GBe at FinCausal 2020, Task 2: Span-based Causality Extraction for Financial Documents

Guillaume Becquin guillaume.becquin@gmail.com

Abstract

This document describes a system for causality extraction from financial documents submitted as part of the FinCausal 2020 Workshop. The main contribution of this paper is a description of the robust post-processing used to detect the number of cause and effect clauses in a document and extract them. The proposed system achieved a weighted-average F1 score of more than 95% for the official blind test set during the post-evaluation phase and exact clauses match for 83% of the documents.

1 Introduction

The FinCausal 2020 shared task (Mariko et al., 2020) focuses on the identification of causality in financial documents. Beyond factual data, causality detection and characterization in financial documents help to identify the reasons leading to a given quantified financial event. FinCausal 2020 proposes two shared tasks: the first one focuses on the detection of causality in documents (binary classification), while the second task consists of the actual extraction of the cause and effect spans from a document expressing causality (token classification or span extraction).

This document describes a system submitted for the second task. Based on a deep learning language model, it leverages an architecture similar to question answering and therefore addresses the task as a span extraction problem. The post-processing and filtering of unrealistic spans are of high importance given that several clauses (at least a cause and an effect) need to be extracted from the documents. The extracted clauses are further refined to align with the task labeling guidelines.

The proposed system ranked 2nd for the FinCausal 2020 Task 2, achieving a weighted-average F1 score of 94.7% and predicted exact matches for the cause and effect clauses for 73.7% of documents. Further training refinement led to an improved F1 score of 95% and exact match for 83.3% of the documents in the official blind test set in the post-evaluation phase. The code for the proposed solution is available at https://github.com/guillaume-be/Financial-Causality-Extraction

2 System Description

The core system used for all submissions is described in Figure 1. Features are generated from an input text using the Transformers library (Wolf et al., 2019) with high-performance tokenizers (Hugging Face, 2020). While the task dataset contains only rather short (2 to 3 sentences) texts, the system can handle longer documents exceeding the language model context using a sliding window mechanism. The tokenized input is then passed through a language model based on the Transformers (Vaswani et al., 2017) architecture. The last layer of hidden states is connected to a dense layer with 4 output dimensions to generate logits for the start and end positions of the cause and effect span. This mechanism is similar to Question Answering state-of-the-art architectures with two spans extracted instead of one.

The output of the model is an array of 4 vectors per document representing the logits for the start and end of the cause and effect. A first step cuts-off the logits vectors to the top-N elements, limiting the potential solution space to a size of N^4. Several of these solutions are unrealistic and must be

Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation, pages 40–44
Barcelona, Spain (Online), December 12, 2020.

filtered out. Some of these filtering steps are leveraged from the BERT (Devlin et al., 2019) Question Answering system for SQuAD (Rajpurkar et al., 2016). This includes for example filtering out start and end combinations where the end is before the start or the span length is longer than the maximum allowed length. The cause and effect extraction tasks add complexity in that two spans need to be extracted. Additional filters ensuring that the cause and effect clauses do not overlap have been added (for example $start_{cause} < start_{effect} < end_{cause}$ would be filtered out).

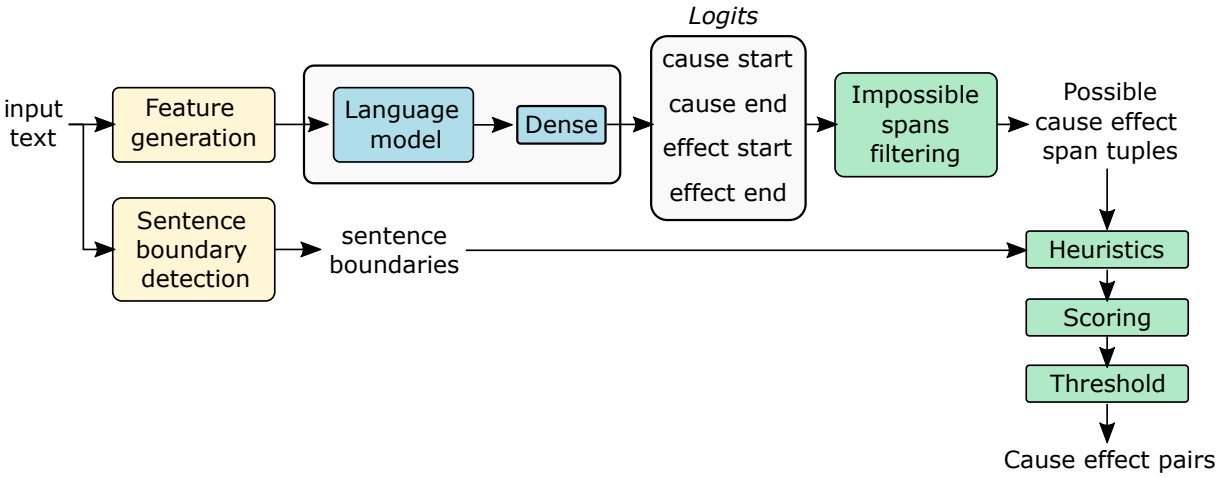

Figure 1: System architecture.

Finally, task-specific heuristics aligned with the labeling guidelines (Mariko et al., 2020) have been implemented. In addition to the filtered logits, these heuristics take the sentence boundaries as an input (these are extracted with the PySBD library (Sadvilkar and Neumann, 2020)). The following heuristics have been implemented:

- **H1**: A clause (cause or effect) may not span over multiple sentences.
- **H2**: If a sentence contains only one clause (cause or effect), extend the clause to the entire sentence

The filtered and improved (via heuristics) possible cause-effect span combinations are then ranked based on the sum of their 4 corresponding logits for start and end of cause and effect. The number of cause/effect combinations to return is either set externally (via the number of duplicated entries in the shared task dataset implicitly defining the number of cause/effects combinations to extract) or dynamically based on the span probability and a threshold.

3 Experimental Results

3.1 Data

The data used for all experiments is the data for the FinCausal 2020 shared task (Mariko et al., 2020). This data was extracted from 2019 financial news (QWAM, 2020). The task 2 dataset contains a subset of documents containing at least one cause and effect relationship. When several cause/effect relationships exist in a sample, this sample is duplicated for each additional cause/effect pair to be found. The data contains two labeled subsets called *Trial* (641 samples, 500 unique), *Practice* (1109 samples, 913 unique) and a blind test dataset *Evaluation* (638 samples, 452 unique). All models have been trained by merging the *Trial* and *Practice* dataset and performing a random split keeping 90% of the data for training and 10% for validation.

3.2 Experiments

The proposed system was trained for 20 epochs using the AdamW (Loshchilov and Hutter, 2017) optimizer. The learning rate is set to a maximum value of $2.5e-5$ and follows a cosine annealing schedule (Loshchilov and Hutter, 2016) with 100 warm-up steps. The effective batch size was kept constant at

12, using gradient accumulation when necessary. Experiments were performed on a RTX2070 GPU and Google Colaboratory notebooks.

Five different language models have been investigated: DistilBERT (Sanh et al., 2019), BERT-base and BERT-large (Devlin et al., 2019), RoBERTa-base and RoBERTa-large (Liu et al., 2019) using the implementation available in the Transformers library (Wolf et al., 2019). For each of these, the impact of pre-training on a question answering task (SQuAD dataset (Rajpurkar et al., 2016)) is evaluated. The rationale is that question answering is also a span extraction task and that the causality extraction architecture proposed is similar to that of extractive question answering systems. The impact of the language model and its pre-training setup is available in Figure 2. The average for three runs of the weighted-averaged F1 score over the token categories (cause, effect, other) and the fraction of documents with exact spans extractions are shown by the dots. The range represents the minimum and maximum values. The metrics are reported for the bind test dataset following an early stopping on the hold-out validation dataset.

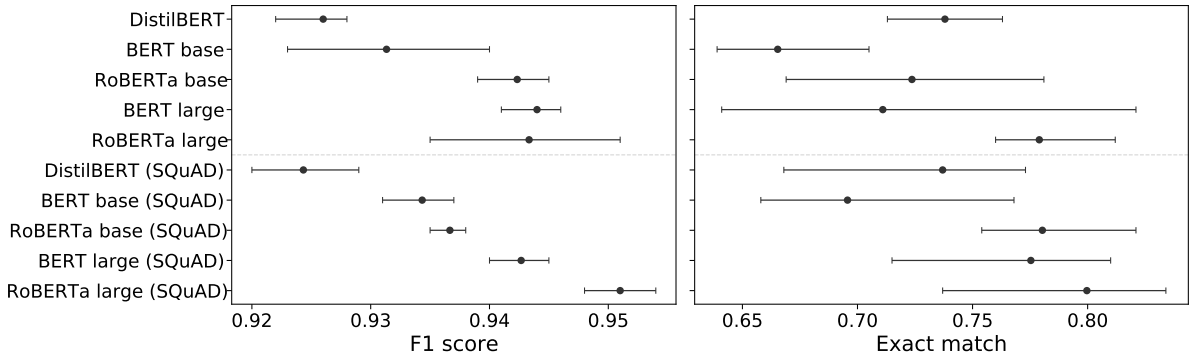

Figure 2: Language model selection and impact on performance.

The training of the system was more stable using models pre-trained for question answering: the weighted-averaged F1 score spread is significantly smaller for these models compared to the unsupervised pre-trained version. The larger models (BERT-large and RoBERTa-large) perform better than smaller models (base versions and DistilBERT) for a significantly higher computational cost. The impact of the model architecture and pre-training for the fraction of exact matches is less significant, probably because the metric is very sensitive to single token prediction changes.

3.3 Ablation study

An ablation study was conducted to identify significant contributions to the proposed model performance. For all ablations, the weighted-average F1 score and exact matches fraction on the blind test set is reported. The reported value is the average over 3 runs and subscripts show the standard deviation. The pre-training on a question answering task has a significant impact on both the weighted average F1 score and the standard deviation, indicating a more stable training regime. The question answering pre-training also improves the number of exact matches significantly, further highlighting the value of pre-training on a similar span extraction task.

Model	F1 score (%)	Exact match (%)	$\Delta_{F1}(\%)$
Full model	$\mathbf{94.68}_{0.3}$	$\mathbf{79.94}_{4.4}$	
without SQuAD pre-training	$93.92_{0.6}$	$77.90_{2.3}$	-0.76
Single cause/effect returned	$82.86_{0.2}$	$75.29_{0.1}$	-11.81
without Heuristic 1 (max. 1 sentence per clause)	$94.60_{0.1}$	$79.89_{4.4}$	-0.08
without Heuristic 2 (clause extended to full sentence)	$93.24_{0.1}$	$78.53_{2.8}$	-1.44

The *Single cause/effect return* represents a system that outputs a single cause and effects for all documents provided (instead of returning multiple feasible causes and effects). Since the dataset contains a

significant amount of documents with multiple causes and effects this severely impacts the performance. The impact of the first heuristic is marginal and within the uncertainty margins. The second heuristic, extending the clause to the entire sentence, has a significant impact (more than 1% point weighted-average F1 score). Additional training data would likely allow the model to learn more accurate span predictions and reduce the impact of this heuristic.

3.4 Multiple cause/effect predictions

A document may contain multiple cause/effect pairs to extract (for example 1 cause and 2 effects). For the shared task, the example is duplicated for each cause/effect pair, indicating the number of predictions to output. In a real-world setting the model should be able to assess how many predictions (causes and effects) to output for a given example. The following analysis evaluates if the value of the logits produced by the model can be used as a proxy for the confidence of the model for how many examples to return:

1. Rank possible extracted spans by the sum of the logits of start/end positions of the cause/effect
2. Filter out the spans with a cause or effect fully overlapping with more likely spans
3. Calculate the span probability ($P(cause_{start}) * P(cause_{end}) * P(effect_{start}) * P(effect_{end})$)
4. The ground truth of the number of spans (N) to extract is known from the number of duplicates of the given example. The top-N (excluding the first) extracted spans are stored as valid prediction probabilities, the rest of the extracted spans and their probability are stored as invalid predictions.

Figure 3 shows the results on the combined hold-out evaluation and blind test datasets:

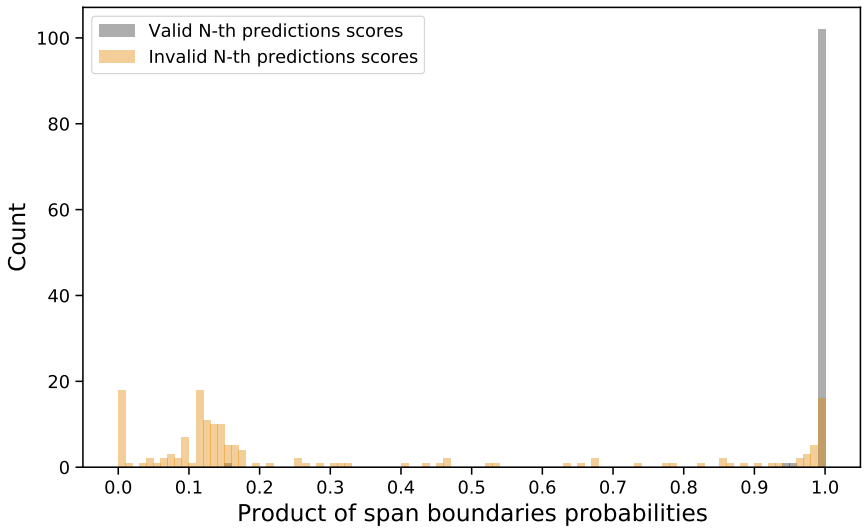

Figure 3: Probability of cause/effect prediction based on actual number of expected predictions.

The distribution of probability scores shows good separation of valid and invalid predictions. Using a confidence threshold of 99%, the model is able to identify the number of cause/effect relationships to extract with a F1 score of 91.4% (precision: 86.4%, recall: 97.1%).

4 Conclusion

A system for the second task of the FinCausal 2020 workshop has been described. The problem was framed as a span extraction task leveraging pre-trained model and post-processing elements from state-of-the-art question answering systems. This strong baseline was complemented with extensions to handle the extraction of two spans per document. The ability of the system to not only extract cause and effect, but also to predict the number of cause/effect relationships in a document was also demonstrated. The proposed model accurately extract causes and effect from documents.

References

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.

Hugging Face. 2020. tokenizers. `https://github.com/huggingface/tokenizers`.

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.

Ilya Loshchilov and Frank Hutter. 2016. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Dominique Mariko, Hanna Abi Akl, Estelle Labidurie, Stephane Durfort, Hugues de Mazancourt, and Mahmoud El-Haj. 2020. The Financial Document Causality Detection Shared Task (FinCausal 2020). In *The 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation (FNP-FNS 2020*, Barcelona, Spain.

QWAM. 2020. Qwam homepage.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Nipun Sadvilkar and M. Neumann. 2020. Pysbd: Pragmatic sentence boundary disambiguation. *ArXiv*, abs/2010.09657.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

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.