NEXT: An Event Schema Extension Approach for Closed-Domain Event Extraction Models

Elena Tuparova[1,2], **Petar Ivanov**[1], **Andrey Tagarev**[1], **Svetla Boytcheva**[1,2], **Ivan Koychev**[2]
[1]*Ontotext AD, Bulgaria*
[2]*Faculty of Mathematics and Informatics, Sofia University, Bulgaria*
elena.tuparova@ontotext.com, petar.ivanov@ontotext.com,
andrey.tagarev@ontotext.com, svetla.boytcheva@ontotext.com,
koychev@fmi.uni-sofia.bg

Abstract

Event extraction from textual data is an NLP research task relevant to a plethora of domains. Most approaches aim to recognize events from a predefined event schema, consisting of event types and their corresponding arguments. For domains such as disinformation, where new topics frequently emerge, there is a need to adapt such a fixed schema of events to accommodate new types of events. We present NEXT (New Event eXTraction) - a resource-sparse approach to extend a close-domain model to novel event types, that requires a very small number of annotated samples for fine-tuning performed on a single GPU. Furthermore, our results suggest that this approach is suitable not only for the extraction of new event types but also for the recognition of existing event types, as the use of this approach on a new dataset leads to improved recall for all existing events while retaining precision.

1 Introduction

Event extraction from text is a research task with applications in a wide range of domains (Liu et al., 2021), including finance (Sheng et al., 2021a), social (Ritter et al., 2012; Kunneman and Van Den Bosch, 2016), biomedical (Wei et al., 2020) and security (Tanev et al., 2008).

The goal of the event extraction task is to determine the event type, extract the trigger - the most relevant word to the event, as well as any event arguments - other words and phrases relevant to the event (Liu et al., 2021). This is often approached as a closed-domain problem where the model aims to detect events from a predefined event schema consisting of a fixed set of event types and their corresponding argument types (Sheng et al., 2021b). In contrast, when the set of event types is not fixed or is not completely known at the onset, an open-domain approach is more suitable (Liu et al., 2019).

We explore the task of event extraction within the field of fake news and disinformation as a closed-domain problem. Nonetheless, the highly dynamic nature of the field implies that a methodology for easy extension of an existing closed domain event extraction approach to new event types is necessary. Ideally, such methodology would perform well with a small number of annotated samples, as producing a large annotated dataset for each newly emerging event type would be a very long and expensive process.

In this paper, we present a work-in-progress methodology which satisfies the requirements mentioned above. We select an existing model and extend it for a novel event type identified in fake news debunks with minimal resources when it comes to annotated data. We present how we define and annotate a new event, followed by how we fine-tune an existing model. Next, we provide a detailed analysis of how well the model learns the new event type, as well as how well it retains the ability to predict the event types for which it was previously trained.

2 Related Work

Event extraction is a widely studied topic and many different approaches towards it exist. Li et al. (2022) identify two main paradigms to solving the event extraction task - the pipeline paradigm, where event type, trigger and argument classification are done in sequence (Zhao et al., 2018; Chen et al., 2015; Li et al., 2020), and the joint paradigm, where event and arguments are classified simultaneously (Sheng et al., 2021b; Wadden et al., 2019; Yang et al., 2019). The latter paradigm prevents error propagation from one classification sub-task to the next. Other notable approaches to event extraction are as a classification task (Zhao et al., 2018; Chen et al., 2015), question answering task (or machine reading comprehension task) (Li et al., 2020; Zhou et al., 2021; Lu et al., 2023), sequence labelling task (Sheng et al., 2021b; Wadden et al., 2019) or sequence-to-structure generation task (Lu

Proceedings of the 6th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text, pages 113–123, Varna, Bulgaria, Sep 7, 2023.
https://doi.org/10.26615/978-954-452-089-2_016

et al., 2021). Another interesting approach to the event detection task is the presented in Peng et al. (2023) reinforcement learning one.

Lu et al. (2021) point out that most event extraction methods, among them pipeline and joint paradigm approaches, apply a decomposition strategy where event extraction sub-tasks are solved independently and their results are then combined to predict the whole event entity. This strategy has some drawbacks, such as the need for annotations for different sub-tasks and the difficulty of composing an optimal architecture for different sub-tasks. Lu et al. (2021) addresses both of these by modelling all sub-tasks in a uniform sequence-to-structure generative model called Text2Event, which extracts events from a text in an end-to-end manner. Another advantage of the model is being able to easily transfer to new event types.

In our study we aim to find a transferable low-resource solution to event extraction, such that it adapts well to new corpora and new event types with small amounts of annotated data and can be run on a single GPU. While there are other approaches to event extraction with little annotated data such as semi-supervised (Zhou et al., 2021; Huang and Ji, 2020), few-shot (Lai et al., 2020; Deng et al., 2020) and zero-shot (Huang et al., 2018; Lyu et al., 2021; Yue et al., 2023) learning, we chose the Text2Event model for its reported high performance in both supervised and transfer learning settings. For these purposes we extend the Text2Event model (Lu et al., 2021) with a novel event type by fine-tuning it on a small annotated sample set and then evaluate how well the model retains its performance on its original event types on a novel dataset of fake news debunks.

3 Data

3.1 Exploratory data analysis

For the purposes of the present research we work with a database of fake news debunks. We have extracted a total of 78,246 short documents in different languages, where each document is a fact-checked claim. Most claims are one to two sentences in length but can go up to a few paragraphs. We used SpaCy[1] to filter claims in other languages, resulting in 42,555 claims in English. Additionally, we split these claims into 54,280 individual sentences, using a sentence tokenizer from NLTK[2].

Table 1: Results from running the Text2Event dyiepp_ace2005_en_t_large pre-trained model on our datasets of whole claims and individual sentences

	Whole claims	**Sentences**
No event	32,967	43,259
At least one event	9,588	11,021
Single event	6,509	8,602
Multiple events	3,079	2,419
All documents	**42,555**	**54,280**

As a first step, we want to know what event types from widely used taxonomies can be recognized in this data, as it has no labels regarding events. To achieve this we ran the dyiepp_ace2005_en_t_large version of Text2Event[3] (which comes pre-trained on the ACE 2005 dataset[4]) on our dataset of claims and also on the dataset of individual sentences from claims. We aim to find out what events from the ACE 2005 taxonomy are present and in how many documents[5]. The number of documents with recognized events is presented in Table 1. In both settings in only around one-fifth of the documents, there is at least one recognized event.

Figure 1 shows the numbers of documents containing predictions for the top 10 most recognized event types, using whole claims and sentences as input respectively.

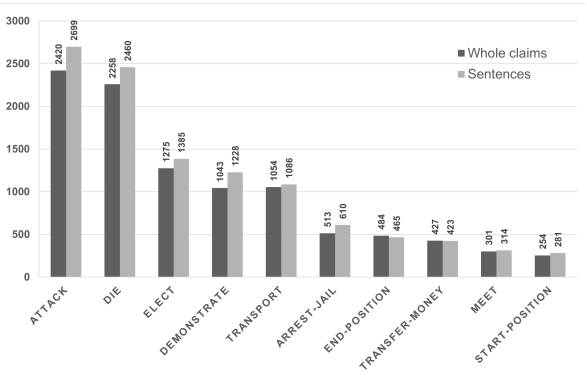

Figure 1: Number of documents containing predictions for the top 10 most recognized event types

[1] https://spacy.io/

[2] https://www.nltk.org/api/nltk.tokenize.html

[3] https://github.com/luyaojie/Text2Event

[4] https://catalog.ldc.upenn.edu/LDC2006T06

[5] When we write a document, we refer to an entity in the dataset, i.e. either a whole claim or a sentence, depending on the dataset.

3.2 New event type definition and annotated dataset creation

As the domain of our dataset is fake news and disinformation we define a new "Cure-Claim" event type which is of relevance to this particular area. A Cure-Claim event can be described as the act of stating whether something is a cure for a given medical condition or disease. We extracted 637 Cure-Claim candidate samples by selecting claims containing likely triggers (e.g. "cure", "treat", "heal"). Next, we defined the following seven arguments of the Cure-Claim event (step "Define new event annotation rules" on Figure 2):

- `Source` makes the claim;
- `Cure` is the remedy;
- `Condition` is what is treated by the cure;
- `Patient` is helped by the cure;
- `CureCreator` created the cure;
- `CureAdministrator` applies the cure.

An example of a document, mentioning the Cure-Claim event, is the following:

> "*Multiple posts shared repeatedly on Facebook* claim that *drinking tea made with pepper stems* is effective in preventing or ***curing*** *Covid-19*. The claim is false; the Association of Korean Medicine said there is no scientific evidence to support the claim."

Here, the first sentence is the event extent. ***Curing*** is the event trigger, and *"Multiple posts shared repeatedly on Facebook"*, *"drinking tea made with pepper stems"* and *"Covid-19"* are event arguments, respectively `Source`, `Cure` and `Condition`.

Due to the limited resources for annotation that we had, we selected 65 of these claims (around 10%) for manual annotation. Based on the official ACE event guidelines[6], we developed extensive annotator guidelines specifically for annotating Cure-Claim events. Following this, each document was annotated by three independent annotators, where the agreement between the majority was taken as final annotations. The resulting dataset contains 65 documents, of which 54 (83%) contain a Cure-Claim event. After performing sentence segmentation we obtain 74 sentences, of which again 54 (73%) contain a Cure-Claim event.

[6] https://shorturl.at/DEFV4

	ACE 2005	CCD
Documents	599	65
Sentences	16,372	74
Triggers	5,272	54
Arguments	9,612	147
Avg. no. triggers per event type	159.75	54
Med. no. triggers per event type	<100	54
Avg. no. sentences per document	27.33	1.14

Table 2: Comparison between ACE 2005 and novel Cure-Claim event dataset (CCD)

Comparison of statistics for the ACE 2005 dataset (as reported in Yang and Mitchell (2016)) and our annotated dataset for the Cure-Claim event are presented in Table 2.

3.3 Finding *k* for *k*-shot learning

We are interested in whether fewer than 74 annotated sentences would be sufficient to fine-tune the model for a new event type. To explore this, we use four different data splits of the type X/Y, where X is the percentage of training documents and Y is the percentage of test documents out of our annotated dataset. The data splits in question are 20/80, 40/60, 60/40 and 80/20.

4 Model

4.1 Text2Event overview

Text2Event (Lu et al., 2021) is a sequence-to-structure generative model that uses a transformer-based encoder-decoder architecture (Vaswani et al., 2017) to generate whole event structures from text in an end-to-end manner. The model is trained on the ACE 2005 and ERE datasets for English documents.

Text2Event is shown to perform well in transfer learning. The authors demonstrate fine-tuning on new event types on a separate subset of the same corpus. In contrast, we take the model pre-trained for the existing 33 event types on the whole ACE 2005 English dataset and fine-tune it for a new event type on a new corpus with different statistics from ACE 2005 (such as document length).

Text2Event can be trained or fine-tuned using substructure learning - the model learns separate substructures such as "(type, trigger words)" and "(role, argument words)", full structure learning - the whole event structure is learned at once, or curriculum learning, which combines the two.

4.2 Fine-tuning approach

We fine-tune the dyiepp_ace2005_en_t_large model which is pre-trained on the whole ACE 2005 English dataset on one NVIDIA RTX A5000 GPU.

We forgo substructure learning and use full structure learning only to fine-tune the model on the new event type. We use a learning rate of 1e-4 and a batch size of 16.

Given the small number of annotated training samples we use 5-fold cross-validation and compare the mean results of the models fine-tuned for different numbers of epochs (30, 100, 300 and 500) and on different train/test dataset splits (20/80, 40/60, 60/40 and 80/20).

Figure 2 illustrates our approach to annotated dataset creation and to using this dataset for fine-tuning the model.

5 Evaluation

5.1 Cross-validation experiments

We compare the performance of the fine-tuned models using precision, recall and F1-score on three subtasks: event type classification, trigger classification and argument classification. Definitions of true/false positives/negatives for trigger and argument classification are provided in the Appendix.

Table 3 contains the mean results from our cross-validation experiments. We first fine-tuned a model for 30 epochs which scored 0 on all metrics across all data splits. We then increased the number of fine-tuning epochs to 100 and more.

We first observe that when fine-tuned for larger number of epochs both event and trigger classification require as little as 12 samples to achieve the same level of precision as with four times as many samples. Recall, however, is poorer with fewer samples and improves significantly as the train set size increases. With 60 annotated samples the model learns to retrieve over 90% of the annotated Cure-Claim events.

Next, we examine the results for argument classification. We report separately scores for the cases when Cure-Claim events are predicted with the correct trigger (Correct-Trig-Arg-C columns) and when Cure-Claim events are predicted but with a wrong trigger (Wrong-Trig-Arg-C columns). Overall, both precision and recall tend to improve as the train set size increases, although drops in performance for the larger train set sizes are observed.

Compared to event type and trigger classification argument classification requires larger number of annotated training to achieve high precision, recall and F1 scores.

Standard deviations of the reported scores for Event-C, Trig-C and Correct-Trig-Arg-C range from 0.006 to 0.15 with only one outlier of 0.48. For Wrong-Trig-Arg-C the standard deviations range from 0.06 to 0.48, which could be due to this group being fairly smaller than the rest.

In addition to these results, in Figures 3 and 4 we also compare the number of additional argument classification mistakes from either false negative or false positive trigger classification cases. In the former case an event is annotated but not predicted, so all annotated arguments are counted as false negatives (Figure 3). In the latter case no event is annotated but one is predicted, so all predicted arguments are counted as false positives (Figure 4). We observe that as the train set size increases and the event classification precision and recall improve, the number of false positive or negative event predictions drops and so do consequently the corresponding false positive or negative argument predictions.

For all event classification subtasks the performance of the fine-tuned models increases with increase of the epoch count - the best results are generally reported for models fine-tuned for 500 epochs. Also, in most cases a bigger train set leads to better results. The biggest improvement in performance with increasing the training set size is observed for the models fine-tuned for 100 epochs. The models fine-tuned for 30 epochs output no significant results. All other models perform similarly when fine-tuned on the largest training set.

5.2 Cure-Claim prediction precision on broader dataset

We next fine-tuned the baseline model on the whole annotated dataset for the Cure-Claim event for 100 and 500 epochs. We evaluate the performance of the models on 2 broader datasets - the full dataset and a filtered subset with Cure-Claim candidate documents (10 times larger than our annotated dataset). For each dataset we manually evaluate 60 samples per model - half predicted only by that model and half predicted by both models. Comparing the two models by precision on those samples and by number of predictions made allows us to estimate whether performance worsens with more epochs (step "Estimate overfitting on new event type" in Figure 2).

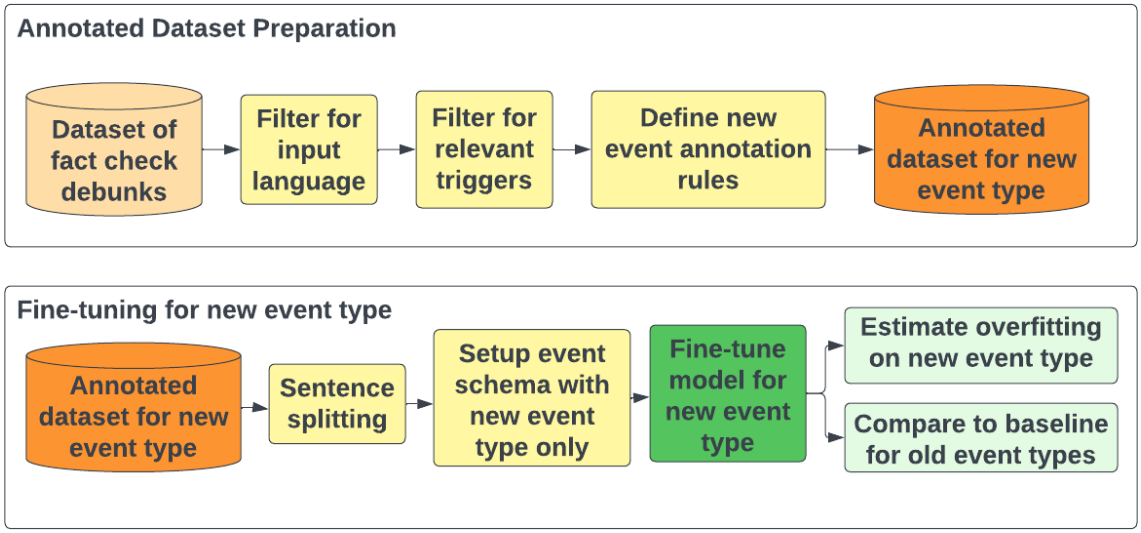

Figure 2: Steps towards building an annotated dataset and using it to fine-tune model on new event type

Model		Event-C			Trig-C			Correct-Trig-Arg-C			Wrong-Trig-Arg-C		
	split	P	R	F1	P	R	F1	P	R	F1	P	R	F1
100 epochs	20/80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	40/60	0.60	0.02	0.04	0.60	0.02	0.04	0.30	0.60	0.40	0.00	0.00	0.00
	60/40	0.83	0.84	0.82	0.76	0.83	0.78	0.54	0.75	0.63	0.75	0.83	0.79
	80/20	0.84	0.83	0.83	0.78	0.82	0.80	0.52	0.72	0.59	0.67	0.70	0.68
300 epochs	20/80	0.87	0.65	0.73	0.75	0.62	0.66	0.38	0.50	0.43	0.24	0.44	0.31
	40/60	0.80	0.78	0.79	0.71	0.76	0.73	**0.60**	0.74	**0.64**	0.43	0.76	0.55
	60/40	0.86	0.75	0.80	0.80	0.74	0.76	0.53	0.70	0.60	0.83	0.88	0.85
	80/20	0.84	0.83	0.84	0.80	0.82	0.81	0.53	0.73	0.61	0.60	0.60	0.60
500 epochs	20/80	0.85	0.69	0.75	0.72	0.66	0.68	0.48	0.60	0.53	0.32	0.54	0.40
	40/60	0.82	0.81	0.81	0.71	0.78	0.74	0.56	0.71	0.62	0.48	0.79	0.59
	60/40	**0.88**	0.83	0.85	**0.83**	0.82	0.82	0.55	0.74	0.63	**1.00**	**1.00**	**1.00**
	80/20	0.86	**0.92**	**0.89**	0.77	**0.92**	**0.84**	0.57	**0.76**	**0.64**	0.93	**1.00**	0.96

Table 3: Mean Precision (P), Recall (R) and F1-score for Cure-Claim event, trigger and argument classification (Event-C, Trig-C, Arg-C) for various train/test splits and number of fine-tuning epochs. Results for model fine-tuned for 30 epochs not shown as it scored 0 on all metrics across all train/test splits.

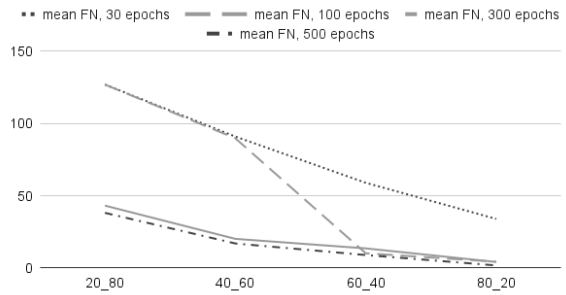

Figure 3: Number of false negative arguments for Cure-Claim event type across dataset splits and fine-tuning epochs

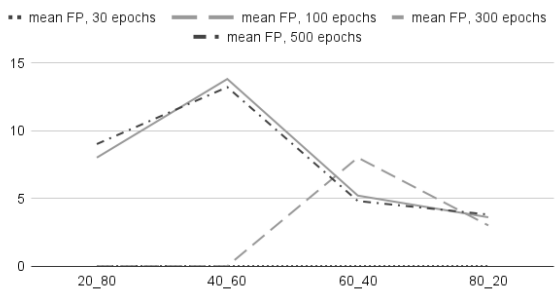

Figure 4: Number of false positive arguments for Cure-Claim event type across dataset splits and fine-tuning epochs

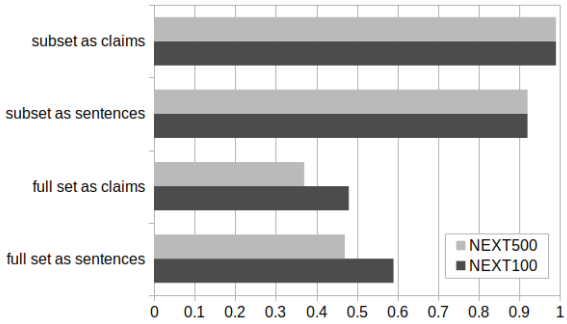

Figure 5: Estimated precision of event type classification on the filtered subset and the full dataset of fake news debunks for input provided as sentences and as claims.

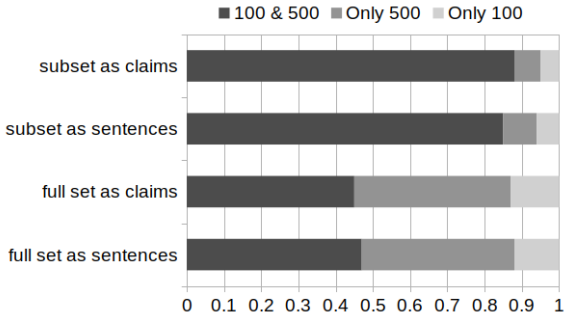

Figure 6: Fraction of predicted Cure-Claim events by each model on the filtered subset and the full dataset of fake news debunks for input provided as sentences and as claims.

Figure 5 shows the resulting estimates for precision of Cure-Claim event-type classification on the filtered subset and the full dataset. Both NEXT100 and NEXT500 achieve over 0.9 precision for the Cure-Claim event on the filtered subset samples, both when the sample is a whole claim and an individual sentence. These results surpass the models' precision in the earlier cross-validation experiments (Table 3) for all train/test splits on a larger evaluation set (60 samples per model).

On the other hand, precision drops significantly for both models on the samples from the whole dataset. For these more diverse samples we see that both fine-tuned models perform better when the input is provided as individual sentences. However, we also note that NEXT100 is more precise.

Figure 6 shows the fraction of predicted Cure-Claim events made by each fine-tuned model. We see that almost all predictions made on the filtered subset are made by both fine-tuned models. On the broader dataset, however, NEXT500 makes about 50% more predictions for Cure-Claim events than NEXT100. This, combined with the above-mentioned drop in precision of NEXT500 shown in Figure 5, suggests overfitting for NEXT500.

5.3 Overlap in original event types predictions between baseline and fine-tuned models

An essential part of the model fine-tuning is to assure that the model has not worsened its performance on the event types it was previously trained on. We don't have access to the annotated dataset with all event types that Text2Event was trained on, so to examine whether the fine-tuned model has retained the abilities of the original one, we compared their performance on the whole dataset of fake news debunks consisting of 42,555 claims and 54,280 sentences respectively. We compare the number of predictions per event type from the baseline model and the fine-tuned models, as well as the overlap of predicted events between any two or all three models.

Figure 7 shows that for the top 10 most common events the fine-tuned models predict many more event occurrences compared to the baseline model.

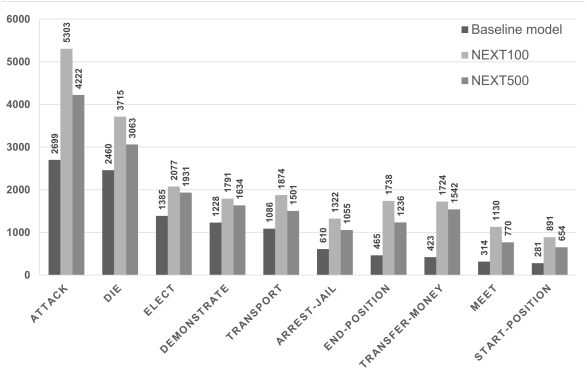

Figure 7: Number of documents containing predictions by baseline model, NEXT100 and NEXT500 (using sentence as input)

Figure 8 shows that for all event types almost all predictions by the baseline model are also predicted by the fine-tuned models, with NEXT100 having a higher overlap compared to NEXT500.

Another way to explore these overlaps is shown in Figure 9 where for each event type we can see what fraction of all predictions were made by all three models, by a particular pair of models, or by an individual model. We can observe that over half of all predictions either overlap between all three models or between the two fine-tuned models. Unlike the other two models, NEXT100 produces a significant number of predictions not matched by

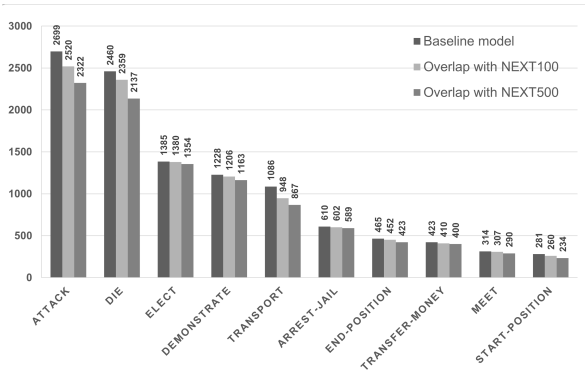

Figure 8: Number of documents containing predictions by baseline model and overlapped with NEXT100 and NEXT500 (using sentences as input)

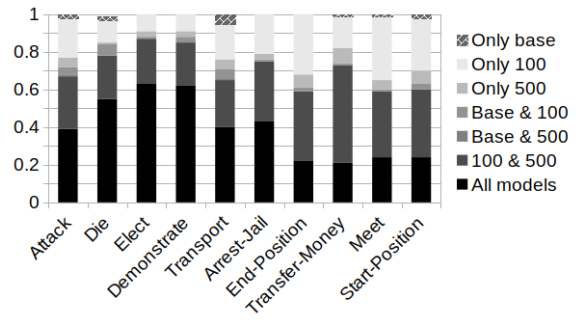

Figure 9: Overlap in event predictions between models (using sentences as input). For each event type the bar shows what fraction of predictions are made by all three models, by two of the models, or by a single one.

any of the other models. It is also worth noting that all predictions made by NEXT100, whether in agreement with other models or not, account for over 90% of all predictions.

Results of model evaluatuation on whole claims, rather than individual sentences, are very similar and figures and tables are included in the appendix.

5.4 Comparing precision of original event types predictions between baseline and fine-tuned models

For each event type we sample up to 20 predictions for each overlap subset (individual models, pairs of models, all models). For each prediction, we manually evaluated whether the corresponding document contains an event of such type, regardless of whether the trigger prediction is also correct. The resulting estimates for event type classification precision are given in Table 4. The estimated precision scores for individual models are obtained by combining the estimates over all relevant subsets (e.g. for the baseline model we add the number of correct predictions from only baseline, baseline & NEXT100, baseline & NEXT500 and all models) and are shown in Table 5.

5.5 Estimating recall of original event types predictions for baseline and fine-tuned models

We are unable to calculate recall and F1-score as those would require knowing the total number of positive samples for each event type for our fake news debunk dataset.

However, the precision of sampled predictions not made by the baseline model (i.e. made either by a single or by both fine-tuned models only) is on-par with the precision of sampled predictions made by the baseline model (usually also predicted by one or both of the fine-tuned models).

We can thus reason that the fine-tuned models not only retain the baseline model's recall but improve on it 2- to 4-fold, since for all event types the fine-tuned models generate two to four times as many predictions, as already shown in Figure 8.

6 Discussion

Our proposed approach NEXT to extend an existing event schema with new event types has a few advantages, but also limitations.

A notable advantage of this approach is that a dozen annotated samples are sufficient for achieving high precision given a sufficient number of fine-tuning epochs. Learning good recall, however, is a more challenging task and requires a larger number of samples - about 50.

Fine-tuning the model also does not require significant computational resources. All reported experiments were performed on a single CUDA-enabled GPU. Each fine-tuning of an individual model took a few minutes.

As expected, we observe that fine-tuning for many epochs leads to overfitting on the new event type. Namely, the precision of predictions for the new event type decreases with a larger number of epochs, while the number of predictions grows simultaneously. This problem can be mitigated by pre-filtering the sentences or claims on which the model is used, with a rule as simple as checking whether they contain likely triggers for the event type (e.g. "cure", "treat", "heal" for Cure-Claim events), as seen in Figure 5. Another solution would be to adopt a voting approach by considering only predictions made by both NEXT100 and

Event type	only base	only 100	only 500	base & 100	base & 500	100 & 500	all models
Attack	0.65	0.70	0.60	0.70	0.90	0.80	0.95
Die	0.95	1.00	1.00	1.00	1.00	1.00	1.00
Elect	1.00	0.85	0.83	1.00	1.00	1.00	1.00
Demonstrate	1.00	0.90	0.87	0.95	1.00	0.95	1.00
Transport	0.95	1.00	0.95	0.90	1.00	1.00	1.00
Arrest-Jail	0.83	0.85	0.80	1.00	0.50	0.90	1.00
End-Position	1.00	0.90	0.85	1.00	1.00	1.00	0.95
Transfer-Money	0.80	0.85	0.80	0.77	1.00	1.00	1.00
Meet	1.00	1.00	1.00	1.00	0.00	1.00	1.00
Start-Position	1.00	0.90	1.00	1.00	1.00	0.95	1.00

Table 4: Event type classification precision on subsets of sampled predictions made by individual models (baseline, NEXT100 or NEXT500) or by two or all models.

Event type	Base	NEXT100	NEXT500
Attack	0.91	0.84	0.87
Die	1.00	1.00	1.00
Elect	1.00	0.99	0.99
Demonstrate	1.00	0.98	0.98
Transport	0.99	0.99	1.00
Arrest-Jail	1.00	0.93	0.95
End-Position	0.95	0.95	0.97
Transfer-Money	0.99	0.97	0.98
Meet	1.00	1.00	1.00
Start-Position	1.00	0.95	0.97

Table 5: Event type classification precision on sampled predictions for baseline, NEXT100 and NEXT500 models.

NEXT500 models (or majority rule if a third fine-tuned model is used as well) as predictions shared between models tend to be more accurate compared to predictions made by individual models (Table with comparison is available in the Appendix).

Despite the large number of fine-tuning epochs for the new event type, this approach retains the model's capability of predicting existing event types. We showed that the majority of such predictions made by the baseline Text2Event model are also made by the fine-tuned models. Furthermore, the fine-tuned models generate two to four times as many predictions as the baseline model. This has only a minor effect on precision - a small drop in performance compared to the baseline model can be observed in Table 5. The largest drops in precision are by 0.07 for NEXT100 (Attack and Arrest-Jail) and 0.05 for NEXT500 (Arrest-Jail).

We attribute this rise in recall to the baseline model not having been trained or fine-tuned on samples from our claim debunks dataset. Though this dataset consists of texts from the same domain as ACE2005 (news media / publishing), the datasets differ on other parameters such as the average number of sentences per text. We observe that a few annotated samples for fine-tuning on one event type are sufficient to boost recall of all other event types.

7 Conclusion and further work

In this work we presented an approach to extend an existing event schema with new event types for closed-domain event extraction. Our approach uses a very small number of annotations containing full event structures (event type, trigger and arguments are all annotated).

The proposed approach also leads to improvement in the recall of existing event types, on which the model was pre-trained while retaining precision. It can thus be used not only to fine-tune the event extraction model for a new event type but to also simultaneously fine-tune the model for the existing event types on a new dataset without the need for annotation for all event types.

An interesting direction for future research would be evaluating whether this boost in performance would also be observed when the task is transferred to a dataset from a less related domain, e.g. biomedical, manufacturing, energy, etc. Further pre-training might also be of interest.

In terms of evaluation, it would be interesting to explore how our proposed approach compares to alternatives, such as open-domain approaches. Also, more documents from the initial dataset could be annotated for the original event types, in order to obtain a clearer picture of the baseline's model performance on them.

8 Acknowledgements

This research has received funding from the European Union's Horizon research and innovation programme under grant agreement No 101073921 (VIGILANT, https://www.vigilantproject.eu/).

References

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. ACM.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724, Online. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Florian Kunneman and Antal Van Den Bosch. 2016. Open-domain extraction of future events from twitter. *Natural Language Engineering*, 22(5):655–686.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45, Online. Association for Computational Linguistics.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics.

Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al. 2022. A survey on deep learning event extraction: Approaches and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Jiangwei Liu, Liangyu Min, and Xiaohong Huang. 2021. An overview of event extraction and its applications. *arXiv preprint arXiv:2111.03212*.

Xiao Liu, Heyan Huang, and Yue Zhang. 2019. Open domain event extraction using neural latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871, Florence, Italy. Association for Computational Linguistics.

Di Lu, Shihao Ran, Joel Tetreault, and Alejandro Jaimes. 2023. Event extraction as question generation and answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1666–1688, Toronto, Canada. Association for Computational Linguistics.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806, Online. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Hao Peng, Ruitong Zhang, Shaoning Li, Yuwei Cao, Shirui Pan, and Philip S. Yu. 2023. Reinforced, incremental and cross-lingual event detection from social messages. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):980–998.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112.

Jiawei Sheng, Shu Guo, Bowen Yu, Qian Li, Yiming Hei, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2021a. Casee: A joint learning framework with cascade decoding for overlapping event extraction. *arXiv preprint arXiv:2107.01583*.

Jiawei Sheng, Shu Guo, Bowen Yu, Qian Li, Yiming Hei, Lihong Wang, Tingwen Liu, and Hongbo Xu. 2021b. CasEE: A joint learning framework with cascade decoding for overlapping event extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 164–174, Online. Association for Computational Linguistics.

Hristo Tanev, Jakub Piskorski, and Martin Atkinson. 2008. Real-time news event extraction for global crisis monitoring. In *Natural Language and Information Systems: 13th International Conference on Applications of Natural Language to Information Systems, NLDB 2008 London, UK, June 24-27, 2008 Proceedings 13*, pages 207–218. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. *arXiv preprint arXiv:1909.03546*.

Qiang Wei, Zongcheng Ji, Zhiheng Li, Jingcheng Du, Jingqi Wang, Jun Xu, Yang Xiang, Firat Tiryaki, Stephen Wu, Yaoyun Zhang, et al. 2020. A study of deep learning approaches for medication and adverse drug event extraction from clinical text. *Journal of the American Medical Informatics Association*, 27(1):13–21.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Zhenrui Yue, Huimin Zeng, Mengfei Lan, Heng Ji, and Dong Wang. 2023. Zero- and few-shot event detection via prompt-based meta learning.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 414–419, Melbourne, Australia. Association for Computational Linguistics.

Yang Zhou, Yubo Chen, Jun Zhao, Yin Wu, Jiexin Xu, and Jinlong Li. 2021. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14638–14646.

A Appendix

A.1 Keywords

For the building of the Cure-Claim event type dataset, we have used the following keywords:

- cure, cures, cured, curing
- heal, heals, healed, healing
- treat, treats, treated, treating, treatment, treatments
- remedy, remedies
- relieve, relieves, relieved, relieving
- boost, boosts, boosted, boosting

In addition to the listed above keywords, the following ones were identified as triggers during the annotation process: stop, kill, prevent, regular.

A.2 Trigger classification evaluation

We classify trigger prediction as follows:

- TP (true positive) - event is annotated and prediction matches its type and trigger;
- TN (true negative) - no event is annotated and no event is predicted;
- FP (false positive) - no event is annotated but one is predicted OR event is annotated but predicted trigger does not match;
- FN (false negative) - event is annotated but none is predicted.

A.3 Argument classification evaluation

We consider the following four different scenarios:

1. An annotated event is predicted with the correct trigger.
2. An annotated event is predicted, but with a wrong trigger.
3. There is an annotated Cure-Claim event, but none is predicted. In this case we count the event and all its annotated arguments as false negatives.
4. There is no annotated Cure-Claim event, but one is predicted. In this case we count the event and all its predicted Cure-Claim arguments as false positives.

For the first two scenarios we report mean precision, recall and F1-score. In both cases we classify the argument prediction as follows:

- TP (true positive) - the argument prediction matches an annotated argument's type and span;
- FP (false positive) - the argument prediction matches an annotated argument's type but not span OR the argument prediction does not match any annotated argument's type;
- FN (false negative) - there is no argument prediction that matches an annotated argument's type and/or span.

We don't report true negative predictions for argument classification.

When event is annotated, but not predicted, we count all annotated arguments as false negative predictions. When event is not annotated, but is predicted, we count all predicted arguments as false positive predictions.

A.4 Additional results of baseline and fine-tuned models comparison

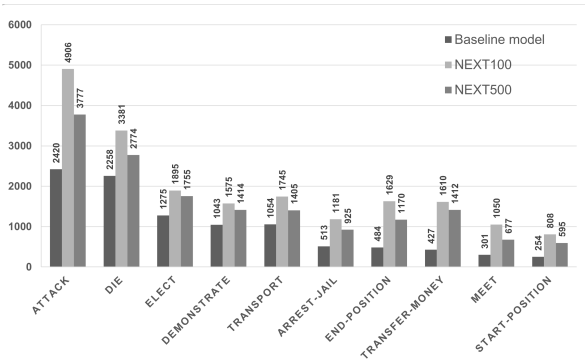

Figure 10: Number of documents containing predictions by baseline model, NEXT100 and NEXT500 (using claims as input)

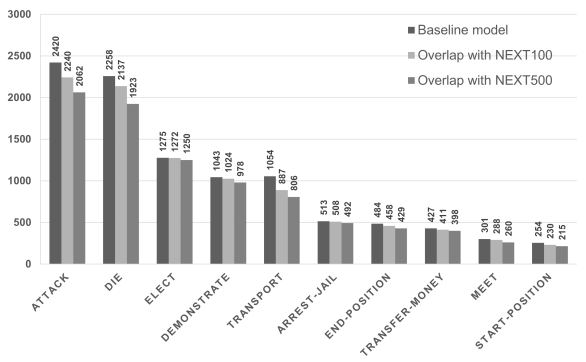

Figure 11: Number of documents containing predictions by baseline model and overlapped with NEXT100 and NEXT500 (using claims as input)

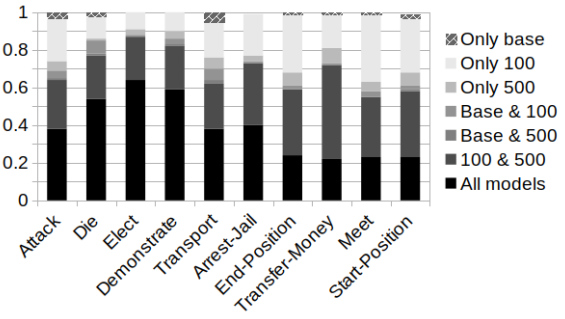

Figure 12: Overlap in event predictions between models (using whole claims as input). For each event type the bar shows what fraction of predictions is made by all three models, by two of the models, or by a single one.

Input	NEXT100 only	NEXT500 only	both
full set as sentences	0.53	0.33	0.60
full set as claims	0.50	0.27	0.47
subset as sentences	0.83	0.87	0.93
subset as claims	0.88	0.83	1.00

Table 6: Event type classification precision for CureClaim predictions made by NEXT100 only, NEXT500 only, or both models.