NLU-Co at SemEval-2020 Task 5: NLU/SVM based model apply to characterise and extract counterfactual items on raw data

MBONING TCHIAZE Elvis
ERTIM / 2 rue de Lille, Paris
elvis.mboning@inalco.fr

Damien Nouvel
ERTIM / 2 rue de Lille, Paris
damien.nouvel@inalco.fr

Abstract

In this article, we try to solve the problem of classification of counterfactual statements and extraction of antecedents/consequences in raw data, by mobilizing on one hand Support vector machine (SVMs) and on the other hand Natural Language Understanding (NLU) infrastructures available on the market for conversational agents. Our experiments allowed us to test different pipelines of two known platforms (Snips NLU and Rasa NLU). The results obtained show that a Rasa NLU pipeline, built with a well-preprocessed dataset and tuned algorithms, allows to model accurately the structure of a counterfactual event, in order to facilitate the identification and the extraction of its components.

1 Introduction

We can define counterfactuals as thoughts about alternatives to past events (factual), that is, thoughts of what might have been (counter). These counterfactual thoughts are preventive in nature because they help us better anticipate and guide our daily actions. Practically, counterfactual statements require a deep semantic analysis to identify them and extract their characteristic components (antecedent and consequent). Though Natural Language Understanding (NLU) is usually used in chatbots, digital personal assistants, or spoken dialogue systems (SDS), its techniques can also be applied in information retrieval/extraction and more specifically in the semantic classification of events (Catizone et al., 2008), (Wang and Zhang, 2011). This proposal is based on the starting baseline (Yang et al., 2020). We want to exploit the pipelines offered by two mainstream NLU libraries to classify and extract the consequent and antecedent segments occurring in a textual sequence of counterfactual type. For subtask 1, two systems are proposed: the first based on Support Vector Machine (SVM) and the second on NLU. For subtask 2, we only use NLU pipelines. The two NLU solutions are built on the basis of Snips NLU (Coucke et al., 2018) and Rasa NLU (Bocklisch et al., 2017).

2 Background: counterfactual thoughts and statements

The counterfactual problem is above all a phenomenon of human thinking and reasoning. It was widely studied quite early in several disciplines, mainly in philosophy with Aristotle, Plato, Leibniz (He describes the existence of several worlds gravitating around an event controlled by laws of logic.)(Epstude and Roese, 2008, p.1). As described by (McMullen et al., 1995), this thought can target reflections on the negativity of an event (Downward counterfactuals = ”what else could I have done to do worse ?”) or on its positivity (Upward counterfactuals = ”what else could I have done to do well ?”).

Kahneman and Miller proposed a `Norm theory` (Kahneman and Miller, 1986) which describes counterfactual as ”Unusual experiences tending to result in ‘if only’ thoughts that recapitulate the normal state of affairs.” (Epstude and Roese, 2008). Lewis (Lewis, 1973) on his side works on counterfactual laws that apply to causation, linked by conditions and dependencies between the events of the counterfactual relationship:

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 670–676
Barcelona, Spain (Online), December 12, 2020.

> “If A were the case, C would be the case” is true in the actual world if and only if either (i) there are no possible A-worlds; or (ii) some A-world where C holds is closer to the actual world than is any A-world where C does not hold. (Menzies and Beebee, 2020)

As we can see, the problem of counterfactuals is equally a problem of factual conditionals (Goodman, 1947, p.114). according to the functional theory defined by (Epstude and Roese, 2008), counterfactual thinking is a psychological process which makes it possible to solve problems. In this sense, if a problem arises, counterfactual thinking should have the effect of evoking behaviors that correct these problems.

Other studies arise on the responsibility that we have for a fact towards oneself or the others (self counterfactual), on the cognitive principles guiding the possibilities that people think of, when they imagine an alternative to reality, and on the rational counterfactuals (Marwala, 2014) which are those in the counterfactual universe (the possible worlds) corresponding to a given fact that maximizes a particular objective, according to the neuronal model (counterfactual machine). In other words, the theory of rational counterfactuals tries to identify the antecedent that gives the desired consequent.

These theories have been applied in several real-life problems (decision-making, patient depression), in several fields (engineering, human science, computer science, economics, medicine), the latter being the focus in this work. Concretely, in the next section 3, our question will be: how to mobilize the properties of rational counterfactuals in the SVM and NLU pipelines ?

Our laboratory worked on a project in the space field, aiming at detecting causalities in patents. The goal was to annotate a corpus of patents with causalities using active learning, then use the annotated data to implement a causality detector. This currently completed work allowed us to assess the difficulties for this kind of annotation, but also the possibilities to learn models that can partially detect explicit causality entities and relations.

3 Counterfactual statements classification based on SVM and NLU pipelines

For the experiments below, we used the initial data (csv file) with 13000 sentences which are organised as follows: 1454 counterfactual statements and 11546 uncounterfactual statements. We noticed at first glance that these data were unevenly represented on both sides, with a dominance of uncounterfactual statements at almost 89% over counterfactual statements. As it is and without standardization, the classification of the two classes will be unequal. To confirm this hypothesis, we carried out training sessions on one hand with all the data sets and on the other hand with standardized data sets (in equal proportions).

3.1 SVM experimental setup

Our first attempt to solve the classification problem (task 1) is based on a very simple architecture. We used the Scikit Learn library (Pedregosa et al., 2011) to vectorize texts and implement a classifier. We did some feature engineering and finally chose the following features:

- Frequency of tokens ngrams (1 to 3) (using Scikit Learn `CountVectorizer`)
- Length of the sentence and Number of punctuation
- Flair embeddings (Akbik et al., 2018)
- Absolute and relative (given the sentence size) number and position for each POS in IN MD VB VBD VBN VBP (POS are detected using NLTK)

All these features are provided to a SVM model (Scikit Learn `LinearSVC`. The only adjusted meta-parameter is the class weight, set to 0.008 value for uncounterfactual sentences and left to 1 for counterfactual sentences.

In our experiments, using the training set only with a 20% split for the test sets, our model obtains 70% f1-score (weighted average computed by Scikit Learn). Our submitted file on the challenge obtained only 66% f1-score. This might be due to the heterogeneity between train and test sets. For such a simple system we consider these results as acceptable, taking into account that we did not spend much time on this task, our main objective was to participate to task 2.

3.2 Snips and Rasa NLU experimental setup

In order to solve the problems of classification of counterfactual statements, we wanted to try other techniques different from those used in the baseline of subtask 1.

For this subclassification, we chose the NLU pipelines offered by two cutting-edge libraries: Snips NLU (Coucke et al., 2018) and Rasa NLU (Bocklisch et al., 2017). Our approach goes from the principle according to which the problem to be treated (in the practical sense of the term) is first and foremost a semantic problem[1]. To deal with semantic classification issues, NLU techniques are *prima facie* the first choice because it is centered on the understanding of the text (semantic understanding) in their context of use, while relying on the entities that it can contain. If it is true that this works very well for natural utterances useful for conversational agents, will it be the case for counterfactual statements ?

Our first concern was to find the accurate representation of statements which are counterfactual and those which are not. As we saw above, counterfactual statements are syntactically conditional statements: `if A (factual antecedent) occurs, then C (one possible world=consequent) could happen`. We created a regular expression that tries to extract the main elements of this structure. Table 1 below shows the matching counterfactual pattern with examples from training data.

<table>
<tr><td>Matching structure</td><td>(If|Since|Instead of|Neither|That|But|Also|Had|In|Unless)\s.+\s(would|could|should|I|may|might)+(’nt|’d|’m|’ve|still|be)?\s.+</td></tr>
<tr><td>Examples</td><td><ul><li>If only they had adhered to conservative principles...things would have been DIFFERENT</li><li>Instead of just relying on friendly assurances from Monsanto, Bayer should have insisted on examining all relevant details of the company’s glyphosate legal exposure, he said.</li><li>But if they had some kind of a protection inside the temple, ahhh, maybe it could have been a very much different situation.</li><li>Neither is in place right now, I would turn more bearish on global equities if they were.</li><li>...</li></ul></td></tr>
</table>

Table 1: Counterfactual matching structure and samples from training data (task1-train.csv)

3.2.1 Modeling data

Snips and Rasa NLU use different training data format (respectively `.yaml` and `.md` format). For each of them, we transformed the initial training data to match their format in order to train different NLU classification models. For Rasa training data, we used the counterfactual pattern as regex for features of RegexFeaturizer `(##regex:counterfactual)` cf. 3. As for Snips, we created virtual entities with this regex for each sentence. These virtual entities are supposed to help Snips learn better. In some other cases, this pattern matches uncounterfactual statements: “If you’ve heard of or read about mindfulness - a form of meditation - you might be curious about how to practice it.”.

[1] We associate it with a semantic problem because its properties (antecedents and consequents) are governed by connections or even dependency relationships around a network (continuous sequence of events), as is also the case for semantic connections between words in a language. One or many antecedents can be linked to one or many consequents, and this in a dependency scheme creates a counterfactual case

3.2.2 Modeling classification algorithms

Starting from a first assessment (Braun et al., 2017), Coucke and his colleagues (Coucke et al., 2018, p.14)[2] show that Snips utterance classification algorithm obtains good results compared to similar tools (Rasa, LUIS, Watson Conversation, API.ai, wit.ai, Amazon Lex). Snips uses a pipeline made up of two parsers: a deterministic parser (based on regular expressions) and a probabilistic parser (Logistic regression and Conditional Random Fields -CRFs-), hence why we started with Snips NLU. However, Snips does not allow the manual configuration of those two parsers. Once the formatted data is submitted for input, it takes care of all processes on its own and without any other intervention.

On the other hand, Rasa NLU is more open with respect to the choice of its pipelines internal components. Rasa NLU was used to test several pipelines ranging from the simplest, based on SVM (MITIE[3]), CRFs, to finish with complex pipelines based on deep learning, coupled with transformers features (BERT (Turc et al., 2019), ConveRT (Henderson et al., 2019)[4]).

To train our models, we applied the properties of table 2 to our data. To facilitate our experiments, we decided to divide the initial data (13,000 sentences) into two learning batches: train (1138 - 9232) and tests (286 - 2314). We were aware from the start of the poor distribution of the datasets between the two intents: the starting hypothesis was that the detection scores of the counterfactuals statements had to be significantly higher. Do these tools to overcome such a problem?

Data properties (2 intents)	**DIET properties**
Train set: (counterfactual=1138 / uncounterfactual=9232) Test set (counterfactual=286 / uncounterfactual=2314)	epochs=300 to 1000, learning_rate=0.0001, similarity_type=inner, embedding_dimension=50, number_of_transformer_layers=3, transformer_size=128, batch_strategy=balanced

Table 2: Data and algorithms properties used for each learning experiment

Id	**Pipeline names**	**Pipeline components**
1	Snips NLU	Deterministic Parser + Probabilistic parser
2	Rasa Spacy	SpacyNLP + SpacyTokenizer + SpacyFeaturizer + RegexFeaturizer + CountVectorsFeaturizer + LexicalSyntacticFeaturizer+ DIETClassifier
3	Rasa MITIE	MitieNLP + MitieTokenizer + MitieFeaturizer + RegexFeaturizer + MitieIntentClassifier
4	Rasa ConveRT DIET	ConveRTTokenizer + ConveRTFeaturizer + RegexFeaturizer + DIETClassifier
5	Rasa GPT DIET	HFTransformersNLP (GPT) + ConveRTTokenizer + ConveRTFeaturizer + RegexFeaturizer + DIETClassifier
6	Rasa BERT DIET	HFTransformersNLP (BERT) + ConveRTTokenizer + ConveRTFeaturizer + RegexFeaturizer + DIETClassifier

Table 3: NLU pipelines used for the subtask 1.

3.3 Results

Based on the pipelines described in table 3, table 4 summarizes the evaluations realised. The first thing we noticed is that ConveRT has very good features for classification tasks, even outside its original area (conversational data).

[2] https://github.com/snipsco/... and https://github.com/.../benchmarks.png

[3] https://github.com/mit-nlp/MITIE

[4] https://github.com/PolyAI-LDN/polyai-modelsconvert and GPT

Pipeline names	Precision	Recall	F1-score
Snips	0.57—0.95	0.67—0.92	0.61—0.94
Rasa Spacy	0.65—0.70	0.73—0.61	0.69—0.65
Rasa MITIE	0.70—0.69	0.67—0.72	0.69—0.70
Rasa ConveRT DIET	0.80—0.79	0.79—0.80	0.80—0.80
Rasa GPT DIET	0.77—0.74	0.72—0.79	0.75—0.76
Rasa BERT DIET	0.78—0.78	0.78—0.78	0.78—0.78

Table 4: Evaluation of NLU pipelines for the subtask 1

For each precision, recall and f1-score column in table 4, the first part describes the score of counterfactual statements and the other part, the score of uncounterfactual statements. We found out that most of the tested pipelines have troubles training a successful model with unbalanced learning data. In this task only the pipelines with transformers managed, with a large number of iterations, to have acceptable scores. Another experiment was carried out by balancing the learning data. It appears that with balanced data between the learning classes, all the pipelines reach the minimum score of 0.85 in f1-score.

The best model obtained here was applied to the leaderboard of the subtask 1 and we obtained f1-score=0.603, recall=0.575, precision=0.635, which are acceptable taking into consideration our own test in table 4.

4 Counterfactual components extraction based on Rasa and Snips pipelines

The aim of the subtask 2 was to detect antecedent and consequence in the counterfactual statements. The question was: which segment of the counterfactual statement can refer to antecedent and/or consequent following (Goodman, 1947)'s point of view ? Are all conditional structures counterfactual in nature ?

4.1 Experimental setup: Rasa and Snips NLU

Among the observations we made in subtask 1, we found out that Rasa Mitie and DEIT were the only ones able to meet our extraction needs. We chose only the best pipelines of the subtask 1 and updated the features as presented in table 5.

Id	Pipeline names	Pipeline components
1	Snips Entities	slot-filling (CRF)
3	Rasa MITIE	MitieNLP + MitieTokenizer + MitieFeaturizer + RegexFeaturizer + MitieEntityExtractor
4	Rasa ConveRT CRF	ConveRTTokenizer + ConveRTFeaturizer + RegexFeaturizer + CRFEntityExtractor
5	Rasa ConveRT DIET	ConveRTTokenizer + ConveRTFeaturizer + RegexFeaturizer + DIETClassifier (entity)
6	Rasa BERT DIET	HFTransformersNLP (BERT) + ConveRTTokenizer + ConveRT-Featurizer + RegexFeaturizer + DIETClassifier (entity)

Table 5: NLU pipelines used for the subtask 2.

4.2 Results

Even if in this case, the training data was balanced than the subtask 1, this subtask was particularly subjective (items segmentation) and required more training data. The results below were obtained using the training data only: training data (consequent=2285, antecedent=2734) and test data (consequent=573, antecedent=680). We used the basic configuration for Snips NLU and for Rasa DIET the best parameters obtained in subtask 1.

Pipeline names	Precision	Recall	F1-score
Snips Entities	0.07 — 0.04	0.05 — 0.02	0.06 — 0.03
Rasa MITIE	0.68 — 0.49	0.18 — 0.19	0.28 — 0.28
Rasa ConveRT CRF	0.63 — 0.37	0.12 — 0.03	0.20 — 0.06
Rasa ConveRT DIET	0.84 — 0.67	0.48 — 0.28	0.61 — 0.40
Rasa BERT DIET	0.86 — 0.74	0.42 — 0.25	0.56 — 0.37

Table 6: Evaluation of NLU pipelines for the subtask 1

For each precision, recall and f1-score column in table 6, the first part describes the score of antecedent and the other the consequent. We are quite surprised by the results of Snips on this simulation and the final test (`Evaluation` and `Post-evaluation`) data (f1-score=0.13, recall=0.34, precision=0.09). This can be explained by the impression of the cutting off of the antecedent segments and that of the consequent. When we look at the recognized entities, we can see that it manages to recognize the two segments but it is probably difficult to limit them at the right place. Further experiments and tests are needed to better understand this problem.

We also found out that the ConveRT transformers are very well suited to this task. The extraction of antecedents appears to be easier than that of consequents. Like in the case of Snips, these results become poor on the subtask test data. For the final evaluation (`Post-evaluation`), we create a new pipeline with the 5 best models obtained. These pipelines combine 3 Rasa Mitie models and 2 Snips NLU entity extractor models. We obtained in `Post-evaluation` the results (precision=0.36, recall=0.54, f1-score=0.39), which is more than `evaluation subtask1` (precision=0.06, recall=0.26, f1-score=0.08). This other experiment showed that the encapsulation of several NLU models of different types helps to better classify intentions.

Finally, these results make it possible to note that Snips NLU has difficulties in extracting the entities as they are, but succeed in extracting segments of the entity (antecedent or consequent). Take the example of: ”If enough night-time data had been collected, then even consumer-grade devices therefore may one day be able to offer preventive care.” Here Snips extracts ”had been collected” based on the real value from the list of antecedents ”if fees had been collected”. This problem is mainly due to Snips extraction model: the deterministic part prevails over the probabilistic part and this constitutes an important functional limit for extracting sequences from texts. The optimization of his internal language model and features could solve this problem.

We are well aware that we could have used oversampling and undersampling or pseudo-label (Lee, 2013) methods to try to solve the problem of intention class balancing. But, the urgency for us was to analyze the behavior of popular NLU algorithms on this type of data sets. We will do this for our future experiments.

5 Conclusion

The aim of this work was to classify counterfactual statements according to whether they are counterfactual or uncounterfactual. We participated in this challenges within the framework of a project on causality conducted by our team. The challenge itself aimed at extracting, within a few thousand counterfactual statements, the representative segments of antecedents on one hand and consequents on the other hand. To solve these problems, we opted for the use of SVM and NLU. We implemented a simple system for task 1, which obtained minimal results. The task 2 was much harder and we did not achieve a very good performance, but conducted some additional experiments. Although our results are not satisfactory from a competitive point of view, our experiments have shown the advantages and disadvantages of NLU pipelines, similar to the state of the art, generally used for conversational agents. We remain convinced that further modeling and new experiments will make it possible to adapt this problem of classifying and extracting counterfactual statements to NLU pipelines.

Acknowledgements

This work was experimented with CNES project based on causality detection and extraction of causal relations on patents raw data. All the code of our experiments is hosted at our Github repository: `https://github.com/Levis0045/NLU-Co_SemEval-Task5-2020`.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open Source Language Understanding and Dialogue Management. *arXiv:1712.05181 [cs]*, December. arXiv: 1712.05181.

Daniel Braun, Adrian Hernandez-Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating Natural Language Understanding Services for Conversational Question Answering Systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185, Saarbrücken, Germany. Association for Computational Linguistics.

Roberta Catizone, Alexiei Dingli, Hugo Pinto, and Yorick Wilks. 2008. Information extraction tools and methods for understanding dialogue in a companion. *European Language Resources Association (ELRA)*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, pages 12–16.

Kai Epstude and Neal J. Roese. 2008. The Functional Theory of Counterfactual Thinking. *Personality and social psychology review : an official journal of the Society for Personality and Social Psychology, Inc*, 12(2):168–192, May.

Nelson Goodman. 1947. The Problem of Counterfactual Conditionals. *The Journal of Philosophy*, 44(5):113, February.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien, and Ivan Vulić. 2019. ConveRT: Efficient and Accurate Conversational Representations from Transformers. *CoRR*, abs/1911.03688.

Daniel Kahneman and Dale T. Miller. 1986. Norm theory: Comparing reality to its alternatives. *Psychological Review*, 93(2):136–153.

Dong-Hyun Lee. 2013. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. page 6.

D. Lewis. 1973. Counterfactuals. *Oxford: Blackwell*, pages 1575–1578.

Tshilidzi Marwala. 2014. Rational counterfactuals. *CoRR*, abs/1404.2116.

Matthew N. McMullen, Keith D. Markman, and Igor Gavanski. 1995. Living in neither the best nor worst of all possible worlds: Antecedents and consequences of upward and downward counterfactual thinking. In *What might have been: The social psychology of counterfactual thinking*, pages 133–167. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, US.

Peter Menzies and Helen Beebee. 2020. Counterfactual Theories of Causation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2020 edition.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-Read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv preprint arXiv:1908.08962v2*.

X. Wang and X. Zhang. 2011. Research of web recruitment information extraction based on nlu. In *2011 International Conference on Computer Science and Service System (CSSS)*, pages 1575–1578.

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.