# NLP4ITF @ Causal News Corpus 2022: Leveraging Linguistic Information for Event Causality Classification

**Theresa Krumbiegel**
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg
theresa.krumbiegel@fkie.fraunhofer.de

**Sophie Decher**
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg
sophie.decher@fkie.fraunhofer.de

## Abstract

We present our submission to Subtask 1 of the CASE-2022 Shared Task 3: Event Causality Identification with Causal News Corpus as part of the 5th Workshop on Challenges and Applications of Automated Extraction of Sociopolitical Events from Text (CASE 2022) (Tan et al., 2022a). The task focuses on causal event classification on the sentence level and involves differentiating between sentences that include a cause-effect relation and sentences that do not. We approached this as a binary text classification task and experimented with multiple training sets augmented with additional linguistic information. Our best model was generated by training `roberta-base` on a combination of data from both Subtasks 1 and 2 with the addition of named entity annotations. During the development phase we achieved a macro F1 of 0.8641 with this model on the development set provided by the task organizers. When testing the model on the final test data, we achieved a macro F1 of 0.8516.

## 1 Introduction

Causal event classification can be categorized as a part of the Natural Language Processing (NLP) task of event extraction. When extracting event information from text, the general aim is to identify answers to the 5W1H questions (WHO, WHAT, WHEN, WHERE, WHY, HOW; Karaman et al., 2017). Some of the questions can be answered easily by means of open source NLP tools–a Named Entity Tagger can facilitate the extraction of locations (WHERE) and times or dates (WHEN), for example. However, some event information remains more difficult to identify reliably in texts, such as answers to WHY questions, which is also the type of question that causal event classification addresses. This task presents an opportunity to develop models that detect information about the reason behind a particular event. For this process, a binary classifier is used to determine whether a cause-effect relation is present in the input sentence. In an NLP pipeline, the output of such a classification process is often used as input for a span detection system, which identifies the particular cause and effect text spans in each causal sentence.

As described by Tan et al. (2022b), causality can be expressed either explicitly or implicitly. The authors illustrate this by providing the following examples:

(1) The treating doctors said Sangram lost around 5 kg due to the hunger strike.

(2) Dissatisfied with the package, workers staged an all-night sit-in.

Example 1 displays explicit causality, made apparent by the presence of the causal marker "due to". The organizers of the current shared task also refer to this marker as the *signal*. In contrast, the causal relation between the sit-in and worker dissatisfaction in Example 2 is implicit, as the sentence does not contain a causal marker.

In their survey of causal relation extraction in natural language texts, Yang et al. (2022) emphasize the potential of domain-specific pre-trained models in combination with graph-based models. They also stress the importance of leveraging linguistic information in order to identify both implicit and explicit causal relations. For this reason, the current study focuses primarily on experiments regarding the integration of linguistic information in the training data, to be used as input for the fine-tuning of pre-trained transformer models.

The remainder of this paper is structured as follows: Section 2 introduces the shared task and the dataset. In Section 3, we describe the training process, model configuration details, and the linguistic dataset alterations that we tested. Results are presented in Section 4 and discussed in Section 5, followed by concluding remarks and a summary of our findings in Section 6.

*Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE)*, pages 16 - 20
December 7-8, 2022 ©2022 Association for Computational Linguistics

## 2 Dataset and Task

The CASE-2022 Shared Task 3 on Event Causality Identification is divided into two subtasks. The data provided by the organizers stems from the Causal News Corpus, a collection of 3,559 English annotated event sentences from 869 news articles about protests (Tan et al., 2022b). The goal of Subtask 1 is to determine whether an event sentence contains a cause-effect relation. Subtask 2 is concerned with identifying the spans that correspond to cause, effect, or signal in each causal sentence. We developed and submitted models for the first of these two subtasks.

For the development phase, the task organizers provided a training dataset consisting of 2925 training instances. Sentences with the label 0 ($n = 1322$) did not contain a causal relation, while sentences with the label 1 included a causal relation and were in the majority ($n = 1603$). In addition, an unlabeled development set of 323 sentences was made available in order to allow for model testing via the CodaLab submission portal.

Preliminary exploratory analysis of the data provided for the development phase revealed an average inter-annotator agreement of 88.27% for causal sentences, while sentences labeled as containing no causal relation had an average agreement of 77.89%. Between 1 and 3 annotators labeled each sentence, coming to a consensus of 100% agreement for 70.31% ($n = 1127$) of the causal sentences but only 47.35% ($n = 626$) of the non-causal sentences.

For the test phase, the previously unlabeled development set was re-released with annotations so that it could be used as additional training data. An unlabeled test set of 311 previously unseen sentences was made available for the final testing and scoring process.

## 3 Methodology

We fine-tuned pre-trained language models (PLMs) on the training data and adjusted the model hyperparameters accordingly. We then tested four different methods of augmenting the training data with linguistic information and compared their efficacy.

### 3.1 Model settings

We used the Flair framework (Akbik et al., 2019) for model configuration and training. For the development phase, the original data was shuffled and divided into train, validate, and test sets (80/10/10). Using the `roberta-base` and `bert-base-cased` PLMs for comparison, we applied document embeddings to each sentence and fine-tuned the learning rate and batch size hyperparameters (Devlin et al., 2018; Liu et al., 2019). Weights were assigned to the different classes during training to account for the unbalanced distribution in the data with the help of the Scikit-learn `class_weight` parameter (Pedregosa et al., 2011). As the negative class was slightly underrepresented, it was assigned a proportionally higher weight.

### 3.2 Data Manipulation

In addition to adjusting model settings, we experimented with manipulating the model input and adding pertinent linguistic information during the development phase of the shared task. During the final testing phase, we retrained and tested our best model again using the additional data provided by the organizers. Regardless of the training data used, the test instances were always in the form of individual sentences, with no additional information added.

**Baseline dataset** In order to have a baseline for comparison, we used an unchanged version of the training data to fine-tune both the BERT and RoBERTa PLMs. This data consisted of individual sentences and corresponding binary labels (cf. Example 3).

(3) Some protesters attempted to fight back with fire extinguishers. 0

**Flair NER** We used the standard 4-class Flair NER model (pre-trained on the English CoNLL-03 task) to identify named entities of type **Person** (PER), **Location** (LOC), **Organization** (ORG), and **Miscellaneous** (MISC) in the training data, creating new training sets that contained all possible combinations of the four named entity classes. The identified text spans were replaced with the appropriate named entity tag (cf. Example 4).

(4) On Monday, the African National Congress condemned the shooting of Malunga, the Oshabeni branch chairman, and Chiliza, the branch secretary.

On Monday, the ORG condemned the shooting of PER, the LOC branch chairmain, and PER, the branch secretary.

**AllenNLP** The AllenNLP library was used to annotate the original training data with the semantic role labels ARG0 (proto-agent), ARG1 (proto-patient) and ARGM-CAU (cause clause) (Shi and Lin, 2019). The annotations were then added to the sentences as illustrated in Example 5.

(5) [ARG0: Mainland authorities] have launched [ARG1: a massive crackdown against terrorism] [ARGM-CAU: in wake of a string of violent attacks in the restive Xinjiang region and other cities on the mainland].

The starting point for the semantic role annotations was always the root of the sentence (e.g. the word "launched" in Example 5), which was determined with the help of the spaCy English dependency parser (Honnibal and Montani, 2017). The inclusion of explicit annotations for cause clauses in the training data seemed promising in the context of the given task. However, the AllenNLP model was only able to identify cause clauses in 1.6% of the training instances. We suspect that the model fails primarily at recognizing cause clauses in sentences that contain cause-effect relations only implicitly. Due to the small amount of annotations, we determined that this feature was not meaningful enough to improve classifier performance.

**Cause-Effect-Signal Spans** A further training dataset was created by adding information from the data provided for Subtask 2, which was identical to the Subtask 1 data, with the addition of Cause-Effect-Signal (CES) span annotations. All sentences from the negative class in the Subtask 1 data were added to the new training set without any annotations.

(6) <ARG1>They then decided to call off the protest</ARG1> <SIG0>as</SIG0> <ARG0>the police had ceded to their demand</ARG0> .

Sentences from the positive class were replaced with the corresponding annotated version from the Subtask 2 data (cf. Example 6). If the Subtask 2 data listed more than one possible annotation option for a sentence, the first option was selected.

**NER & Cause-Effect-Signal Spans** After creating the training set with Cause-Effect-Signal span annotations, we also used the 4-class Flair NER model to identify named entities and replaced the named entity text spans with the corresponding label (PER, LOC, ORG, MISC) in all training instances.

(7) <ARG1>Police took into custody fifteen activists</ARG1> <SIG0>for</SIG0> <ARG0>blocking the traffic in LOC</ARG0>.

Example 7 shows a training instance from the positive class containing both NER and Cause-Effect-Signal annotations. We experimented by including all possible combinations of named entity classes during training.

## 4 Results

### 4.1 Development phase

Models trained using `roberta-base` outperformed those trained with `bert-base-cased`. For this reason, we choose to focus on the models trained with the former architecture in the following pages. Regardless of the data used, the following hyperparameters worked best for all models: a batch size of 8, a learning rate of 3e-5, and the ADAM optimizer. The maximum number of epochs was set to 20, but training was terminated early if it became obvious that the model was overfitting the data, which could be observed as early as epoch 3.

Three of the four methods for adding linguistic information to the model input positively affected model performance: 1) Flair NER annotations; 2) Cause-Effect-Signal spans from the Subtask 2 data; or 3) a combination of both NER and Cause-Effect-Signal spans. When training models with data containing Flair NER annotations, we found that including only the PER and LOC classes (RoBERTa+PER+LOC) resulted in the best performance. When the training data contained both Cause-Effect-Signal spans and Flair NER classes, however, performance was better when only the PER class was included (RoBERTa+PER+CES).

The best performing model on the development set provided by the organizers was RoBERTa+PER+LOC with a macro F1 of 0.8802 (cf. Table 1). However, performance was inconsistent. When we tested the model on our self-

| Model configuration | Precision | Recall | Macro F1 |
|---|---|---|---|
| RoBERTa baseline | 0.8256 | 0.9045 | 0.8633 |
| RoBERTa+PER+LOC | 0.8729 | 0.8876 | 0.8802* |
| RoBERTa+CES | 0.8571 | 0.8427 | 0.8499 |
| RoBERTa+PER+CES | 0.8368 | 0.8933 | 0.8641 |

Table 1: Results of development phase scoring. Best performing model is marked with *.

| Model configuration | Precision | Recall | Macro F1 |
|---|---|---|---|
| BERT baseline (Tan et al., 2022a) | 0.7801 | 0.8466 | 0.8120 |
| LSTM baseline (Tan et al., 2022a) | 0.7268 | 0.8466 | 0.7822 |
| RoBERTa baseline | 0.8000 | 0.9091 | 0.8511 |
| RoBERTa+PER+LOC | 0.7914 | 0.8409 | 0.8154 |
| RoBERTa+CES | 0.8239 | 0.8239 | 0.8239 |
| RoBERTa+PER+CES | 0.8245 | 0.8807 | 0.8516* |
| RoBERTa+PER+CES+FullDataset | 0.8343 | 0.8580 | 0.8459 |

Table 2: Results of final test phase scoring. Best performing model is marked with *.

compiled test set during the training phase, the macro F1 score peaked at 0.8578, leading us to question the robustness of the model.

The RoBERTa baseline and the RoBERTa+PER+CES models performed similarly (macro F1 scores of 0.8633 and 0.8641, respectively) with regard to the development set and exhibited more robustness, i.e. the variance between development and self-compiled test set was comparatively small. The RoBERTa+CES model scored slightly lower than the other models with a macro F1 of 0.8499.

### 4.2 Test phase

Table 2 shows our results from the final testing phase of the shared task, as well as the baselines provided by the organizers. Hyperparameter settings used for development were kept constant for the final testing phase, as were the training datasets, with the exception of RoBERTa+PER+CES+FullDataset. This model was trained using the additional labeled data provided by the organizers for the final testing phase.

Models trained during the development phase consistently achieved lower macro F1 scores on the final testing data. The best model from the development phase (RoBERTa+PER+LOC) performed poorly with a macro F1 of 0.8154, supporting the idea that the model was not robust. The RoBERTa+PER+CES model achieved the highest macro F1 score of 0.8516, outperforming the RoBERTa baseline model by only 0.0005. Surprisingly, re-training this best model with the additional training data provided by the organizers did not improve model performance, resulting in a macro F1 score of only 0.8459.

## 5 Discussion

The discrepancies between the scores for the development and final testing phases call for a closer investigation of the model input and output. The results from the development phase suggest that model performance increases when training data includes linguistic information in the form of 1) named entity annotations for the PER and LOC classes, or 2) as a combination of both PER named entity annotations and Cause-Effect-Signal spans. Adding only Cause-Effect-Signal spans, however, appears to have had a negative impact on model test scores.

The fact that the RoBERTa+PER+LOC model outperformed the RoBERTa+PER+CES model also suggests that named entity information may prove more useful than Cause-Effect-Signal spans. It is frequently the case that named entities of type PER, i.e. proper nouns, have the semantic role of AGENT or PATIENT in a sentence. Replacing these nouns, along with location names, with named entity tags distills this important information and reduces the number of superfluous words in the data. We suggest that this creates a clearer pattern for the model, which in turn improves performance.

In the final test phase, however, only RoBERTa+PER+CES outperformed our established RoBERTa baseline by a small margin, while RoBERTa+PER+LOC and RoBERTa+CES had the lowest macro F1 scores. According to these results, it seems that adding linguistic information to the training data in the form of named entity annotations or Cause-Effect-Signal spans only leads to minute increases in model performance. It may be that our RoBERTa baseline model is able to extract this particular linguistic information on its own without the need for additional feature engineering. Further experimentation with linguistic features is needed in this area.

With only 2925 sentences, the size of the original amount of training data is also a potential factor that affected model performance. More training data would most likely increase model performance.

A closer investigation of the data revealed that some annotations also leave room for discussion, such as the sentence in Example 8:

(8) The house of a PDP MP was torched in south Kashmir. 1

The sentence is labeled as belonging to the positive class, but we were unable to identify a cause-effect relation. This shows that identifying causality can pose difficulties for expert human annotators. Such instances may negatively influence the detection of causal patterns during training.

Interestingly, re-training our best model with the additional training data in the final testing phase did not improve performance. Furthermore, the testing data used for evaluation in the development phase appears to consist of sentences from only two news articles. The data basis for development was accordingly very homogeneous and most likely did not provide an accurate representation of all possible articles that the model might need to classify in a real-world application. Optimization based on homogeneous data can lead to a preference for models that work well with that specific data but fail to generalize to more diverse data. The difference in model performance between the development and test phases might be evidence of this phenomenon.

## 6 Conclusion

Training data—including the source, domain, amount, and any added features—plays an important role when it comes to model optimization for NLP tasks, and the subfield of event causality is no exception. Our findings show that the generalizability of a model depends heavily on the quality and content of the model input. In our case, adding linguistic information to the training data only led to a minute increase in model performance as compared to our established RoBERTa baseline. It is possible that a larger training dataset would improve results. In addition, a larger, more diverse testing dataset is necessary in order to adequately evaluate the robustness of the model and predict its effectiveness for real-world applications. Future directions might also include a greater focus on strategies for the identification of implicit cause-effect relations.

## References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Çağla Çığ Karaman, Serkan Yalıman, and Salih Atılay Oto. 2017. Event detection from social media: 5W1H analysis on big data. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.

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.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Fiona Anting Tan, Ali Hürriyetoğlu, Tommaso Caselli, Nelleke Oostdijk, Hansi Hettiarachchi, Tadashi Nomoto, Onur Uca, and Farhana Ferdousi Liza. 2022a. Event causality identification with Causal News Corpus - Shared Task 3, CASE 2022. In *Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2022)*, Online. Association for Computational Linguistics.

Fiona Anting Tan, Ali Hürriyetoğlu, Tommaso Caselli, Nelleke Oostdijk, Tadashi Nomoto, Hansi Hettiarachchi, Iqra Ameer, Onur Uca, Farhana Ferdousi Liza, and Tiancheng Hu. 2022b. The causal news corpus: Annotating causal relations in event sentences from news. In *Proceedings of the Language Resources and Evaluation Conference*, pages 2298–2310, Marseille, France. European Language Resources Association.

Jie Yang, Soyeon Caren Han, and Josiah Poon. 2022. A survey on extraction of causal relations from natural language text. *CoRR*, arXiv:2101.06426.