

# BUT-FIT at SemEval-2020 Task 5: Automatic detection of counterfactual statements with deep pre-trained language representation models

**Martin Fajcik, Josef Jon, Martin Docekal, Pavel Smrz**  
Brno University of Technology, Faculty of Information Technology  
612 66 Brno, Czech Republic  
{ifajcik, ijon, idocekal, smrz}@fit.vutbr.cz

## Abstract

This paper describes BUT-FIT’s submission at SemEval-2020 Task 5: Modelling Causal Reasoning in Language: Detecting Counterfactuals. The challenge focused on detecting whether a given statement contains a counterfactual (Subtask 1) and extracting both antecedent and consequent parts of the counterfactual from the text (Subtask 2). We experimented with various state-of-the-art language representation models (LRMs). We found RoBERTa LRM to perform the best in both subtasks. We achieved the first place in both exact match and F1 for Subtask 2 and ranked second for Subtask 1.

## 1 Introduction

One of the concerns of SemEval-2020 Task 5: Modelling Causal Reasoning in Language: Detecting Counterfactuals (Yang et al., 2020) is to research the extent to which current state-of-the-art systems can detect counterfactual statements. A counterfactual statement, as defined in this competition, is a conditional composed of two parts. The former part is the antecedent – a statement that is contradictory to known facts. The latter is the consequent – a statement that describes what would happen had the antecedent held<sup>1</sup>. To detect a counterfactual statement, the system often needs to possess a commonsense world knowledge to detect whether the antecedent contradicts with it. In addition, such a system must have an ability to reason over consequences that would arise had the antecedent would have been true. In some cases, the consequent might not be present at all, but instead a sequence resembling consequent, but with no consequential statement, might be present. Figure 1 shows a set of examples drawn from the data.

**If that was my daughter, I would have asked if I did something wrong.**

Nike’s revenue last quarter alone climbed a solid 5%, to \$8.432 billion, and **would have increased 7% had it not been for foreign-currency exchange.**  
Al Sharpton’s storefront headquarters in Harlem, the gatherings were solemn, spirited and reflected the fraught nature of **what would have been the Rev.**

Figure 1: Three examples from the training data containing counterfactual statements. Antecedents are highlighted with red bold, consequents with blue bold italic. The last example has no consequent.

Counterfactuals were studied across a wide spectrum of domains. For instance, logicians and philosophers focus on logical rules between parts of counterfactual and its outcome (Goodman, 1947). Political scientists conducted counterfactual thought experiments for hypothetical tests on historical events, policies or other aspects of society (Tetlock and Belkin, 1996). However, there is only a small amount of work in computational linguistics studying this phenomenon. SemEval-2020 Task 5 aims at filling this gap in the field. The challenge consists of two subtasks:

1. Detecting counterfactual statements – classify whether the sentence has a counterfactual statement.
2. Detecting antecedent and consequence – extract boundaries of antecedent and consequent from the input text.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>.

<sup>1</sup>According to several definitions in literature, e.g. (Starr, 2019), the antecedent of counterfactual might not need to counter the facts.

The approaches we adopted follow recent advancements from deep pre-trained language representation models. In particular, we experimented with fine-tuning of BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019) models. Our implementation is available online<sup>2</sup>.

## 2 System overview

### 2.1 Language Representation Models

We experimented with three language representation models (LRMs):

**BERT** (Devlin et al., 2019) is pre-trained using the multi-task objective consisting of denoising LM and inter-sentence coherence (ISC) sub-objectives. The LM objective aims at predicting the identity of 15% randomly masked tokens present at the input<sup>3</sup>. Given two sentences from the corpus, the ISC objective is to classify whether the second sentence follows the first sentence in the corpus. The sentence is replaced randomly in half of the cases. During the pre-training, the input consists of two documents, each represented as a sequence of tokens divided by special *[SEP]* token and preceded by *[CLS]* token used by the ISC objective, i.e. *[CLS]document<sub>1</sub>[SEP]document<sub>2</sub>[SEP]*. The input tokens are represented via jointly learned token embeddings  $E_t$ , segment embeddings  $E_s$  capturing whether the word belongs into *document<sub>1</sub>* or *document<sub>2</sub>* and positional embeddings  $E_p$  since self-attention is position-invariant operation. During fine-tuning, we leave the second segment empty.

**RoBERTa** (Liu et al., 2019) is a BERT-like model with the different training procedure. This includes dropping the ISC sub-objective, tokenizing via byte pair encoding (Sennrich et al., 2016) instead of WordPiece, full-length training sequences, more training data, larger batch size, dynamic token masking instead of token masking done during preprocessing and more hyperparameter tuning.

**ALBERT** (Lan et al., 2019) is a RoBERTa-like model, but with n-gram token masking (consecutive n-grams of random length from the input are masked), cross-layer parameter sharing, novel ISC objective that aims at detecting whether the order of two consecutive sentences matches the data, input embedding factorization, SentencePiece tokenization (Kudo and Richardson, 2018) and much larger model dimension. The model is currently at the top of leaderboards for many natural language understanding tasks including GLUE (Wang et al., 2018) and SQuAD2.0 (Rajpurkar et al., 2018).

### 2.2 Subtask 1: Detecting counterfactual statements

The first part of the challenge is a binary classification task, where the participating systems determine whether the input sentence is a counterfactual statement.

A baseline system applying an SVM classifier (Cortes and Vapnik, 1995) over TF-IDF features was supplied by the organizers. We modified this script to use other simple classifiers over the same features – namely Gaussian Naive Bayes and 6-layer perceptron network, with 64 neurons in each layer.

As a more serious attempt at tackling the task, we compare these baselines with state-of-the-art LRMs – RoBERTa and ALBERT. The input is encoded the same way as in 2.3. We trained both models with cross-entropy objective and we used the linear transformation of CLS-level output after applying dropout for classification. After the hyperparameter search, we found that RoBERTa model performed the best on this task. For our final system, we built an ensemble from the best checkpoints of RoBERTa model.

### 2.3 Subtask 2: Detecting antecedent and consequence

We extended each LRM in the same way Devlin et al. (2019) extended BERT for SQuAD. The input representation for input  $x$  is obtained by summing the input embedding matrices  $E = E_t + E_s + E_p \in \mathbb{R}^{L \times d_i}$  representing its word embeddings  $E_t$ , position embeddings  $E_p$  and segment embeddings<sup>4</sup>  $E_s$  with  $L$  being the input length and  $d_i$  input dimensionality. Applying LRM and dropout  $\delta$ , an output matrix  $\delta(LRM(E)) = H \in \mathbb{R}^{L \times d_o}$  is obtained,  $d_o$  being the LRM’s output dimensionality. Finally, a linear transformation is applied to obtain logit vector for antecedent start/end  $a_s, a_e$  and consequent start/end  $c_s,$

<sup>2</sup>[https://github.com/MFajcik/SemEval\\_2020\\_Task-5](https://github.com/MFajcik/SemEval_2020_Task-5)

<sup>3</sup>The explanation of token masking is simplified and we refer readers to read details in the original paper (Devlin et al., 2019).

<sup>4</sup>RoBERTa is not using segment embeddings.

$c_e$ .

$$P_j(\cdot|x) \propto \exp(\mathbf{w}_j^\top \mathbf{H}); j \in \{a_s, a_e, c_s, c_e\} \quad (1)$$

For consequent, we do not mask CLS-level output and use it as a no consequent option for both  $c_s$  and  $c_e$ . Therefore we predict that there is no consequent iff model’s prediction is  $c_s = 0$  and  $c_e = 0$ ; assuming 0 is the index of CLS-level output. Finally, the log-softmax is applied and model is trained via minimizing cross-entropy for each tuple of inputs  $x$  and target indices  $t$  from the dataset  $\mathcal{D}$ .

$$- \sum_{(x,t) \in \mathcal{D}} \sum_{j \in \{a_s, a_e, c_s, c_e\}} \log P_j(t_j|x) \quad (2)$$

An ensemble was built using a greedy heuristic seeking the smallest subset from the pool of trained models s.t. it obtains best exact match on a validation set <sup>5</sup>.

### 3 Experimental setup

#### 3.1 Data

For each subtask, training datasets without split were provided. Therefore we took the first 3000 examples from Subtask 1 data and 355 random examples from Subtask 2 data as the validation data. The train/validation/test split was 10000/3000/7000 for Subtask 1 and 3196/355/1950 for Subtask 2. 88.2% of the validation examples in Subtask 1 were labeled 0 (non-counterfactual).

#### 3.2 Preprocessing & Tools

In case of Subtask 1, after performing a length analysis on the data, we truncated input sequences at length of 100 tokens for the LM based models in order to reduce worst-case memory requirements, since only 0.41% of the training sentences were longer than this limit. A histogram of the example lengths in tokens is presented in Appendix A.2. For Subtask 2, all the input sequences fit the maximum input length of 509 tokens.

For the preliminary experiments with simpler machine learning methods, we adopted the baseline script provided by the organizers, which is based on `sklearn` Python module. We implemented our neural network models in `PyTorch` (Paszke et al., 2019) using `transformers` (Wolf et al., 2019) library. In particular, we experimented with `roberta-large` and `albert-xxlarge-v2` in Subtask 1 and with `bert-base-uncased`, `bert-large-uncased`, `roberta-large` and `albert-xxlarge-v1` models in Subtask 2. We used `hyperopt` (Bergstra et al., 2013) to tune model hyperparameters. See Appendix A.1 for further details on hyperparameters. We used the Adam optimizer with a decoupled weight decay (Loshchilov and Hutter, 2017). For Subtask 2, we combined this optimizer with lookahead (Zhang et al., 2019). All models were trained on 12GB GPU.

### 4 Results and analysis

For **Subtask 1**, we adapted the baseline provided by the task organizer to asses how more classical machine learning approaches perform on the dataset. After seeing the subpar performance, we turned our attention to pre-trained LRMs, namely RoBERTa and ALBERT. The results of the best run of each model can be found in Table 1. A more comprehensive list of results for different hyperparameters can be found in the Appendix 3.

Our final submission is an ensemble of RoBERTa-large models since we found that this LRM performs better than ALBERT for this task. We trained a number of models on the train set and computed F1 scores on the validation part. 10 best (in terms of F1) single models were selected, and the output probabilities were averaged for all the possible combinations of these models. The combination with highest F1 score was selected as a final ensemble. Then we trained new models with the same parameters as the models in the ensemble, but using the whole training data, including the part that was previously used for validation. Finally, for our submitted ensemble, we used checkpoints saved after the same number of updates as the best checkpoints for the systems trained only on part of the training data.

<sup>5</sup>For more details on how the ensemble was built, see TOP-N fusion in Fajcik et al. (2019).

| Model             | Precision               | Recall                  | F1                      |
|-------------------|-------------------------|-------------------------|-------------------------|
| SVM               | 80.55                   | 8.19                    | 14.87                   |
| Naive Bayes       | 22.81                   | 28.81                   | 25.47                   |
| MLP               | 39.01                   | 29.09                   | 33.33                   |
| RoBERTa-large     | <b>98.11</b> $\pm$ 0.11 | <b>85.71</b> $\pm$ 0.61 | <b>91.48</b> $\pm$ 0.51 |
| ALBERT-xxlarge    | 97.99 $\pm$ 0.15        | 84.19 $\pm$ 0.55        | 90.59 $\pm$ 0.67        |
| RoBERTa-large-ens | 98.95                   | 87.30                   | 92.76                   |

Table 1: Results of different models on validation part of Subtask 1 training data (first 3000 sentences). Results for RoBERTa and ALBERT models are averaged over ten training runs with the best found hyperparameters.

We performed an error analysis of the best single RoBERTa and ALBERT models. RoBERTa model misclassified 52 examples (29 false positives, 23 false negatives), while ALBERT misclassified 60 examples (32 false positives, 23 false negatives). 29 wrongly classified examples were common for both of the models. Examples of wrongly classified statements are presented in the Appendix A.3.

|                             | BERT-base       | BERT-large      | ALBERT          | RoBERTa                | RoBERTa <sub>(Ensemble)</sub> |
|-----------------------------|-----------------|-----------------|-----------------|------------------------|-------------------------------|
| <b>EM</b>                   | 73.03 $\pm$ 0.4 | 72.68 $\pm$ 0.6 | 73.59 $\pm$ 0.8 | <b>73.92</b> $\pm$ 0.3 | 74.93                         |
| <b>F1</b>                   | 87.13 $\pm$ 0.4 | 87.31 $\pm$ 0.5 | 88.13 $\pm$ 0.4 | <b>88.27</b> $\pm$ 0.4 | 80.80                         |
| <b>A<sub>EM</sub></b>       | 77.18           | 76.56           | <b>77.97</b>    | 77.58                  | 79.44                         |
| <b>A<sub>F1</sub></b>       | 91.11           | 90.93           | <b>91.71</b>    | 91.33                  | 91.96                         |
| <b>C<sub>EM</sub></b>       | 68.87           | 68.70           | 69.21           | <b>70.28</b>           | 70.42                         |
| <b>C<sub>F1</sub></b>       | 83.15           | 83.70           | 84.54           | <b>85.23</b>           | 85.65                         |
| <b>ACC<sub>no-c</sub></b>   | 90.37           | 91.10           | 91.49           | <b>91.69</b>           | 91.83                         |
| <b>#<math>\theta</math></b> | 109M            | 335M            | 223M            | 355M                   | -                             |

Table 2: Results on the Subtask 2 validation data. For EM/F1, we report means and standard deviations. The statistics were collected from 10 runs. # $\theta$  denotes the number of model’s parameters. We also measured EM/F1 for the extraction of antecedent/consequent separately; denoted as  $A_{EM}$ ,  $A_{F1}$  and  $C_{EM}$ ,  $C_{F1}$  respectively. At last  $ACC_{no-c}$  denotes no-consequent classification accuracy.

For **Subtask 2**, the results are presented in Table 2. The hyperparameters were the same for all LRMs. An ensemble was composed of 11 models drawn from the pool of 60 trained models. We found the ALBERT results to have a high variance. In fact, we recorded our overall best result on validation data with ALBERT, obtaining 75.35/89.00 EM/F1. However, in competition, we submitted only RoBERTa models due to less variance and slightly better results on average<sup>6</sup>.

## 5 Related work

Closest to our work, Son et al. (2017) created a counterfactual tweet dataset and built a pipeline classifier to detect counterfactuals. The authors identified 7 distinct categories of counterfactuals and firstly attempted to classify the examples into one of these categories using a set of rules. Then for certain categories, they used a linear SVM classifier (Cortes and Vapnik, 1995) to filter out tricky false positives.

A large effort in computational linguistics was devoted to the specific form of counterfactuals – so-called *what-if* questions. A recent paper by Tandon et al. (2019) presents a new dataset for what-if question answering, including a strong, BERT-based baseline. The task is to choose an answer to a hypothetical question about cause and an effect, e.g. *Do more wildfires result in more erosion by the ocean?*. Each question is accompanied by a paragraph focused on the topic of the question, which may or may not contain enough information to choose the correct option. The authors show that there is still a large

<sup>6</sup>We submitted the best ALBERT model in the post-evaluation challenge phase, obtaining worse test data results than the ensemble.

performance gap between humans and state-of-the-art models (73.8% accuracy for BERT against 96.3% for a human). This gap is caused mainly by the inability of the BERT model to answer more complicated questions based on indirect effects, which require more reasoning steps. However, the results show that the BERT model was able to answer a large portion of the questions even without accompanying paragraphs, indicating that the LRM models have a notion of commonsense knowledge.

## 6 Conclusions

We examined the performance of current state-of-the-art language representation models on both subtasks and we found yet another NLP task benefits from unsupervised pre-training. In both cases, we found RoBERTa model to perform slightly better than other LRMs, while its results also being more stable. We have ended up first in both EM and F1 on Subtask 2 and second in Subtask 1.

## Acknowledgements

This work was supported by the Czech Ministry of Education, Youth and Sports, subprogram INTERCOST, project code: LTC18006.

## References

- James Bergstra, Daniel Yamins, and David Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123.
- Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.
- 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.
- Martin Fajcik, Pavel Smrz, and Lukas Burget. 2019. But-fit at semeval-2019 task 7: Determining the rumour stance with pre-trained deep bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 1097–1104.
- Nelson Goodman. 1947. The problem of counterfactual conditionals. *The Journal of Philosophy*, 44(5):113–128.
- Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.
- 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*.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

- Youngseo Son, Anneke Buffone, Joe Raso, Allegra Larche, Anthony Janocko, Kevin Zembroski, H Andrew Schwartz, and Lyle Ungar. 2017. Recognizing counterfactual thinking in social media texts. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 654–658.
- William Starr. 2019. Counterfactuals. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition.
- Niket Tandon, Bhavana Dalvi Mishra, Keisuke Sakaguchi, Antoine Bosselut, and Peter Clark. 2019. Wiqua: A dataset for "what if..." reasoning over procedural text. In *EMNLP/IJCNLP*.
- Philip E Tetlock and Aaron Belkin. 1996. *Counterfactual thought experiments in world politics: Logical, methodological, and psychological perspectives*. Princeton University Press.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *EMNLP 2018*, page 353.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.
- 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.
- Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. 2019. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, pages 9593–9604.

## A Supplemental Material

### A.1 Hyperparameters

#### A.1.1 Subtask 1

The results of RoBERTa models with their training hyperparameters are presented in Table 3.

| batch size | learning rate | best acc            | best F1             |
|------------|---------------|---------------------|---------------------|
| 48         | 2.00E-05      | 0.9829943314        | 0.9209302326        |
| 70         | 1.00E-05      | 0.9823274425        | 0.9180834621        |
| 72         | 2.00E-05      | 0.9829943314        | 0.9199372057        |
| 90         | 4.00E-05      | 0.9829943314        | 0.9209302326        |
| 90         | 1.00E-05      | 0.9783261087        | 0.8992248062        |
| 96         | 3.00E-05      | <b>0.9839946649</b> | <b>0.9240506329</b> |
| 120        | 3.00E-05      | 0.9809936646        | 0.9107981221        |
| 132        | 4.00E-05      | 0.9796598866        | 0.9060092450        |

Table 3: Different batch sizes and learning rates used to train RoBERTa-large models, results of the best checkpoint on the validation part of the data.

We kept other RoBERTa model hyperparameters as shown in Table 4 for all training runs.

| Hyperparameter       | Value                      |
|----------------------|----------------------------|
| Max gradient norm    | 1.0                        |
| Epochs               | 8                          |
| Maximum input length | 100                        |
| Dropout              | 0.1                        |
| Optimizer            | Adam ( $\epsilon = 1e-8$ ) |

Table 4: Hyperparameters for Subtask 2, shared for all runs.

### A.1.2 Subtask 2

Our tuned hyperparameters are in Table 5. All other hyperparameters were left the same as PyTorch’s default. We did not use any learning rate scheduler.

| Hyperparameter            | Value    |
|---------------------------|----------|
| Dropout rate (last layer) | 0.0415   |
| Lookahead K               | 1.263e-5 |
| Lookahead $\alpha$        | 0.470    |
| Max gradient norm         | 7.739    |
| Batch size                | 64       |
| Weight Decay              | 0.02     |
| Patience                  | 5        |
| Max antecedent length     | 116      |
| Max consequent length     | 56       |

Table 5: Hyperparameters for Subtask 2. We tune only dropout at the last layer (the dropout mentioned in 2.3). Patience denotes the maximum number of epochs, after which we stop the training if there was no EM improvement. All parameters were tuned using HyperOpt (Bergstra et al., 2013).

### A.2 Data analysis

The distribution of lengths for examples from Subtask 1 is presented in Figure 2. We truncate sequences in this subtask to maximum of 100 tokens per example.

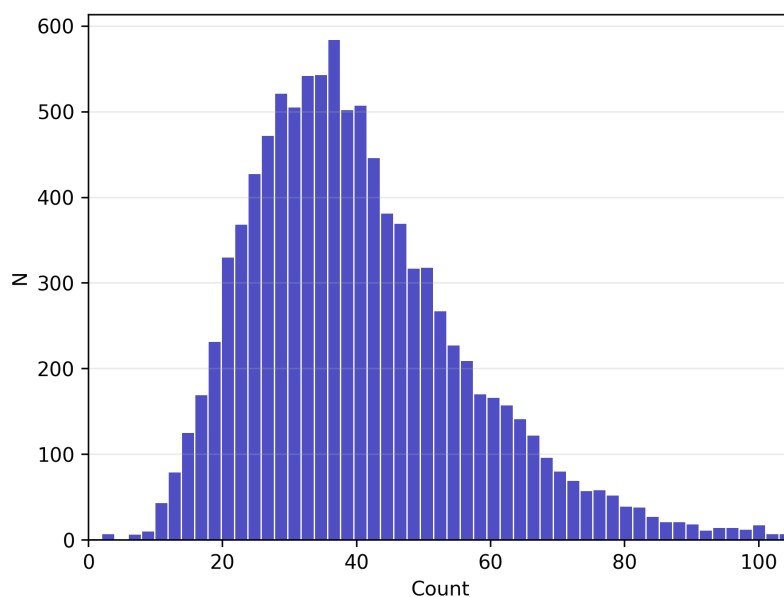


Figure 2: Histogram of example lengths in tokens in the training data for Subtask 1.

### A.3 Wrongly classified examples

Table 6 shows examples of statements classified wrongly by both ALBERT and RoBERTa models.

### A.4 Ambiguous labels

During the error analysis, we noticed a number of examples where we were not sure whether the labels are correct (see Table 7).

| Statement   | Predicted | Correct |
|---|-----------|---------|
| MAUREEN DOWD VISITS SECRETARY NAPOLITANO - "New Year's Resolutions: If only we could put America in Tupperware": "Janet Napolitano and I hadn't planned to spend New Year's Eve together. | 0         | 1       |
| If the current process fails, however, in hindsight some will say that it might have made more sense to outsource the whole effort to a commercial vendor.                                | 1         | 0       |

Table 6: Examples of wrong predictions.

| Statement   | Label |
|---|-------|
| Given that relatively few people have serious, undiagnosed arrhythmias with no symptoms (if people did, we would be screening for this more often), this isn't the major concern. | 0     |
| A flu shot will not always prevent you from getting flu, but most will have a less severe course of flu than if they hadn't had the shot," Dr. Morens said.                       | 0     |

Table 7: Examples of ambiguous annotation.

## A.5 Measurement of results

The individual measurements for Subtask 2 statistics presented in 2 can be found at <https://tinyurl.com/y8zncw7p>. Note that we did not use the same evaluation script as used in official baseline. Our evaluation script was SQuAD1.1 like, ground truth and extracted strings were firstly normalized the same way as in SQuAD1.1, then the strings were compared. For details see our implementation of method `evaluate_semeval2020_task5` in `scripts/common/evaluate.py`.

## A.6 Wrong predictions in Subtask 2

| Ground Truth   | Prediction   |
|--|--|
| GLOBAL FOOTPRINT Mylan said in a separate statement that the combination would create "a vertically and horizontally integrated generics and specialty pharmaceuticals leader with a diversified revenue base and a global footprint." On a pro forma basis, <b>the combined company would have had revenues of about \$4.2 billion</b> and a gross profit, or EBITDA, of about \$1.0 billion in 2006, Mylan said. | GLOBAL FOOTPRINT Mylan said in a separate statement that the combination would create "a vertically and horizontally integrated generics and specialty pharmaceuticals leader with a diversified revenue base and a global footprint." <b>On a pro forma basis, the combined company would have had revenues of about \$4.2 billion and a gross profit, or EBITDA, of about \$1.0 billion in 2006,</b> Mylan said. |
| <i>Shortly after the theater shooting in 2012, he told ABC that the gunman was "diabolical" and would have found another way to carry out his massacre if guns had not been available, a common argument from gun-control opponents.</i>   | Shortly after the theater shooting in 2012, he told ABC that <i>the gunman was "diabolical" and would have found another way to carry out his massacre if guns had not been available,</i> a common argument from gun-control opponents.   |
| Now, <b>if the priests in the Vatican had done their job in the first place, a quiet conversation, behind closed doors</b> and <i>much of it would have been prevented.</i>  | Now, <b>if the priests in the Vatican had done their job in the first place, a quiet conversation, behind closed doors and much of it would have been prevented.</b>   |
| The CPEC may have some advantages for Pakistan's economy – for one, it has helped address the country's chronic power shortage – but the costs are worrisome and <b>unless they can be wished away with a wand,</b> it will present significant issues in the future.  | The CPEC may have some advantages for Pakistan's economy – for one, it has helped address the country's chronic power shortage – but the costs are worrisome and <b>unless they can be wished away with a wand, it will present significant issues in the future.</b>  |

Table 8: An example of bad predictions from LRM over Subtask 2 validation data. Antecedents are highlighted with red bold, consequents with blue bold italic.