IDIAPers @ Causal News Corpus 2022: Efficient Causal Relation Identification Through a Prompt-based Few-shot Approach

Sergio Burdisso[*,1,2], **Juan Zuluaga-Gomez**[1,3], **Esaú Villatoro-Tello**[1,5], **Martin Fajcik**[1,4]
Muskaan Singh[1], **Pavel Smrz**[4], **Petr Motlicek**[1]
[1]Idiap Research Institute, Martigny, Switzerland
[2]Universidad Nacional de San Luis (UNSL), San Luis, Argentina
[3]Ecole Polytechnique Fédérale de Lausanne, Switzerland
[4]Brno University of Technology, Brno, Czech Republic
[5]Universidad Autónoma Metropolitana Unidad Cuajimalpa, Mexico City, Mexico
[*]*corresponding author: sergio.burdisso@idiap.ch*

Abstract

In this paper, we describe our participation in the subtask 1 of CASE-2022, Event Causality Identification with Casual News Corpus. We address the Causal Relation Identification (CRI) task by exploiting a set of simple yet complementary techniques for fine-tuning language models (LMs) on a small number of annotated examples (i.e., a *few-shot* configuration). We follow a prompt-based prediction approach for fine-tuning LMs in which the CRI task is treated as a masked language modeling problem (MLM). This approach allows LMs natively pre-trained on MLM problems to directly generate textual responses to CRI-specific prompts. We compare the performance of this method against ensemble techniques trained on the entire dataset. Our best-performing submission was fine-tuned with only 256 instances per class, 15.7% of the all available data, and yet obtained the second-best precision (0.82), third-best accuracy (0.82), and an F1-score (0.85) very close to what was reported by the winner team (0.86).[1]

1 Introduction

Causal relation identification aims to predict whether or not there exists a cause-effect relation between a pair of events mentioned in a given text. For example, in the sentence *"Protests spread to 15 towns and resulted in the destruction of property"*, the automatic causal identification system must be able to realize that there is cause-effect relation between the events *"protest"* and *"destruction"*.

Hence, understanding causal relations within a text is an essential aspect of natural language processing (NLP) and understanding (NLU) (Ayyanar et al., 2019a; Li et al., 2021; Tan et al., 2022c). Once the causal information is identified within a text, such knowledge becomes beneficial for many other downstream NLP tasks, e.g., Information Extraction, Question Answering, Text Summarization (Ayyanar et al., 2019a; Man et al., 2022). However, due to the ambiguity and diversity in written documents, causality identification is not easy and remains a challenging problem.

The Event Causality Identification with Causal News Corpus (CASE-2022) shared task (Tan et al., 2022b) addresses this problem on a recently created corpus named the Causal News Corpus (CNC) (Tan et al., 2022a). Contrary to previous existing causality corpora, the CNC dataset, manually annotated by experts, incorporates a broader set of causal linguistic constructions, i.e., not only limited to explicit constructions, resulting in a more challenging dataset.

In this paper, we describe our followed methodology for addressing the causal event classification shared task (subtask 1) during the CASE-2022 competition (Tan et al., 2022b).[2] Our primary method, based on a *few-shot* configuration, follows a prompt-based approach for fine-tuning the language model (LM). The intuitive idea of this approach is to allow the LM to directly auto-complete natural language prompts. Following this technique, we leverage the LM's knowledge and let it decide the correct label of the input sequence. Additionally, we evaluate the performance of ensemble techniques trained using the entire dataset available. Our results demonstrate that our few-shot, prompt-based, fine-tuning approach can generalize well even when using as few as 256 samples per class for training, outperforming ensemble techniques trained with the entire dataset, as well as most of other teams' submissions.

The rest of the paper is organized as follows.

[1] Code available at https://github.com/idiap/cncsharedtask.

[2] We refer the reader to our standalone publication (Fajcik et al., 2022) to know our results for subtask 2.

Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE), pages 61 - 69
December 7-8, 2022 ©2022 Association for Computational Linguistics

Section 2 describes relevant related work, Section 3 describes the components of our main method, namely the prompt-based approach. Section 4 describes the experimental setup, i.e., datasets, additional baselines, experiments configuration and obtained results. Finally, Section 5 depicts our main conclusions and future work directions.

2 Related Work

Previous work on causal relation identification varies from knowledge-based to deep neural network approaches (Deep-NN). Knowledge-based systems rely on linguistic patterns extracted using an exhaustive exploration of the data, where lexico-semantic and syntactic analysis lead to the identification of relevant structures and keywords that depict the presence of a causal relation in the text (Garcia, 1997; Khoo et al., 2000). Although interpretable, these methods require a lot of human effort to generate relevant patterns and result in models that are not readily applicable in different domains.

Statistical machine learning (ML) approaches leave to the selected algorithm to find patterns in the data on the basis of the manual annotation. Traditionally, using different NLP tools, it is possible to compute various features for a given collection and apply any ML pipeline to train a causality relation classifier, e.g., (Rutherford and Xue, 2014; Hidey and McKeown, 2016). However, one main disadvantage of these techniques is the language dependency and error propagation of the NLP tools, e.g., syntactic parsers.

Finally, recent approaches based on Deep-NN have become popular, given their powerful representation learning ability. Typical approaches include convolutional neural networks (Ayyanar et al., 2019b), long short-term memory networks (Li et al., 2021), and pre-trained transformer-based LMs such as BERT (Devlin et al., 2019), where following a standard fine-tuning approach makes possible the detection of causality relations (Tan et al., 2022c; Khetan et al., 2022; Fajcik et al., 2020). Normally, these methods involve high computational costs and large amounts of labeled data. However, in this work, we show that pre-trained LMs can still be effective even when fine-tuned with very few instances.

Contrary to previous work, we evaluate the effectiveness of very recent prompt-based prediction approaches under a *few-shot* configuration for causal relation identification.

3 Prompt-Based Approach

In the "pre-train, prompt, and predict" paradigm, unlike the standard "pre-train and fine-tune" paradigm, instead of adapting pre-trained LMs to downstream tasks via objective engineering,[3] downstream tasks are reformulated to look more like those solved during the LM pre-training phase (Liu et al., 2021). More precisely, prompt-based prediction treats the downstream task as a masked language modeling problem, where the model directly generates a textual response (referred to as a *label word*) to a given prompt defined by a task-specific *template* (Gao et al., 2021). For instance, when identifying the sentiment of a movie review like "I love this movie." we may continue with "Overall, it was a `[MASK]` movie." and ask the LM to fill the mask with a sentiment-bearing word. In this example, the original input text x ("I love this movie.") is modified using the *template* "`[x]` Overall, it was a `[MASK]` movie." into a textual string prompt x' in which the mask will be filled with a *label word*. Some examples of *label words* for this example could be "fantastic" or "boring".

In the case of classification tasks, in addition to defining a set of possible *label words*, it is necessary to define a mapping between each one and the actual output labels. For instance, if labels $+$ and $-$ refer to positive and negative sentiment, respectively, "fantastic" in previous example could be mapped to output label $+$, and "boring" to $-$.

Formally, let $\mathcal{L}$ be a pre-trained language model, $f_t(x)$ a function that converts the input x into a prompt by instantiating template t which contains one `[MASK]` token, $mask$. Let $word : \mathcal{Y} \rightarrow \mathcal{W}$ be a mapping from the task label space, $\mathcal{Y}$, to the *label words* set, $\mathcal{W}$. Then, the classification task is converted to a *masked language modeling* (MLM) task in which the probability of predicting class $y \in \mathcal{Y}$ is modeled as:

$$\begin{aligned} p(y|x) &= p(mask = word(y)|f_t(x)) = \\ &= \frac{exp(\mathbf{w}_{word(y)} \cdot \mathbf{h}_{mask})}{\sum_{y' \in \mathcal{Y}} exp(\mathbf{w}_{word(y')} \cdot \mathbf{h}_{mask})}, \end{aligned} \tag{1}$$

where $\mathbf{h}_{mask}$ is the hidden vector of `[MASK]` and $\mathbf{w}_v$ denotes the vector encoding word v. Note that

[3] *Objective engineering* referes to both the pre-training and fine-tuning stages of LMs (Liu et al., 2021).

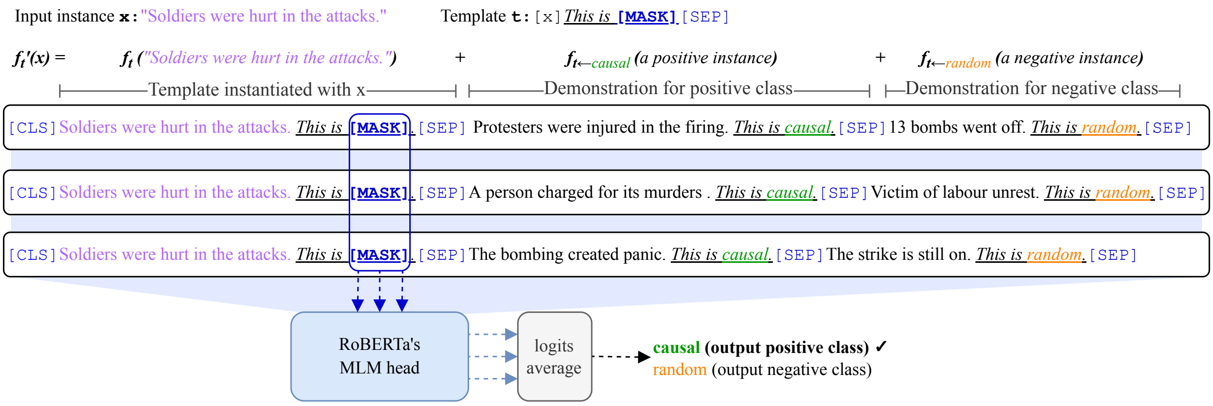
Input instance x: "Soldiers were hurt in the attacks."
Template t: [x] This is [MASK]. [SEP]
f_t'(x) = f_t ("Soldiers were hurt in the attacks.") + f_t←causal (a positive instance) + f_t←random (a negative instance)
Template instantiated with x
Demonstration for positive class
Demonstration for negative class
[CLS] Soldiers were hurt in the attacks. This is [MASK]. [SEP] Protesters were injured in the firing. This is causal. [SEP] 13 bombs went off. This is random. [SEP]
[CLS] Soldiers were hurt in the attacks. This is [MASK]. [SEP] A person charged for its murders . This is causal. [SEP] Victim of labour unrest. This is random. [SEP]
[CLS] Soldiers were hurt in the attacks. This is [MASK]. [SEP] The bombing created panic. This is causal. [SEP] The strike is still on. This is random. [SEP]
RoBERTa's MLM head
logits average
causal (output positive class) ✓
random (output negative class)

Figure 1: Augmented prompt-based classification for causality identification task. First, the input instance x = *"Soldiers were hurt in the attacks"* is converted into three different input prompts by applying $f'_t(x)$ three times. Then, these three prompts are given to a RoBERTa model, and one logit vector is obtained for each. These vectors are then averaged, and the word with the highest score, *"causal"*, is selected. Finally, this word is mapped to its corresponding class, and x is classified as positive. Note that, in this example, we have the following word-to-class label mapping $word(positive)$ = *"causal"* and $word(negative)$ = *"random"*.

when fine-tuning $\mathcal{L}$ to minimize the cross-entropy loss, the pre-trained weights $\mathbf{w}_v$ are re-used, and there's no need to introduce any new parameter. On the contrary, with standard fine-tuning a task-specific head, $softmax(\mathbf{W}_o\mathbf{h}_{\texttt{[CLS]}})$, has to be added, with new task-specific learnable parameters $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{Y}|\times d}$, which increases the gap between pre-training and fine-tuning.

Hereafter we will refer to the "causal" and "non-causal" classes as "positive" (+) and "negative" (−) respectively. In addition, and following previous work by Gao et al. (2021), we append one answered prompt for each class to the input prompt as *demonstrations*.[4] More precisely, let $\mathcal{Y} = \{+, -\}$ be the set of labels for the binary causality identification task, let $t \leftarrow v$ be the template t in which its [MASK] token has been filled with word v, and $w^y = word(y)$ the *word label* for class $y \in \mathcal{Y}$, then we redefine $f_t(x)$ in Equation 1 as $f'_t(x)$ defined as:

$$f'_t(x) = f_t(x) \parallel f_{t\leftarrow w^+}(x^+) \parallel f_{t\leftarrow w^-}(x^-) \quad (2)$$

where $\parallel$ is the string concatenation operator, and x^y is an instance of class y randomly sampled from the training set. Figure 1, depicts an example of three different input prompts are shown by applying $f'_t(x)$ three times to the input instance x.

Classification process: the process is illustrated in Figure 1. First, the input instance x is converted into d different input prompts by applying $f'_t(x)$, d times. Then, each input prompt is given to the LM to obtain d logit vectors holding the word scores for the mask in each prompt. A simple ensemble scheme is then applied by averaging all d logit vectors, and the *word label* with the highest score is selected, which is finally mapped to its corresponding class y using mapping $word(y)$.

Training and model selection: for developing our prompt-based models, we performed a simplified version of the process described in previous work by Gao et al. (2021). Namely, we carried out the following six steps:

Step 1: we created a new training set, τ_k, by extracting k instances per class from the original train partition, and used the remaining $2925 - 2 \times k$ instances as a large evaluation set δ_{T-k} (dataset stats are given in Table 2).

Step 2: in order to add *demonstrations* to a given input x (see Equation 1), we uniformly sampled x^- and x^+ from the top-50% most similar instances in τ_k.[5] To do so, we pre-computed the sentence embeddings of training instances using a pre-trained SBERT (Reimers and Gurevych, 2019) model, and cosine distance was used as a similarity metric.

Step 3: using *"causal"* and *"random"* as *word labels*,[6] the next step was to generate candidate

[4] These *demonstrations* (Gao et al., 2021) are used to demonstrate the LM, in-context, how it should provide the answer to the input prompt.

[5] We tested different percentages, however 50% was the best-performing one.

[6] We performed some simple preliminary tests using different words like "coincidence", "choice", "causal", "cause", with few trivial hand-crafted templates (e.g. "[x] *It was* [MASK]"), from which "random" and "casual" where selected.

	Precision		Recall		Accuracy		F1-Score	
Submission	*dev*	*test*	*dev*	*test*	*dev*	*test*	*dev*	*test*
Ensemble-10m	**88.46**	82.78	90.45	84.66	**88.26**	81.35	**89.44**	83.70
Prompt-256	85.49	**82.80**	**92.70**	87.50	87.30	**82.64**	88.95	**85.08**
Prompt-356e	82.72	80.41	88.76	**88.64**	83.60	81.35	85.63	84.32
Prompt-1000	84.56	81.08	91.57	85.22	86.07	80.39	87.87	83.10
Ensemble-8p	86.10	81.15	90.44	88.07	86.69	81.67	88.22	84.47

Table 1: Official performance metrics in percentages (%) from the selected methods in dev and test partitions of the Causal News Corpus.

templates automatically using T5. First, each training instance x of class y in τ_k was converted to "`[x]<P>`$word(y)$`<S>`" where `<P>` and `<S>` are T5 mask tokens, and used a 100 wide beam search to decode multiple template candidates by filling `<P>` and `<S>` tokens.

Step 4: next step was sorting all 100 final candidate templates by F1 score. However, since this is a time-consuming step, a subset of the evaluation set was used by sampling 256 unique positive and negative instances from δ_{T-k}. Note that no fine-tuning is used at this point, just the out-of-the-box pre-trained LM.

Step 5: we selected the top-10 best-performing templates as final candidates. For each candidate template we fine-tuned the LM as a MLM task (see Equation 1) on the training set, τ_k, evaluating it on the complete evaluation set, δ_{T-k}.

Step 6: finally, the model with the best F1 score on the official dev set was selected as a candidate for submission —we also checked that the F1 score on δ_{T-k} was among the first ones too (if not first). Note that in this step we're evaluating the model on unseen data since the official dev set is being used as an unofficial test set.

The above process was repeated varying the number k of training instances, with k = 256, 356, 512, and 1000;[7] the number d of input prompts to ensemble during classification stage, with d from 1 to 9; and using RoBERTa (large and base), and DeBERTa V3 (base) as pre-trained LMs. In step 5, models were fine-tuned for a maximum of 1000 steps using AdamW (Loshchilov and Hutter, 2019) optimizer (β_1=0.9, β_2=0.999, ϵ=1e$-$8) with a learning rate of γ=1e$-$5 with no weight decay (λ=0). Models were evaluated every 100 steps and check-pointed when new best F1 scores were obtained.

Label	Train	Dev	Test	Total
Causal	1603	178	176	1957
Non-causal	1322	145	135	1602
Total:	2925	323	311	**3559**

Table 2: Number of positive (causal) and negative (non-causal) instances in the *train, dev,* and *test* sets of the shared task. We refer the interested reader to (Tan et al., 2022b) to know more details about the data and the labeling process.

4 Results & Discussion

In this section we provide the details of the employed dataset, a set of additional experiments based on recent ensemble techniques, and the final configuration of our submitted runs to the subtask 1 of CASE 2022.

4.1 Dataset

As mentioned earlier, the main goal of subtask 1 of CASE-2022 is to classify whether or not a given sentence contains a *cause-effect* relation. Thus, systems have to be able to predict *Causal* or *Non-causal* labels per sentence. Table 2 contains a few statistics regarding the distribution of the classes in the *train, dev*, and *test* partitions.

4.2 Ensemble-based Approach

We also performed several ensembles of different fine-tuned LMs to increase the generalization and compensate for the overfitting of the models. We followed the approach described in Fajcik et al. (2019), called *TOP-N* fusion. In this formulation, we first define a *set* of M pre-trained LMs, varying the training seed. *TOP-N* fusion starts by choosing

[7] Inspired by evidence showing a performance saturation when k = 256 (Figure 3 in Gao et al. (2021)), compared to standard fine-tuning on the entire dataset, we decided to start from this value.

one uniformly random model from the *set*, which is added to the ensemble. Next, it randomly shuffles the rest of the models and tries adding them into the ensemble once, as long as the F1 score improves. Each time a model is added to the ensemble, its performance gets measured. The model would stay in the ensemble only and only if it improved the overall performance. This aims at an iterative optimization of the ensemble's F1 score by averaging the output probabilities. As the selection process is stochastic, we repeat the process N=10000 times. We construct a new ensemble for each iteration, independently of the previous ones. Finally, we select the best performing ensemble for submission. Further details are given in Appendix B (Figure 2).

4.3 Official Submissions

Next, we describe each one of our submissions:

Ensemble-10m: ensemble model described in subsection 4.2 with 10 final models obtained from a set of 150 initial ones (50 fine-tuned *bert-base-cased*, *roberta-base*, and *deberta-v3-base* models).

Prompt-256: prompt-based *roberta-large* model with k=256 training instances per class, d=3 input prompts to ensemble during classification stage; and template t = "`[x]` *This is not* `[MASK]`".

Prompt-1000: The same previous model but with t = "`[x]` *There were no* `[MASK]`*ities in this*", k=1000, and d=1.

Ensemble-8p: ensemble model described in subsection 4.2 with 8 final models obtained from the top-50 best performing prompt-based models as the initial set.

Prompt-356e: three prompt-base models trained with k=356 instances. The first two models have the same template as *Prompt-1000* but with d=2 and 3, respectively. The third one uses the template t = "`[x]` *The incident is not* `[MASK]`" with d=1.[8] Finally, a simple majority voting ensemble among these three models generates the output.

4.4 Results

Table 1 shows the official results, both in dev and test partitions, for our five submissions. As expected, the ensemble of several LMs (Ensemble-10m) was able to obtain outstanding performance across several metrics during the validation phase (i.e., dev partition[9]). However, the performance dropped significantly in the test partition (F1= 89.44 → F1= 83.70). On the contrary, our prompt-based approach trained on 256 instances per class (Prompt-256) could generalize better on the test partition. Such submission obtained 2nd place in terms of precision (82.80%), 3rd in accuracy (82.64%), and 5th in F1 (85.08%) —the best F1 was 86.19%. However, the main advantage of our approach is that it allows the LM to be trained in a few-shot setting, making it harder for the model to overfit the data. Moreover, most of the available data can be kept and used for measuring the generalization power of the model instead. For instance, our best-performing model (Prompt-256) was fine-tuned only on 15.7% of all available data,[10] allowing the remaining 84.3% to be used for evaluation and model selection (74.3% as evaluation set and 10% as our own test set). Therefore, model selection choice is more robust since the risk of performance drop on unseen data, such as the official test set, is expected to be lower.

5 Conclusions

This paper describes our participation in the CASE-2022 subtask 1. Our proposed approach uses a few-shot configuration in which a prompt-based model is fine-tuned using *only 256 instances* per class and yet was able to obtain remarkable results among all 16 participant teams. The comparison against traditional fine-tuning techniques, ensemble approaches, as well as the other participating models, show the potential of the proposed approach for better generalizing the posed task.

For future work, we plan to perform further ablation studies when we have access to test set ground truth labels. For instance, measuring the dev-to-test performance drop in relation to k or the robustness against different training and demonstration sampling given a fixed k.

Acknowledgements

This work was supported by CRiTERIA, an EU project, funded under the Horizon 2020 programme, grant agreement no. 101021866 and the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140). Esaú Villatoro-Tello, was supported partially by Idiap, SNI CONACyT, and UAM-Cuajimalpa Mexico.

[8] Note that these prompts, as well as previous ones, were automatically generated as described in section 3.

[9] We further performed a 5-cvf experiment on six different architectures, see the results on Table 3 in Appendix A.

[10] i.e. train + dev sets in Table 2

References

Raja Ayyanar, George Koomullil, and Hariharan Ramasangu. 2019a. Causal relation classification using convolutional neural networks and grammar tags. In *2019 IEEE 16th India Council International Conference (INDICON)*, pages 1–3. IEEE.

Raja Ayyanar, George Koomullil, and Hariharan Ramasangu. 2019b. Causal relation classification using convolutional neural networks and grammar tags. In *2019 IEEE 16th India Council International Conference (INDICON)*, pages 1–3.

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. Association for Computational Linguistics.

Martin Fajcik, Josef Jon, Martin Docekal, and Pavel Smrz. 2020. BUT-FIT at SemEval-2020 task 5: Automatic detection of counterfactual statements with deep pre-trained language representation models. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 437–444, Barcelona (online). International Committee for Computational Linguistics.

Martin Fajcik, Muskaan Singh, Juan Zuluaga-Gomez, Esaú Villatoro-Tello, Sergio Burdisso, Petr Motlicek, and Pavel Smrz. 2022. Idiapers @ causal news corpus 2022: Extracting cause-effect-signal triplets via pre-trained autoregressive language model. In *The 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE @ EMNLP 2022)*. Association for Computational Linguistics.

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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Daniela Garcia. 1997. Coatis, an nlp system to locate expressions of actions connected by causality links. In *Knowledge Acquisition, Modeling and Management*, pages 347–352, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *ArXiv preprint*, abs/2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *ArXiv preprint*, abs/1606.08415.

Christopher Hidey and Kathy McKeown. 2016. Identifying causal relations using parallel Wikipedia articles. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1424–1433, Berlin, Germany. Association for Computational Linguistics.

Vivek Khetan, Roshni Ramnani, Mayuresh Anand, Subhashis Sengupta, and Andrew E. Fano. 2022. Causal bert: Language models for causality detection between events expressed in text. In *Intelligent Computing*, pages 965–980, Cham. Springer International Publishing.

Christopher S. G. Khoo, Syin Chan, and Yun Niu. 2000. Extracting causal knowledge from a medical database using graphical patterns. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 336–343, Hong Kong. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaoning Li, Qi Li, Xiaotian Zou, and Jiangtao Ren. 2021. Causality extraction based on self-attentive bilstm-crf with transferred embeddings. *Neurocomputing*, 423:207–219.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ArXiv preprint*, abs/2107.13586.

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*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Hieu Man, Minh Nguyen, and Thien Nguyen. 2022. Event causality identification via generation of important context words. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, pages 323–330, Seattle, Washington. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Attapol Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 645–654, Gothenburg, Sweden. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv preprint*, abs/1910.01108.

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

Fiona Anting Tan, Ali Hürriyetoğlu, Tommaso Caselli, Nelleke Oostdijk, Hansi Hettiarachchi, Tadashi Nomoto, Onur Uca, and Farhana Ferdousi Liza. 2022b. 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, Xinyu Zuo, and See-Kiong Ng. 2022c. Unicausal: Unified benchmark and model for causal text mining. *ArXiv preprint*, abs/2208.09163.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Baseline results

We performed standard cross-entropy fine-tuning on six different pre-trained LMs (see first column in Table 3) to produce baselines. We perform 5-fold cross-validation for each architecture following the partitions proposed in Tan et al. (2022a). Each system is fine-tuned on the sequence classification task to discriminate between casual and non-causal text input sequences. We report the mean and standard deviation (mean ± std) on the official development set over several metrics, see Table 3.

During experimentation, we use the same learning rate of $\gamma = 5\text{e}{-5}$ with a linear learning rate scheduler. Dropout is set to $dp = 0.1$ for the attention and hidden layers, while Gaussian Error Linear Units (GELU) is used as activation function (Hendrycks and Gimpel, 2016). We fine-tune each model with an effective batch size of 32 for 50 epochs with AdamW (Loshchilov and Hutter, 2019) optimizer (β_1=0.9, β_2=0.999, ϵ=1e−8). We noted that *deberta-v3-base* performed systematically better in all metrics as shown in Table 3.

B Ensembling

We compose ensembles before submission to leaderboard in two manners. `Ensembling-type-1` and `Ensembling-type-2`:

- `Ensembling-type-1`: we define a *set* of models, which contains only baseline LMs fine-tuned on the sequence classification task (see Table 3). We fine-tune 50 LMs for each architecture from first column of Table 3. Next, we run our *TOP-N* fusion algorithm (see subsection 4.2) with the *set* of models previously defined. The model submitted with `Ensembling-type-1` is **Ensemble-10m**, reporting its performance in Table 1.

- `Ensembling-type-2`, we define a *set* of models containing prompt-based LMs. We select the top models for leaderboard submission. The overall process for ensembling is illustrated in Figure 2. Even though the figure only depicts our first approach (explained above), we perform exactly the same with the prompt-based models explained in section 3. The model submitted with `Ensembling-type-2` is **Ensemble-8p**, reporting its performance in Table 1.

Details about the ensemble: we select the best ensembles based on its F1-score performance on the dev set. For example, in Table 4 we list the performance of the `Ensembling-type-1` system (i.e., **Ensemble-10m**) we used for our submission in the leaderboard.

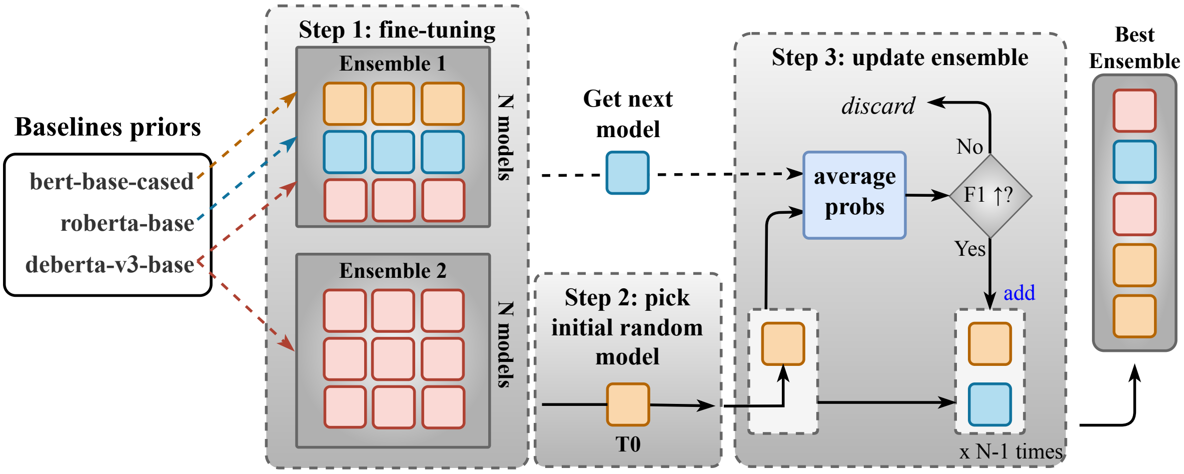

Figure 2: Our proposed method to ensemble N fine-tuned LMs, based on Fajcik et al. (2019) approach. We fine-tune several LMs by modifying only the training seed. Our implementation uses the sequence classification task from HuggingFace toolkit (Wolf et al., 2020; Lhoest et al., 2021).

Model	Precision	Recall	Accuracy	F1-score	Reference
`bert-base-cased`	83.52 ± 1.01	87.88 ± 3.08	79.68 ± 1.83	81.03 ± 1.20	(Devlin et al., 2019)
`bart-base`	84.21 ± 0.88	87.80 ± 2.26	80.99 ± 2.19	81.98 ± 0.95	(Lewis et al., 2020)
`roberta-base`	85.13 ± 1.11	87.86 ± 2.41	82.66 ± 2.35	83.21 ± 1.10	(Liu et al., 2019)
`distilroberta-base`	84.41 ± 1.20	88.05 ± 1.69	81.12 ± 2.09	82.22 ± 1.12	(Sanh et al., 2019)
`deberta-base`	82.67 ± 2.76	85.74 ± 2.72	80.32 ± 6.49	80.31 ± 3.44	(He et al., 2021b)
`deberta-v3-base`	**85.87** ± **1.18**	**88.88** ± **1.74**	**83.18** ± **3.16**	**84.00** ± **1.18**	(He et al., 2021a)

Table 3: Mean and standard deviation (mean ± std) of different metrics on the dev set using a 5-fold cross validation scheme on the CNC dataset. We report results for six different architectures of pre-trained LMs.

Model	F1-score (%)
`bert-base-cased`	85.15
`roberta-base`	86.76
`deberta-v3-base`	89.69
Ensemble-10m[†]	**89.7**

Table 4: Obtained F1-scores on the *dev* partition of subtask 1 of the Causal News Corpus. Results depict the top performance of three models that belong to the ***Ensemble-10m*** configuration. The last row corresponds to an ensemble model composed of ten independent LMs, namely, six *deberta-v3-base*, two *bert-base-cased*, and two *roberta-base*. More details about the ensemble construction are described in subsection 4.2 and Fajcik et al. (2019).