HEVS-TUW at SemEval-2023 Task 8: Ensemble of Language Models and Rule-based Classifiers for Claims Identification and PICO Extraction

Anjani Dhrangadhariya[1,2†] **Wojciech Kusa**[3†]
Henning Müller[1,2] **Allan Hanbury**[3]

[1]University of Geneva, Geneva, Switzerland
[2]HES-SO Valais-Wallis, Sierre, Switzerland
{anjani.dhrangadhariya,henning.mueller}@hevs.ch
[3]TU Wien, Vienna, Austria
{wojciech.kusa,allan.hanbury}@tuwien.ac.at

Abstract

This paper describes the HEVS-TUW team submission to the SemEval-2023 Task 8: Causal Claims. We participated in two subtasks: (1) causal claims detection and (2) PIO identification. For subtask 1, we experimented with an ensemble of weakly supervised question detection and fine-tuned Transformer-based models. For subtask 2 of PIO frame extraction, we used a combination of deep representation learning and a rule-based approach. Our best model for subtask 1 ranks fourth with an F1-score of 65.77%. It shows moderate benefit from ensembling models pre-trained on independent categories. The results for subtask 2 warrant further investigation for improvement.

1 Introduction

Identification and verification of causal claims from unstructured text data is essential for various decision-making processes, particularly in healthcare. The SemEval-2023 Task 8 (Khetan et al., 2023) aims to advance the state-of-the-art in this area by focusing on two subtasks: identification of causal claims and extraction of Population, Intervention, and Outcome (PIO) entities. The first subtask involves identifying the span of text that contains one of the four entities: a *causal claim*, a *personal experience*, a *personal experience based on a claim* or a *question*. This can be done at the sentence level, but only a part of a sentence may be annotated with one of these categories. The second subtask involves extracting the PIO frame related to the identified causal claim in a text snippet. The model utilizes both word-level, including contextual information and character-level features capturing different aspects of the data.

For subtask 1, our approach consisted of an ensemble of pre-trained language models with a rule-based question classifier. Apart from training multi-class classification models, with the assumption that *personal experience based on a claim* category can be treated both as an instance of a *personal experience* and a *causal claim*, language models were also fine-tuned as three independent binary classifiers. In subtask 2, we experimented with a system combining a deep learning entity extraction pipeline incorporating different textual features and followed by a rule-based approach to combine separate PIO tokens predictions into a consensus prediction sequence.

Our approach to subtask 1 fared well at rank 4, with F1-score 32.77% better than the last ranked approach and 12.7% behind the approach ranked first. For subtask 2, the approach ranks second last on the leaderboard, and the system mainly struggles with identifying the population frame. These subtasks have potential applications in content moderation, insurance claim identification, and hypothesis generation from clinical notes. The shared task will motivate further research in this direction and lead to the development of more effective and accurate methods for causal claim identification and PIO frame extraction.

2 Background

Causal claims identification in the open domain is widely researched, but the healthcare domain has only garnered attention recently (Mueller and Huettemann, 2018; Wang et al., 2019; Parveen et al., 2021; Islam et al., 2021). In the healthcare domain, large amounts of medical notes, social media posts, research articles, and patient forums are generated daily. Manually extracting causal claims and PIO

† Equal contribution.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1776–1782

frames from such data is time-consuming and error-prone.

For a decade, PIO extraction was limited to sentence-level information extraction due to the unavailability of frame-annotated datasets (Boudin et al., 2010; Jin and Szolovits, 2018). After the release of the EBM-PICO corpus, the extraction efforts moved to span and frame extraction (Nye et al., 2018). Nonetheless, previous studies on PIO frame extraction primarily concentrated on extracting them from well-written, peer-reviewed literature (Brockmeier et al., 2019; Zhang et al., 2020; Dhrangadhariya et al., 2021). The SemEval-2023 Task 8 overtakes the challenge of extracting these frames from noisy social media data. The task organizers provide 597 English-language PIO-labelled Reddit posts. We approach PIO frame extraction as binary sequence labelling and use a combination of deep learning and a rule-based approach that captures multiple feature representations from the data, as the dataset is relatively small and noisy.

The SemEval-2023 Task 8 provides an opportunity for researchers to develop novel methods for causal claim identification and PIO frame extraction from noisy social media data and to benchmark their performance against state-of-the-art methods. We hope that the shared task will lead to the development of more effective and accurate methods for identifying and extracting causal claims and PIO frames from unstructured text data.

3 System overview

We participated in both subtasks of SemEval-2023 Task 8. In this section we describe our approach.

3.1 Subtask 1

For subtask 1 we implement two components: weakly supervised question detection and a Transformers-based supervised classifier. Even though it would be possible to classify the sentences on a sentence-level, we decided to conduct more fine-grained token-level classification.

3.1.1 Question detector

We design a weakly supervised question detector approach (QD). We use a spaCy sentencizer to split the text into sentences. Next we search for the occurrences of the question mark token '?' and we assign it an end token of a question span. To find the beginning span, we either look for a punctuation token from '.', '!', '?' or a token in 'how', 'what', 'where', 'why'.

3.1.2 Supervised classification

We fine-tune five models using a training subset of the released dataset:

1. RoBERTa$_A$ – RoBERTa model (Liu et al., 2019a) fine-tuned on all entities.
2. distilBERT$_A$ – distilBERT model (Sanh et al., 2019) fine-tuned on all entities.
3. distilBERT$_q$ – distilBERT fine-tuned on extracting only questions.
4. distilBERT$_{pe}$ – distilBERT model fine-tuned on personal experience entities
5. distilBERT$_c$ – distilBERT model fine-tuned on claim entities

The three last models are trained as a binary classification. We assumed that *personal experience based on a claim* category could be treated both as an instance of a *personal experience* and a *causal claim* category. Therefore, all *personal experience based on a claim* tokens are treated as positive for both categories when fine-tuning distilBERT$_{pe}$ and distilBERT$_c$ models. This approach allows us to increase the training dataset size.

3.2 Subtask 2

PIO extraction was a three-part system: a text preprocessing module, a deep learning entity extraction pipeline and a rule-based approach to combine separate PIO predictions.

3.2.1 Preprocessing module

The PIO dataset was processed to parse annotation from the Reddit posts. The empty or partially deleted samples were removed, leaving 522 samples. The text tokens were enriched with the part-of-speech (POS) tags and lemmas using scispaCy[1].

3.2.2 Deep learning module

The deep learning system was built on combining a feature representation (word-level and character-level) component followed by a linear sequence labelling layer. We developed our feature representation approach based on the work of Aguilar et al. (2019), but with the difference that we did not train our system as a multi-task learning system. Social media text is highly noisy with writing variations, non-standard abbreviation, spelling errors

[1] https://spacy.io/universe/project/scispacy

and grammatically flawed. Character level representations are useful for capturing the finer details of language, such as spelling variations, unusual short forms, and other non-standard forms of language that are often used in social media. Word level representations are important for capturing the overall meaning and context of the language used in social media.

Word-level features: The word-level features included transformer and POS features. Transformer models, specifically **RoBERTa** and **BioMed-RoBERTa**, were used to extract $\mathbf{T}^{d \times l}$ dimensional contextual features from input samples, where d is the transformer model's hidden layer dimension (Liu et al., 2019b; Gururangan et al., 2020). POS information helps NLP models better understand the syntactic structure of a sentence. The **POS** embeddings were $\mathbf{P}^{d_p \times 512}$ dimensional one-hot sparse vectors corresponding to the 18-dimensional POS features and the maximum tokens ($l = 512$) allowed per transformer input. POS features were either **one-hot** encoded or transformed using a **BiLSTM** (bidirectional Long short-term memory) to encode long-term dependencies and learn a task-specific grammatical structure from the input samples (Hochreiter and Schmidhuber, 1997). Transformer and POS features were concatenated to obtain a word-level representation (see Figure 1).

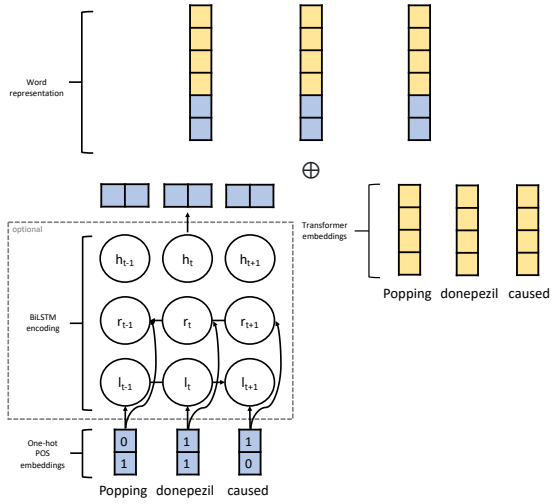

Figure 1: Word representation using concatenation of the POS embeddings (in blue, with or without BiLSTM transformation) and transformer embeddings (yellow).

Character-level features: To obtain the **character features**, input characters are embedded into a $\mathbf{C}^{d_c \times wl}$ dimensional one-hot encoded vector, where d_c is the dimension of the features per character, and wl is the maximum length of characters per word. **Orthographic features** are the character-based features $\mathbf{O}^{d_o \times wl}$ that encapsulate word shape, including letter capitalization, punctuation, and digits, e.g., "SemEval 2023!" encoded as "CccCccc nnnnp". A maximum of 20 characters per token (wl) were allowed applying post-padding on shorter tokens and truncating the longer ones. Character features are matrices encapsulating character-level information, including individual alphabet and punctuation and are hence sparser than the orthographic features. Our system either used the orthographic or the character features, which were transformed using a 1-dimensional convolutional neural network (**1D-CNN**) followed by either a max pooling (**MP**) or global average pooling (**GAP**) operation (Zhou et al., 2016). Next, the results were fed through a fully-connected (**fc**) layer via **ReLU** (Rectified Linear Unit) to obtain a character-level representation (see Figure 2). The final word- and character-level representations were concatenated and fed to a **linear layer** to predict the entity sequence.

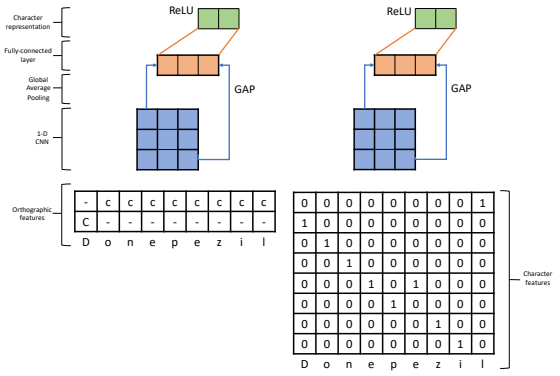

Figure 2: Character representation was was either obtained using the orthographic encoding (left) or character encoding (right) as explained in the system overview.

3.2.3 Rule-based module

The processed word and character representations were concatenated and the results were fed to a linear layer to individually predict the PIO entity. Finally, when multiple predictions were made for the PIO entities, the most common prediction was used, with a random choice being made for ties.

4 Experimental setup

This section describes our experimental setup for subtasks (1) and (2), both of which were evaluated

using a macro F1-score measure.

4.1 Subtask 1

We train all Transformer-based models for 30 epochs. Following task organizers, we perform splitting on whitespaces. We use 80% of the dataset for training models and 20% for validation. We use Flair 0.11.3 to implement our models (Akbik et al., 2019).

For subtask 1, we submit four runs, ensembles of models described in Section 3.1. The runs are described in Table 1.

Run name	Description
Run 1	QD + distilBERT$_q$ + distilBERT$_A$
Run 2	QD + distilBERT$_q$ + distilBERT$_{pe}$ + distilBERT$_A$
Run 3	QD + distilBERT$_q$ + distilBERT$_{pe}$ + RoBERTa$_A$
Run 4	QD + distilBERT$_q$ + distilBERT$_{pe}$ + distilBERT$_A$ + RoBERTa$_A$

Table 1: The table outlining the model architecture utilized in four different runs for the task of causal claims detection.

We do the majority voting (MV) for non-null predictions, i.e. whenever at least one model predicts an entity, we give that prediction higher precedence compared to a zero-class prediction from other models. Moreover, when at least one model predicts the 'claim' category and at least one predicts 'personal experience' we assign to that token the 'claim based on personal experience' entity. We did not submit a run with distilBERT$_c$ model, as the model was not able to predict any entity correctly on a validation set.

4.2 Subtask 2

The experiments were conducted using PyTorch 1.10, scispaCy 0.4.0, and transformers 4.8.2. We submitted five runs with three training experiments each for PIO classes using the system components described in section 3.2. For each PIO run, the experiments were conducted and averaged over the three most common Python random seeds: 0, 1 and 42. The dataset was divided into training (80%) and validation sets (20%). The sequence input length was 512 tokens across the experiments corresponding to the transformer input restriction. The dimensions for the contextual word embeddings were $\mathbf{T}^{768\times512}$ and for the one-hot POS embeddings were $\mathbf{P}^{18\times512}$. The dimensions for the orthographic character embeddings were $\mathbf{O}^{6\times20\times512}$ and for the character embeddings were $\mathbf{C}^{28\times20\times512}$. The Table 2 describes the architecture for each run.

5 Results and Discussion

5.1 Subtask 1

Results for our submissions in subtask 1 are presented in Table 3. Adding distilBERT$_{pe}$ to the ensemble positively impacts Recall and this is the best of our Runs in terms of the F1-score. Replacing distilBERT$_A$ with RoBERTa$_A$ model (Run 3) improves the Precision by almost 4% points, yet it decreases the overall Recall. When using both of these models in the ensemble, the scores on all evaluation measures decreases. We believe that this was due to the usage of a naive MV approach.

5.2 Subtask 2

Subtask 2 F1-scores for the official SemEval-2023 test set are presented in Table 4. Run 3 had the best scoring architecture for the mean macro F1 for PIO classes and fared best for intervention and outcome classes. Run 2 had the best score only for the population class. Our best F1-score for the task was placed fifth on the leaderboard, 17.91% points lower than the approach ranked one and 2.52% better than the last ranked approach.

For PIO extraction, we conducted preliminary experiments using a combination of BiLSTM CRF (Conditional Random Fields), but its linear layer counterpart consistently outperformed the CRF layer. We suspect the reason could be using the IO tagging scheme. In the case of social media entities, the boundaries of these entities may be fuzzy and not well-defined. In this situation, using the IO tagging scheme can be more appropriate than the BIO tagging scheme. The IO tagging scheme only requires a single tag to mark the beginning of an entity, and all subsequent words in the entity are labelled with the same tag. This makes it easier to encode fuzzy boundaries and reduces the number of tags required to label the sequence. We used a linear rather than a CRF layer to model the dependencies between adjacent labels. This decision was influenced by the fact that the IO tagging scheme is better suited for a binary sequence labelling task than a sequence tagging task, better performed by CRF. Using a linear layer, we can model the depen-

Run name	Model architecture description
Run 1	RoBERTa embeddings extracted from the input tokens were concatenated to BiLSTM-transformed POS embeddings to obtain a word-level representation. Character-level orthographic embeddings were CNN transformed, followed by a max pooling operation to obtain the character representation post an fc layer via ReLU.
Run 2	RoBERTa embeddings extracted from the input tokens were concatenated to one-hot encoded POS embeddings. Character vectors were CNN transformed, followed by a max pooling operation to obtain the character representation post an fc layer via ReLU.
Run 3	RoBERTa embeddings were concatenated to one-hot encoded POS embeddings. Character-level orthographic embeddings were CNN transformed, followed by a GAP operation to obtain the character representation post an fc layer via ReLU.
Run 4	RoBERTa embeddings were concatenated to BiLSTM transformed POS embeddings. Character-level orthographic embeddings were CNN transformed, followed by a GAP operation to obtain the character representation post an fc layer via ReLU.
Run 5	Three different architectures, each corresponding to PIO classes, were used for the fifth run. For the population class prediction, the architecture was the same as in run 4, except RoBERTa embeddings were replaced by BioMed-RoBERTa representation. For the intervention prediction, the architecture was the same as in run 3, except RoBERTa embeddings were replaced by BioMed-RoBERTa representation. For the prediction of the outcome, only RoBERTa embeddings were extracted from the input tokens, followed by a linear layer for class prediction.

Table 2: The table outlining the model architecture utilized in five different runs for the task of PIO extraction.

Run	Precision	Recall	F1-score
1	68.13	61.21	64.48
2	66.81	**66.12**	**66.46**
3	**70.32**	63.38	64.52
4	68.73	62.90	65.70

Table 3: Subtask 1 results on the official SemEval-2023 test set. **Bold** values indicate highest score overall.

	Macro F1-acore			
Run	Pop	Int	Out	Overall
1	12.39	19.43	22.10	17.97
2	**21.30**	20.05	20.33	20.56
3	17.44	**26.39**	**22.78**	**22.20**
4	11.54	25.63	22.74	19.97
5	08.18	16.47	12.28	12.31

Table 4: Subtask 2 F1-score on the official SemEval-2023 test set for the Population, Intervention and Outcome classes over five runs. Note: Pop = population, Int = intervention, Out = outcome. **Bold** values indicate highest score overall.

dencies between adjacent labels and still capture the fuzzy boundaries of social media entities.

6 Conclusion

We participated in both subtasks of SemEval-2023 Task 8. Our submissions are mainly based on fine-tuning Transformers-based models and creating an ensemble of these models. Results show a positive impact of using independent binary classification models for each entity type in subtask 1. Source code is available under the following URL: https://github.com/WojciechKusa/pico-semeval2023

Limitations

For subtask 2, the rule-based module uses MV to choose the final prediction. MV selects the final token label supported by most of the model runs by equally weighting each run and discounting the accuracy of each model. The system could benefit from considering model accuracies and weighting predictions from each model to support the final prediction. Additionally, the final prediction aggregation scheme is stringent and selects only the

label predicted by multiple voters. It could increase the impact of out-of-the-span labels constituting the majority class leading to a lower recall and F1 score. Consider, for example, only one voter labelling one of the PIO entities and the rest labelling out-of-the-span entity. For such cases, the rule could be adapted for leniency and selecting the entity in the case at least one voters predict it.

Acknowledgements

This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721) and HES-SO Valais-Wallis, Sierre, Switzerland.

References

Gustavo Aguilar, Suraj Maharjan, Adrian Pastor López-Monroy, and Thamar Solorio. 2019. A multi-task approach for named entity recognition in social media data. *arXiv preprint arXiv:1906.04135*.

A. Akbik, Tanja Bergmann, Duncan A. J. Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. In *North American Chapter of the Association for Computational Linguistics*.

Florian Boudin, Jian-Yun Nie, Joan C Bartlett, Roland Grad, Pierre Pluye, and Martin Dawes. 2010. Combining classifiers for robust PICO element detection. *BMC medical informatics and decision making*, 10(1):1–6.

Austin J Brockmeier, Meizhi Ju, Piotr Przybyła, and Sophia Ananiadou. 2019. Improving reference prioritisation with pico recognition. *BMC Medical Inform Decis Mak*, 19(1):1–14.

Anjani Dhrangadhariya, Gustavo Aguilar, Thamar Solorio, Roger Hilfiker, and Henning Müller. 2021. End-to-end fine-grained neural entity recognition of patients, interventions, outcomes. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 65–77. Springer.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Md Rakibul Islam, Jiaxian Yin, Yanshan Wang, and William Yang Wang. 2021. Identifying causal relations in clinical texts using convolutional neural networks with dependency parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3543–3553.

Di Jin and Peter Szolovits. 2018. PICO element detection in medical text via long short-term memory neural networks. In *Proceedings of the BioNLP 2018 workshop*, pages 67–75. Association for Computational Linguistics.

Vivek Khetan, Somin Wadhwa, Byron Wallace, and Silvio Amir. 2023. Semeval-2023 task 8: Causal medical claim identification and related pio frame extraction from social media posts. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Roland M Mueller and Sebastian Huettemann. 2018. Extracting causal claims from information systems papers with natural language processing for theory ontology learning.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain J Marshall, Ani Nenkova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2018, page 197. NIH Public Access.

Sabiha Parveen, Saad Hasan, Diego Molla, Machteld van der Meijden, and Pinar Ozturk. 2021. Automatic extraction of causal relations from health forums: A comparative study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5803–5813.

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.

Yanan Wang, Chuanxi Li, Yansong Feng, and Dongyan Zhang. 2019. Identifying causal relationships in scientific texts using a graph-based approach. 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 1606–1615.

Tengteng Zhang, Yiqin Yu, Jing Mei, Zefang Tang, Xiang Zhang, and Shaochun Li. 2020. Unlocking the power of deep PICO extraction: Step-wise medical ner identification. *arXiv preprint arXiv:2005.06601*.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.