultant F1 score, our approach would have placed 6^{th} (without ensembling).

(Akl et al., 2020) treats subtask-2 as a named entity recognition task and performs Conditional Random Field and discriminative models based recognition. They utilize BiLSTM-CRF model for the same. (Ding et al., 2020a), (Patil & Baths, 2020), (Fajcik et al., 2020), and (Lu et al., 2020b) utilize transformer based models for this subtask. Due to their superiority in performance, we have also designed transformer, specifically BERT-based models inspired by (Lu et al., 2020b).

3. Data

3.1. Data Acquisition

We use the datasets provided in the SemEval task. Examples from the datasets for both the subtasks are as follows:

- Recognizing counterfactual statements:

  The leak could have been stopped the same hour it was discovered if the well had a working shut-off valve. - **Label 1 i.e., counterfactual**

  The new request, if approved, would keep the military forces on the border through Jan. - **Label 0 i.e., not counterfactual**

- Detecting Antecedents and Consequents:

  The GOP's malignant amnesia regarding the economy would be hilarious were it not for the wreckage they caused.

Antecedent: *were it not for the wreckage they caused*
Consequent: *The GOP's malignant amnesia regarding the economy would be hilarious*
Label: $69,\ 108,\ 0,\ 67$

3.2. Data Cleaning and Preprocessing

For subtask-1, we performed a custom transformer preprocessing which includes the following: removing URL, hashtags, numbers, emojis, punctuations, and converting special few characters into words. We use the pre-trained Language Model (LM) tokenizers like BERT-tokenizer and RoBERTa-tokenizer to generate the embedding vectors for sample sentences. We noticed the optimal sentence length to be 100 in the data (99.24% of sentences had less than 100 words each) and truncated each input sentence length to it before feeding the data into transformers. This was done to reduce training time.

For subtask-2, preprocessing would change the character indices and thus resulting in inaccurate output. Therefore no special preprocessing was performed other than tokenization using a pre-trained tokenizer. All the input sequences fit the maximum input length of 509 tokens.

3.3. Data Exploration

For subtask-1, from data exploration we received the following insights:

- The data is highly skewed with 89% percent of sentences being not counterfactual in nature as shown in Figure 1. This insight prompted us to incorporate class-weighted cross-entropy loss for better performance. We also concluded that F1 metric is a better measure of model performance than accuracy.

- 99.24% of the sentences had a length less than or equal to 100 as shown in Figure 2. This insight allowed us to reduce the model training time by 3.

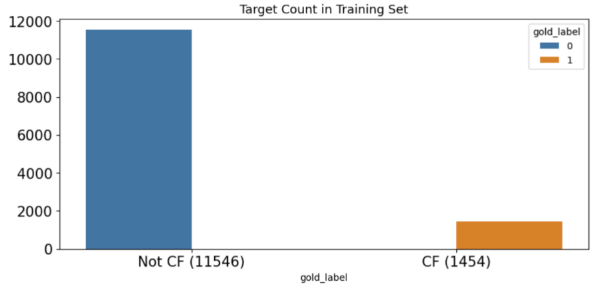

Figure 1. Data Distribution for subtask-1

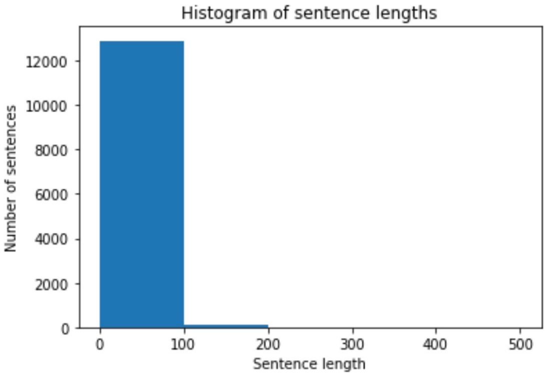

Figure 2. Word length of sentences for subtask-1

4. Model/Approach

4.1. Subtask-1

We have implemented the following models for counterfactual sentence classification: SVM(Baseline), BERT (Devlin et al., 2018), BERT class-weighted, RoBERTa (Liu et al., 2019), RoBERTa class-weighted, and RoBERTa class-weighted Neuro-symbolic. Transformers are the current state-of-the-art for text classification and therefore we implemented them. To account for class imbalance in data, we incorporate a class-weighted cross-entropy loss for training the transformers.

We also ideated a basic neuro-symbolic approach and implemented it. We understood the different types of counterfactual statements as provided in (Yang et al., 2020) and (Son et al., 2017), and implemented a symbolic knowledge encoding of which type of counterfactual statement a particular sentence is. Since not all types have a clear template for being classified as counterfactual, only a few of these types were chosen and others were taken together in a single category. For each of these categories, we trained the RoBERTa model.

Only template-based classification (symbolic approach) will not work since there are negative examples for every template. E.g., ”If I had worked hard, then I would have gotten better scores in the exam” is an example of an ”if...then...” type counterfactual statement. However, ”If I like some subject, then I will study well” is not a counterfactual statement even though it has the ”if...then...” pattern. Although the deep learning-based approach works decently well for this classification task, we theorized that only required predicting from one or few types of counterfactuals compared to all types enables the model to perform better classification due to reduced complexity of the input. Since the data was separable into types, we decided to combine these two approaches to form a neuro-symbolic approach. This approach qualifies to be titled neuro-symbolic because we encode background knowledge information (is the sentence ”if...then...” type or ”what if...” type counterfactual) like that of the symbolic approach before training a deep learning model on it.

We finally compare the results of all our models with each other and a baseline-SVM (provided by the competition host) model.

4.2. Subtask-2

We used the following models for antecedent and consequent extraction: SVM (baseline), BERT-cased, BERT-uncased, and RoBERTa. For the BERT-based models, we approached this problem as a question-answering task similar to (Lu et al., 2020b). Given a counterfactual statement s, we construct 2 questions, q_a for antecedent and q_c for consequent and answer these 2 questions to get the start and end indices of antecedent and consequent. In our case, we set q_a to ”antecedent” and q_c to be ”consequent”. For the context, we directly use the counterfactual statement s.

We combine the question q and the statement s into a single packed sequence: $\{[CLS], q, [SEP], s, [SEP]\}$. We pass this sequence to the BERT-based models. (Note that q and s are tokenized before being passed to the BERT-based models). Let $\mathbf{h}_q^i \in R^d$ be the final output vector of the BERT model corresponding to the i^{th} token of the query (q). Similarly, let $\mathbf{h}_s^i \in R^d$ be the final output vector corresponding to the i^{th} token of the counterfactual statement (s) (context) and $C \in R^d$ be the output vector corresponding to $[CLS]$ token.

To extract the start and end indices of the answer, we use a pointer network which contains 2 vectors. Let the vectors of this pointer network be $\mathbf{w}_{start} \in R^d$ and $\mathbf{w}_{end} \in R^d$. The score of token i (of the counterfactual statement s) being the start of the answer is computed as a dot product between $\mathbf{w}_{start}$ and $\mathbf{h}_s^i$. In a similar way, we use $\mathbf{w}_{end}$ to compute the score of token i being the end of the answer. Therefore, the score of the answer being from position j to position k is calculated a $S_{j,k} = \mathbf{w}_{start} \cdot \mathbf{h}_s^j + \mathbf{w}_{end} \cdot \mathbf{h}_s^k$ where $k \geq j$. Since some statements do not have consequent, we use the $[CLS]$ token as the start and end of the answer span. So the score of a statement without an answer is $S_{null} = \mathbf{w}_{start} \cdot C + \mathbf{w}_{end} \cdot C$.

For training, we maximize the log-likelihood of the data. The corresponding loss function is

$$\mathbf{L} = -\sum_{i \in \mathcal{D}} \sum_{q \in \{q_a, q_c\}} (log(P(y_{start} = j^*|x_i, q)) + log(P(y_{end} = k^*|x_i, q)))$$

$$P(y_{start} = j^*|x_i, q) = \frac{exp(w_{start} \cdot \mathbf{h}_s^{j*})}{exp(w_{start} \cdot C) + \sum_{j=1}^{n} exp(w_{start} \cdot \mathbf{h}_s^j)}$$

$$P(y_{end} = k^*|x_i, q) = \frac{exp(w_{end} \cdot \mathbf{h}_s^{k*})}{exp(w_{end} \cdot C) + \sum_{k=1}^{n} exp(w_{end} \cdot \mathbf{h}_s^k)}$$

where j^* and k^* are the actual labels to the corresponding questions. The parameters of this pointer network are trained from scratch.

5. Evaluation

Since subtask-1 is binary classification, we decided to use accuracy, precision, recall and F1 as our evaluation metrics. For subtask-2, we add a new evaluation metric - exact match as accuracy does not make sense in the case of phrase detection. Furthermore, precision and recall are done character-wise for subtask-2.

- **Accuracy:** Total number of samples predicted correctly out of total sample. Acc = $\frac{TP+TN}{TP+TN+FP+FN}$
- **Precision:** number of correct positive predictions made out of total number of positive predictions. Precision = $\frac{TP}{TP+FP}$
- **Recall:** Number of correct positive predictions made out of all positive samples in their ground truth. Recall = $\frac{TP}{TP+FN}$
- **F1:** Harmonic mean between precision and recall. F1 = $\frac{2*Precision*Recall}{Precision+Recall}$
- **Exact Match:** Number of phrases of antecedent and consequent that exactly matches with the ground truth.

The loss function used for subtask 1 is **binary cross entropy**, which evaluates the performance by comparing the actual class labels and the predicted probabilities.

6. Results & Analysis

Model	Acc.	F1	Prec.	Recall
SVM(Baseline)	0.9006	0.1595	0.7321	0.0895
BERT	0.9680	0.8470	0.8540	0.8401
BERT - class weighted	0.9691	0.8492	**0.8760**	0.8238
RoBERTa	0.9711	0.8624	0.8671	0.8577
RoBERTa - class weighted	**0.9736**	**0.8746**	0.8752	**0.874**
RoBERTa - class weighted Neuro Symbolic	0.9711	0.8631	0.8631	0.8631

Table 1. Subtask-1 Results

From the subtask-1 results, we understand the following:

- SVM performs poorly which can be particularly seen from its very low recall value. Such a low recall value reflects that the SVM is unable to detect the majority of the counterfactuals present in the dataset. We can attribute this to the low classification capacity of SVMs.
- BERT and RoBERTa being transformers, clearly outperform SVM by a huge margin. Additionally, since

the RoBERTa model is a robustly optimized BERT approach, it works better than BERT as expected.

- Class-weighted models perform better than their corresponding backbone models because the class-weighted approach takes into account the high class imbalance in the data.
- RoBERTa - class weighted neuro symbolic model performs comparably to our best model. It is the second-best model as seen from table 1. However, it doesn't perform as well as its backbone model.
  - Since the separation of data into different categories could be uneven in number, we hypothesized that some models in the ensemble could have to overfit. However, we had chosen the checkpoint with the best validation accuracy separately for every model and therefore concluded that the reason for such performance drop cannot be attributed to overfitting.
  - When inputting class weight for the RoBERTa models, we carefully calculated class weights for each RoBERTa separately based on the type of separated data given to it. Therefore incorrect class weights cannot be attributed as a reason for the drop in performance.
  - We noticed that the data is not entirely separable into these types given by the problem statement and sentences may have overlap between these counterfactual statement types. Therefore we have come up with the conjecture that this could be a possible reason for the drop in performance. From this, we ideated that we could group the counterfactual statement data that were separated based on their type into all possible permutations and train a RoBERTa for each of these. However, this is computationally expensive and an infeasible approach.

For subtask 2, we can deduce the following:

- From Table 2, we see that BERT-uncased slightly outperforms BERT-cased model. This is because case information is not very useful in this task. (Case information is more useful in tasks like Named Entity Recognition).
- RoBERTa outperforms both BERT-cased and BERT-uncased on both the metrics - Exact Match and F1 score.
- SVM (baseline model) doesn't perform very well compared to the other models. This is expected because the number of parameters of SVM is very less compared to the number of parameters of BERT-based models.

Model	EM	F1	Prec.	Recall
SVM(Baseline)	0.337	0.543	0.554	0.538
BERT-cased	0.4584	0.8036	0.7823	0.8262
BERT-uncased	0.4617	0.8111	0.7937	0.8293
RoBERTa	**0.4698**	**0.8180**	**0.8058**	**0.8307**

Table 2. Subtask-2 Results

7. Conclusion

We examined the performance of various state-of-the-art language representation models on both the subtasks and we found yet another NLP task benefits from fine-tuning a pre-trained model. In both cases, we found the RoBERTa model to perform slightly better than other LRMs, while its results also being more stable. We also tried out a new approach to Neuro-Symbolic parsing. Though the model was not able to beat RoBERTa, it was comparable to its performance.

8. Future Work

As the next exploration for subtask-1, we hope to look at the attention layer of our RoBERTa-class weighted model and decide what causes a model to classify certain sentences into counterfactuals and others to not. This information will be encoded into the model as part of the symbolic approach.

Training models involved a huge amount of time, as well as incurred expensive computation costs. So, it was quite out of bounds to tune the hyper-parameters for these models. We would also try to conduct our experiments on other LMs and perform the analysis.

One of the tasks in our mind was to check the robustness of these models. We would have added an RND defense for making the model robust to noises. Again, these models involved a high amount of training time and computational resources, so it was quite difficult to implement the robustness feature of the model.

9. Work Division

Subtask-1: Shivam Patel, Abirami Anbumani

Subtask-2: Rohit Sunchu, Dipti Ranjan Sahu

References

Akl, H. A., Mariko, D., and Labidurie, E. Yseop at semeval-2020 task 5: Cascaded bert language model for counterfactual statement analysis. *arXiv preprint arXiv:2005.08519*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, X., Hao, D., Zhang, Y., Liao, K., Li, Z., Qin, B., and Liu, T. Hit-scir at semeval-2020 task 5: Training pre-trained language model with pseudo-labeling data for counterfactuals detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 354–360, 2020a.

Ding, X., Hao, D., Zhang, Y., Liao, K., Li, Z., Qin, B., and Liu, T. HIT-SCIR at SemEval-2020 task 5: Training pre-trained language model with pseudo-labeling data for counterfactuals detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 354–360, Barcelona (online), December 2020b. International Committee for Computational Linguistics. doi: 10.18653/v1/2020.semeval-1.43. URL https://aclanthology.org/2020.semeval-1.43.

Fajcik, M., Jon, J., Docekal, M., and Smrz, P. But-fit at semeval-2020 task 5: Automatic detection of counterfactual statements with deep pre-trained language representation models, 2020. URL https://arxiv.org/abs/2007.14128.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lu, Y., Li, A., Lin, H., Han, X., and Sun, L. ISCAS at SemEval-2020 task 5: Pre-trained transformers for counterfactual statement modeling. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 658–663, Barcelona (online), December 2020a. International Committee for Computational Linguistics. doi: 10.18653/v1/2020.semeval-1.85. URL https://aclanthology.org/2020.semeval-1.85.

Lu, Y., Li, A., Lin, H., Han, X., and Sun, L. Iscas at semeval-2020 task 5: Pre-trained transformers for counterfactual statement modeling. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 658–663, 2020b.

Patil, R. and Baths, V. Cnrl at semeval-2020 task 5: Modelling causal reasoning in language with multi-head self-attention weights based counterfactual detection. *arXiv preprint arXiv:2006.00609*, 2020.

Son, Y., Buffone, A., Raso, J., Larche, A., Janocko, A., Zembroski, K., Schwartz, H. A., and Ungar, L. H. Recognizing counterfactual thinking in social media texts. In *ACL (2)*, pp. 654–658, 2017. URL https://doi.org/10.18653/v1/P17-2103.

Yabloko, L. Ethan at semeval-2020 task 5: Modelling causal reasoning in language using neuro-symbolic cloud computing. 08 2020. doi: 10.18653/v1/2020.semeval-1.83.

Yang, X., Obadinma, S., Zhao, H., Zhang, Q., Matwin, S., and Zhu, X. SemEval-2020 task 5: Counterfactual recognition. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain, 2020.