Rethinking the Event Coding Pipeline with Prompt Entailment

Clément Lefebvre*
Swiss Data Science Center
clement.lefebvre@datascience.ch

Niklas Stoehr*
ETH Zürich
niklas.stoehr@inf.ethz.ch

Abstract

For monitoring crises, political events are extracted from the news. The large amount of unstructured full-text event descriptions makes a case-by-case analysis unmanageable, particularly for low-resource humanitarian aid organizations. This creates a demand to classify events into event types, a task referred to as event coding. Typically, domain experts craft an event type ontology, annotators label a large dataset and technical experts develop a supervised coding system. In this work, we propose PR-ENT[1], a new event coding approach that is more flexible and resource-efficient, while maintaining competitive accuracy: first, we extend an event description such as "Military injured two civilians" by a template, e.g. "People were [Z]" and prompt a pre-trained (cloze) language model to fill the slot Z. Second, we select suitable answer candidates Z^* = {"injured", "hurt"...} by treating the event description as premise and the filled templates as hypothesis in a textual entailment task. In a final step, the selected answer candidate can be mapped to its corresponding event type. This allows domain experts to draft the codebook directly as labeled prompts and interpretable answer candidates. This human-in-the-loop process is guided by our codebook design tool[2]. We show that our approach is robust through several checks: perturbing the event description and prompt template, restricting the vocabulary and removing contextual information.

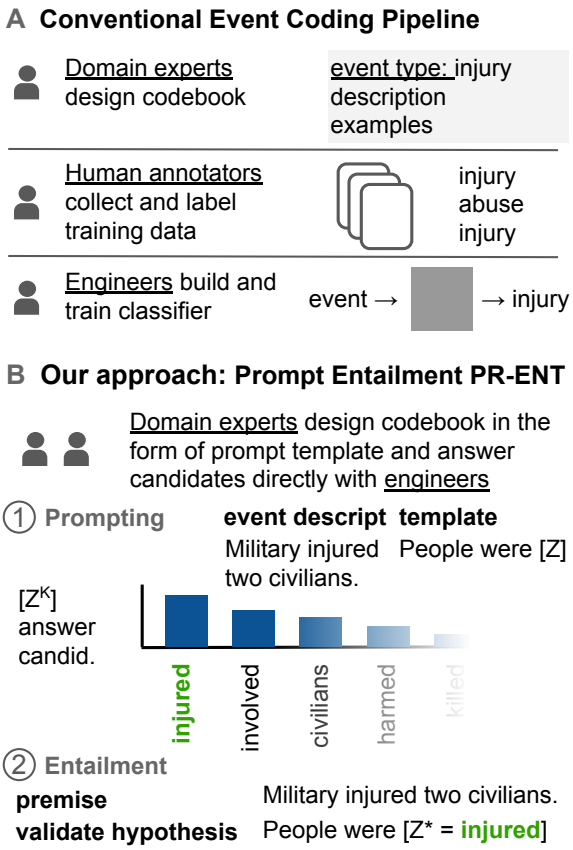

Figure 1: **(A)** The conventional event coding pipeline involves many hand-overs between involved stakeholders and is strictly tailored to the event ontology. **(B)** Our approach combines prompting and textual entailment to perform flexible, unsupervised event coding.

1 Introduction

Decision-makers in politics and humanitarian aid report a growing demand for comprehensive and structured overviews of socio-political events (Lepuschitz and Stoehr, 2021). For this purpose, news papers are automatically screened for event mentions, a task referred to as *event detection* and *extraction*. The sheer amount of extracted, full-text event descriptions day-to-day is impossible to be parsed by humans, especially when limited by scarce financial and computational resources.

Event coding seeks to automatically classify event descriptions into pre-defined event types. Event coding is conventionally approached via a multi-step pipeline as shown in Fig. 1A. It incurs large costs in terms of human labor and time. We sketch out this pipeline expressed in *human intelli-*

*authors contributed equally
[1]https://huggingface.co/spaces/clef/PRENT-Demo
[2]https://huggingface.co/spaces/clef/PRENT-Codebook

Proceedings of the Sixth Fact Extraction and VERification Workshop (FEVER), pages 1–16

gence tasks (HITs)[3] (ul Hassan et al., 2013).

As a first step, an *event ontology* is defined in terms of a codebook. Codebook development requires multiple domain experts (Goldstein, 1992) spending up to 200 HITs. The initial development phase of the widely-used Conflict and Mediation Event Observations (CAMEO) (Schrodt, 2012) codebook reports a 3-year initial development phase. Next, context-relevant event descriptions need to be collected to serve as training data. This often requires paid access to online newspaper distribution services and data collection infrastructure, estimated at 200 HITs. Next, human annotators need to be recruited and trained to annotate data according to the codebook accounting for another 200 HITs. Finally, a machine-based coding system needs to be developed, trained and validated, costing another 200 HITs. In earlier days, systems were dictionary- and pattern- based (King and Lowe, 2003; Norris et al., 2017), while more recently machine learning-based approaches have gained momentum (Piskorski and Jacquet, 2020; Olsson et al., 2020; Hürriyetoğlu, 2021).

In total, the conventional event coding pipeline amounts to roughly 800 HITs. This development cost is often not bearable by non-profit / non-governmental organizations in the humanitarian aid sector. Moreover, the process requires multiple hand-overs between workers of different background which leads to errors, misunderstanding and delays. It is also important to highlight that the developed coding system is specifically tailored to a fixed event ontology. Any post-hoc changes of event types or even a different dataset incurs huge costs. In practice, event types frequently change and even vary widely between different divisions of the same organization.

To address these shortcomings, we present a new paradigm for highly adaptive event coding. Based on our method illustrated in Fig. 1B, domain experts are able to work directly with an interactive coding tool to design a codebook. They express event types by means of prompt templates and single-token answer candidates. For automated coding, a pre-trained language model is prompted to fill in those answer candidates taking a full-text event description as an input. Since prompting can be noisy (Gao et al., 2021), we propose filtering answer candidates based on textual entailment. Specifically, our contributions are as follows: (1) We propose a methodology combining prompting (§3.1) and textual entailment (§3.2) for event coding, termed PR-ENT. (2) We thoroughly evaluate this paradigm based on three aspects: accuracy (§4.1), flexibility (§4.2) and efficiency (§4.3). (3) We present two online dashboards: (a) A demo of the PR-ENT coding tool. (b) An interactive codebook design tool that guides the codebook design by presenting accuracy validation in a human-in-the-loop manner (§6).

2 Event Data and Types

We consider a subset of the Armed Conflict Location and Event Data (ACLED) (Raleigh et al., 2010) dataset. It is widely-used and has large coverage of political violence and protest events around the world. Each event is human annotated with a short description, its event type and additional details such as the number of fatalities and actor and targets. The event types are based on ACLED's own event ontology which distinguishes 6 higher-level and 25 lower-level event types. Some event types are easily separable (e.g. *protests* vs *battles*), while others are harder to distinguish semantically (e.g. *protests* vs *riots*) (see Fig. 9 in the appendix).

We sample 4000 ACLED events (3000 for training, 1000 for testing) in the African region while maintaining the event type distribution of the full dataset (see Fig. 9). We remove empty event descriptions and annotator notes (e.g. "[size: no report]"). In Fig. 8 in the appendix, we present statistics of the test set, showing different aspects of linguistic complexity. In §4.2, we consider the Global Terrorism Dataset (GTD) (LaFree and Dugan, 2007) to study the effect of domain shift.

3 Entailment-based Prompt Selection

Our approach, PR-ENT, represents a real-world use case of prompting and textual entailment to code event descriptions $e \in \mathcal{E}$ into event types $y \in \mathcal{Y}$ as shown in Fig. 1B.

3.1 Prompting

Methodological Approach. In traditional supervised learning, a model is trained to learn a mapping between the input e and the output class y. *Prompting* (Liu et al., 2021) is a learning paradigm

[3] In our formulation, one HIT corresponds to roughly one hour of low-skill work by a single person such as reading and labeling single-sentence event descriptions. Our estimations are based on practical experience in working with domain experts and human annotators in the field of political event coding and serve the purpose of providing a very approximate quantification of required resources and labour.

making use of (cloze) language models that have been trained to predict masked tokens within text.[4] Prompt-based learning transfers this capability to perform classification in the following way:

We extend each *event description* $e \in \mathcal{E}$ by a *template* $t \in \mathcal{T}$ to form the input $\langle e, t \rangle \in \mathcal{E} \times \mathcal{T}$. Each template contains a *masked slot* Z, e.g. "This event involves [Z]", "People were [Z]".[5] The language model takes $\langle e, t \rangle$ as input and returns an *output distribution* of probabilities over the *answer vocabulary* $\mathcal{Z}$. Each token $z_{e,t} \in \mathcal{Z}$ can serve as a potential slot filler to $Z = z_{e,t}$. However, we only consider the top k most probable *answer candidates* $z^k_{e,t} \in \mathcal{Z}^K_{e,t}$. $\mathcal{Z}$ can be a constrained subset $\mathcal{Z}_t$ that only features a template-related answer vocabulary to increase interpretability as pointed out in §5. We discuss how to map answer candidates to event types in §4.1.

Implementation Details. We discuss the design of templates and constrained answer vocabularies resulting in a codebook (Tab. 7) in §6. In particular, we prompt DistilBERT-base-uncased (Sanh et al., 2020), a *distilled* version of the BERT model which is more computationally efficient at the cost of a small performance decrease. For each prompt, we consider the $K = 30$ most probable tokens as the set of answer candidates $\mathcal{Z}^K_{e,t}$. Ideally, we select a larger set, but performance gains are minimal while computational costs increase in subsequent steps.

3.2 Textual Entailment

Limitations of Prompting. Prompting yields event-related tokens for event coding, but comes with challenges. There is no guarantee that a prompted answer candidate $z^k_{e,t} \in \mathcal{Z}^K_{e,t}$ is suited to represent an event. Answer candidates may be semantically unrelated as shown in Fig. 2. To address this shortcoming, we propose filtering $\mathcal{Z}^K_{e,t}$ via textual entailment. Textual entailment, or natural language inference (NLI) (Fyodorov et al., 2000; Bowman et al., 2015) can be framed as the following task: Given a "premise", verify whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral). It has been evaluated as a popular method for performing text classification (Wang et al., 2021).

[4] "Cloze" pertains to filling in missing tokens not necessarily uni-directional left-to-right, but anywhere in a string.

[5] The first prompt template is intended to provide a one-word summary of the event. For the second template, we expect a verb describing the actions undertaken by the actor or a verb that describes what happened to the target.

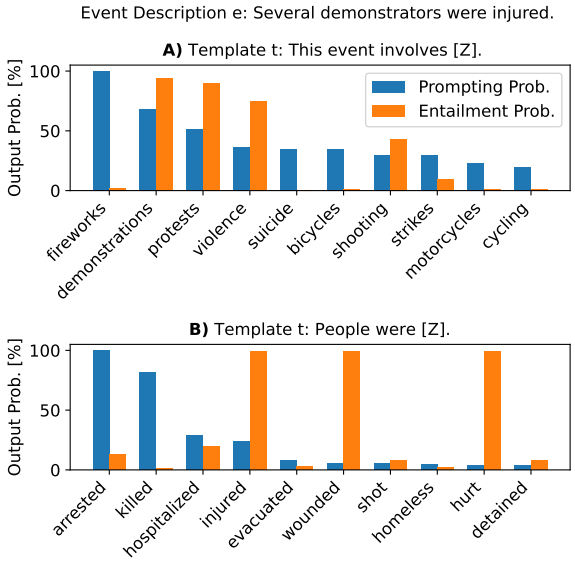

Figure 2: Given the event description "Several demonstrators were injured." and two templates **(A)** and **(B)**, prompting alone can yield tokens that fit syntactically but not semantically (blue bar). In contrast, filtering prompted answer candidates via textual entailment leaves us with tokens more closely related to the event (orange bar). To this end, we treat the event description as premise and the filled template as hypothesis.

Selecting Entailed Answer Candidates. We consider the event description e as premise and the template t' filled with a prompted answer candidate as hypothesis. For example, given the premise "Two bombs detonated...", we automatically construct hypotheses "This event involves $[z^k_{e,t}] \in \mathcal{Z}^K_{e,t} = \{$explosives, civilians...$\}$", see Tab. 1. We pass the concatenation of the premise and hypothesis to RoBERTa-large-mnli (Liu et al., 2019). If the model finds premise and hypothesis to be entailed, then the prompted answer candidate $z^k_{e,t}$ is considered an *entailed answer candidate* $z^*_{e,t}$ (e.g. $z^*_{e,t}$ = explosives). We combine the categories "neutral" and "contradiction" into one since we are interested in a hypothesis being entailed or not.

This means, PR-ENT has two hyperparameters: the top K answer candidate tokens yielded by the prompting step and the acceptance threshold in the entailment step that governs whether an answer candidate is kept. We empirically analyse the effect of both hyperparameters on the final F1 classification score in Fig. 5. In Fig. 5A, we verify that considering the top 30 answer candidate tokens leads to good performance on average. Further, we find a suitable threshold of 0.5 on the entailment model's output probability in Fig. 5B.

Event Description + Template $\langle e, t \rangle$	Answer Candidates $z^k_{e,t}$	Entailed Answer Candidates $z^*_{e,t}$
Several demonstrators were injured. + People were [Z].	arrested, killed, hospitalized, injured, evacuated, wounded, shot, homeless, hurt, detained	injured, wounded, hurt
Several demonstrators were injured. + This event involves [Z].	fireworks, demonstrations, protests, violence, suicide, bicycles, shooting, strikes, motorcycles, cycling	demonstrations, protests, violence
The sponsorship deal between the shoes brand and the soccer team was confirmed. + This event involves [Z].	sponsorship, nike, sponsors, fundraising, cycling, advertising, charity, donations, concerts, competitions	sponsorship, sponsors, advertising, competitions

Table 1: We prompt a language model based on an event description e and template t with slot Z. We keep only those prompted answer candidates $z^k_{e,t} \in \mathcal{Z}^K_{e,t}$ entailed in a subsequent textual entailment task $z^*_{e,t} \in \mathcal{Z}^*_{e,t}$.

4 Evaluation: Event Classification

We compare PR-ENT against the conventional event coding pipeline in an evaluation along three dimensions: accuracy, flexibility and efficiency.

4.1 Accuracy

So far we have not discussed how to map entailed answer candidates $z^*_{e,t} \in \mathcal{Z}^*_{e,t}$ onto event types $y \in \mathcal{Y}$. We choose to do *hard* prompting, as opposed to *soft* prompting. This means, tokens in $\mathcal{Z}^*_{e,t}$ are mapped onto event types y via a simple linear transform $y = f(z^*_{e,t})$. When f is the identity function, no additional mapping is needed (§4.2). Hard prompting allows defining event types, i.e. an event ontology, in terms of interpretable answer candidates. As an example, we present an interpretable event ontology in Tab. 7 in the appendix. We use it to classify "lethal" and "non-lethal" event as explained in §4.2. Generally, we observe a trade-off between accuracy and interpretability. We want different sets of entailed answer candidates to uniquely define different event types at a high accuracy. At the same time, we require the set to be limited to a few, interpretable tokens only, that are highly representative for the event type. In the following, we learn a shallow mapping between $\mathcal{Z}^*_{e,t}$ and the 6 high-level event types $\mathcal{Y}$ provided by the ACLED event ontology as ground truth.

Baselines and Ceilings. As baselines, we consider *bag-of-words* (BoW) and GloVe (Pennington et al., 2014) embeddings of event descriptions. Embeddings are mapped onto event types via logistic regression (LR). Further, we contrast our PR-ENT with a prompting-only (PR) approach also using logistic regression as a classification layer. As a ceiling model, we consider DistilBERT fine-tuned in a sequence classification task.

Model	Accuracy	F1 Score
BoW + LR	80.5	77.1
GloVe + LR	78.5	74.6
Random Tokens + BoW + LR	77.1	72.2
PR + BoW + LR	82.9	80.8
PR-ENT + BoW + LR	85.1	83.7
DistilBERT	**87.1**	**86.0**

Table 2: Classification of 6 event types in the ACLED dataset. As expected, DistilBERT performs best as it is fine-tuned specifically on this classification task. Our approach PR-ENT is more ad-hoc and does not fall far behind. The additional entailment step reduces noise compared to the prompting-only approach PR. On top of the two standard baselines using BoW and GloVe, we introduce an additional baseline where we select 10 random tokens from $\mathcal{Z}^K_{e,t}$ for each $\langle e, t \rangle$. Compared to all baselines, PR-ENT performs better.

Our Approach PR-ENT. To evaluate our approach, we only consider the template "This event involves [Z]" and construct a BoW feature matrix by extending the event descriptions e with the entailed answer candidates $z^*_{e,t}$. The resulting feature matrix serves as input to logistic regression. We report classification results in Tab. 2 and find that PR-ENT is only outperformed by the supervised, fine-tuned DistilBERT ceiling, but performs better than all baselines.

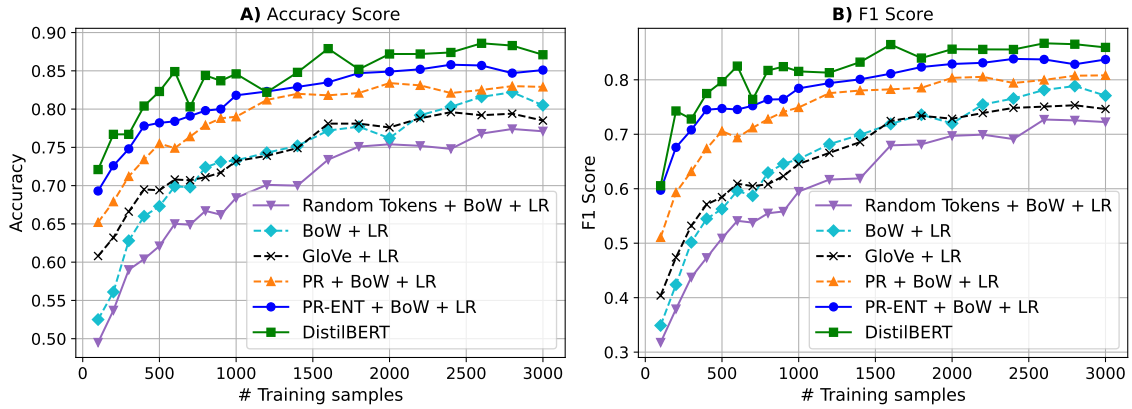

Figure 3: Comparison of the different classification approaches on a varying number of training instances. Our approach PR-ENT shows better performance in terms of accuracy and F1 Score than the baseline models at all points. At the same time, it does not lack far behind the fine-tuned DistilBERT ceiling model, which is however less flexible and resource-intensive. PR refers to prompting-only, BoW to bag-of-words and LR to logistic regression. The baseline "random" consists of sampling 10 random tokens from $\mathcal{Z}_{e,t}^{K}$ for each $\langle e, t \rangle$.

4.2 Flexibility

We explore the flexibility of PR-ENT along 3 dimensions: changing the number of training instances, omitting the shallow mapping for classification and switching to another dataset.

Number of Training Instances. As can be seen in Fig. 3, our approach shines at classifying event types if only few training instances are given. PR-ENT shows better performance than all baseline approaches introduced in §4.1. At the same time, it is not far behind the fine-tuned DistilBERT ceiling model.

Removing the Shallow Mapping. We may remove the requirement of adding a shallow mapping $y = f(z_{e,t}^{*})$. Therefore, we predict if an event is "lethal" ($y = 1$) or not ($y = 0$) based on its description. We use PR-ENT to generate entailed answer candidates $\mathcal{Z}_{e,t}^{*}$ based on the template "People were [Z].". If Z = "killed" $\in \mathcal{Z}_{e,t}^{*}$ then $y = 1$. We compare PR-ENT against fine-tuned DistilBERT trained on 100 samples and present results in Tab. 3. PR-ENT is competitive against DistilBERT, even outperforming it in this setting. Moreover, while the prompting-only approach (PR) has very high recall, it lacks precision. The additional entailment step in PR-ENT balanced this out, yielding a high F1 score.

Domain Shift. We scrutinize the robustness of PR-ENT by switching to another dataset. We repeat the binary "lethal versus non-lethal" classification task on the *Global Terrorism Database* (GTD) (LaFree and Dugan, 2007). The results in Tab. 4, again suggest high performance of PR-ENT.

Model	F1 Score	Precision	Recall
PR-ENT	**91.6**	**85.3**	98.8
Prompting Only	50.6	33.9	**100**
DistilBERT	84.1	76.5	93.4

Table 3: Binary classification of "non-lethal versus lethal" events based on ACLED's fatality counts. In PR-ENT and prompting-only PR, we code "lethal" if "killed" is present in the answer candidates of "People were [Z].". We observe the added value of the entailment step in the increase in precision. PR-ENT outperforms DistilBERT trained on 100 data instances and tested on 1000 event descriptions.

Model	F1 Score	Precision	Recall
PR-ENT	**96.3**	**94.0**	98.8
Prompting Only	67.3	50.7	**100**
DistilBERT	93.4	89.9	97.2

Table 4: Binary classification of "non-lethal versus lethal" based on the Global Terrorism Database (GTD). PR-ENT and prompting-only PR predict "lethal" if "killed" is prompted from "People were [Z].". PR-ENT outperforms DistilBERT trained on 100 data instances and tested on 1000 event descriptions.

4.3 Efficiency

In §1, we estimated the cost of 800 human intelligence tasks (HIT) for the conventional event coding pipeline. We perform the same estimation exercise for our approach: domain experts design suitable

Perturbation Type	Paraphrase		Remove Stop Words		Remove Entities		Duplication	
Model Type	PR	PR-ENT	PR	PR-ENT	PR	PR-ENT	PR	PR-ENT
1 Perturbation	0.33	**0.14**	0.22	**0.15**	0.15	**0.08**	0.18	**0.09**
2 Perturbations	0.34	**0.18**	-	-	-	-	0.28	**0.16**

Table 5: Average Jensen-Shannon distance across 1000 event descriptions. We conduct 4 perturbation tests: paraphrasing the template, removing stop words from the event description, replacing named entities by a placeholder, and duplicating words in the template. PR-ENT is more robust than PR: in all cases, the distance between the output distributions based on the non-perturbed and perturbed input is smaller.

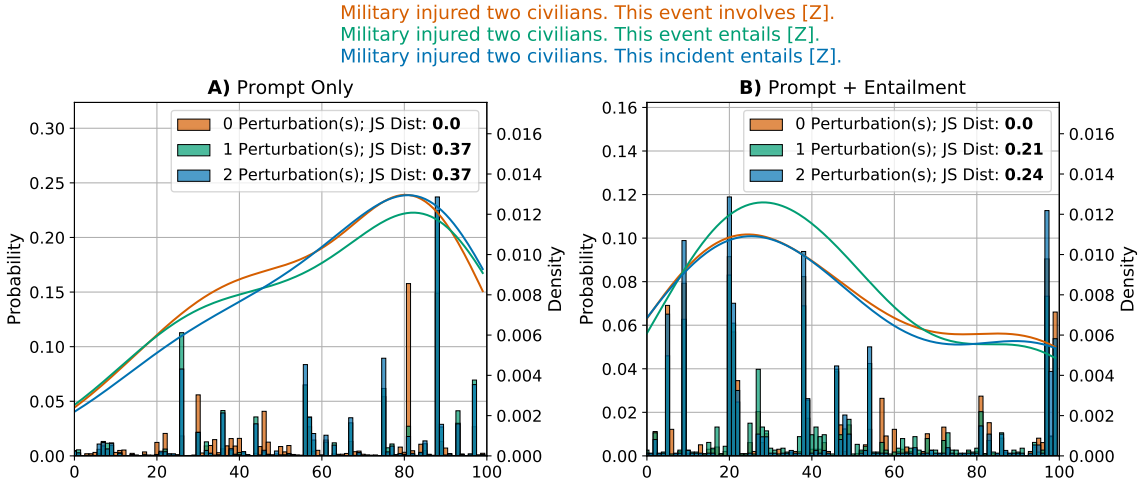

Figure 4: We compare prompting-only PR and our approach PR-ENT when perturbing the input $\langle e, t \rangle$. PR-ENT is more robust to perturbations as indicated by a lower Jensen-Shannon distance between the output distributions over answer candidates based on non-perturbed and perturbed input. PR is highly sensitive to template phrasing. X-label represents the top 100 most frequent tokens from 1000 prompts.

templates and answer candidate sets in a trial and error fashion as elaborated in §6. We estimate total development costs at about 300 HITs, which makes it particularly feasible for small teams with few resources such as non-governmental organizations in humanitarian aid. Overall, our approach requires fewer people and consequently fewer hand-overs. Moreover, it is not tied to a specific event ontology and more flexible for changing event types.

5 Ablation Study

5.1 Perturbation Tests

Our approach is not tailored to a specific event ontology, but to a language model. Any performance gains on these models, such as the recently published ConfliBERT (Hu et al., 2022), will impact our pipeline. A crucial consideration is the presence of biases within language models. In some settings, biases may even be desirable inductive priors, but should at least be known.

We measure the sensitivity of the prompted model's output distribution to changes in the input. To this end: we select a fixed answer vocabulary $\mathcal{Z}_t$ of 100 tokens by taking the most frequent tokens yielded by the prompted model across 1000 event descriptions. We observe the output distribution over tokens in $\mathcal{Z}_t$ before and after perturbing the input $\langle e, t \rangle$. Finally, we measure the difference between the two output distributions in terms of Jensen-Shannon (JS) distance. We show the results of the following four perturbation settings in Tab. 5:

(1) Paraphrasing Two prompt designers could come up with paraphrased templates. In Fig. 4, we show that the additional entailment step makes PR-ENT more robust to perturbations in the template as opposed to prompting only.

(2) Stop Word Removal We remove stop words from the event description to test PR-ENT on non-grammatical text.

(3) Context Removal We remove all named entities in event descriptions and replace them with placeholder tokens such as "organizations" and "locations". This verifies that PR-ENT is less prone to latching onto context instead of content.

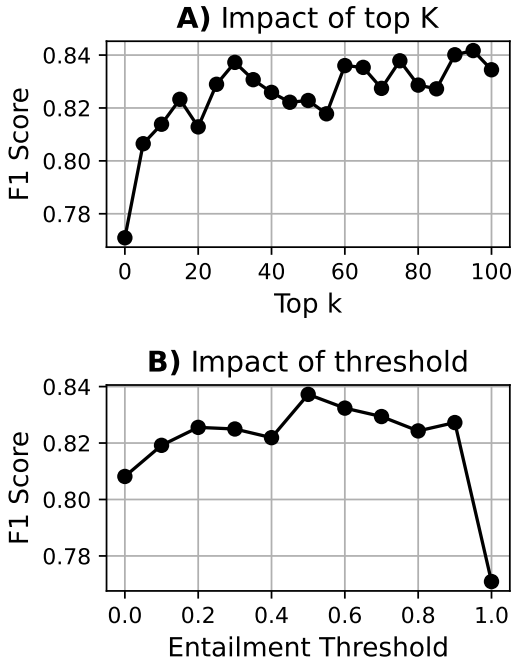

Figure 5: Impact of different parameters of our pipeline on ACLED classification. **(A)** F1 score versus the maximum number K of allowed answer candidates; $K = 0$ means that only the event description is used in the classification. **(B)** F1 score versus entailment threshold; the threshold governs if a hypothesis is entailed with the premise or not, a threshold of 0 means that all prompted answer candidates are considered. A threshold of 1 means only the event description is considered.

(4) Duplication We duplicate some words in the template. Specifically, we test the 3 prompts: "This event involves [Z]", "This event event involves [Z]", "This event event event involves [Z]".

5.2 Comparing Coded Event Time Series

Using PR-ENT, we construct a codebook (Tab. 7) to code ACLED event descriptions without the need of a shallow mapping. We use this codebook to code events that took place in Mali (Fig. 6) and Ethiopia (Fig. 6) between 2009 and 2021. This allows comparing time series of event types between our approach and ACLED's coding. We find that both codings yield very similar time series in which the positioning of spikes align. Yet, the spikes in the PR-ENT time series are higher / steeper indicating that more events are detected. This may be attributed to two reasons: firstly, PR-ENT is potentially more granular and has higher recall. Secondly, PR-ENT is not limited to coding only one event type per event description as ACLED is. For example, the following event description in ACLED (anonymized) is coded as Armed Clash but contains several possible event types (Armed Clash, Killing, Kidnapping, Property Destruction, Looting): "[...] The militants clashed with [ORG], and killed one [ORG] and a civilian driver, abducted one person, burned a vehicle and seized livestock."

5.3 Qualitative Error Analysis

We perform a qualitative error analysis of our proposed method. Within the ACLED data, there are many event descriptions containing mentions of past events (e.g. "Protests over the killing of the journalist [NAME] shot dead on Monday at his home by armed bandits."). Our method, and in fact, any supervised classifier, may have difficulties recognizing event co-references. Another frequent error is due to ACLED event type definitions. For instance, ACLED features the event type "Violence Against Civilians". However, to classify most of the concerned events, the annotator needs to know if the target is a civilian or not. Unfortunately, the dataset does not always contain this information, except if explicitly written in the event description. Another frequently observed error is caused by blurry definition of event types. ACLED, differentiates between "Riots" and "Protests" which often have nearly identical event descriptions.

6 Human-Computer Codebook Design

To make use of PR-ENT, domain experts need to design a codebook (i.e. a mapping), between event types and entailed answer candidates. Creating this mapping is non-trivial as there exists a trade-off between interpretability and accuracy. In essence, a codebook is interpretable when the answer candidates are representative of the corresponding event type. A bad codebook contains a large number of non-readable entailed answer candidates. A codebook is accurate when a few answer candidates are sufficient to allow for a clear differentiation of the event types. To that end, we propose an interactive codebook design tool[6] that helps designing templates and answer candidates by presenting accuracy metrics. The assessment of interpretability is left to the human domain experts.

Codebook Design. Our codebook is a mapping between event types and entailed answer candidates. For example, an event can be classified as "kidnapping" if any of the following templates is

[6] https://huggingface.co/spaces/clef/PRENT-Codebook

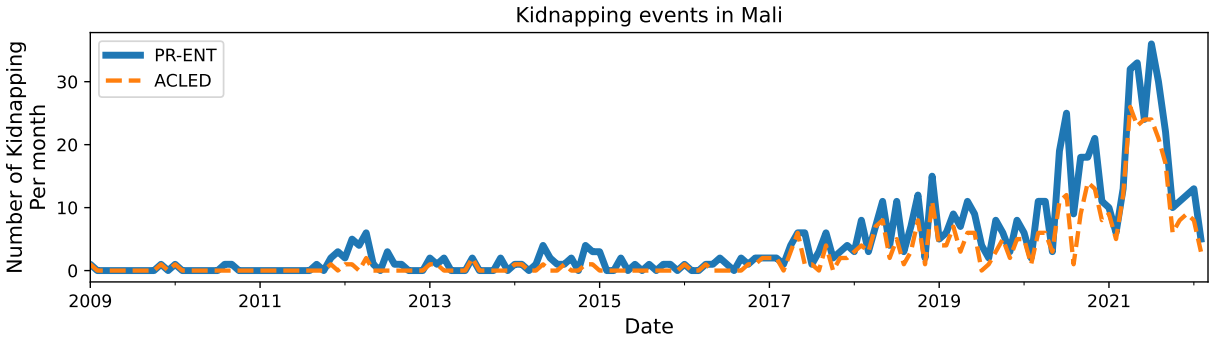

Figure 6: Time series of the number of kidnapping events per month in Mali between 2009-2021. The dashed line corresponds to all kidnapping events coded by ACLED annotators. The blue line corresponds to all kidnapping events coded by PR-ENT. We find that the positions of the time series spikes between PR-ENT and ACLED's coding align well. However, the spikes in the PR-ENT time series are higher indicating that PR-ENT detects more events. This may be due to more granular event coding or the advantage of not being limited to only one event type per event description.

entailed: "This event involves [kidnapping]." `OR` "This event involves [abduction].". A codebook example is shown in Tab. 7 in the appendix.

We assume two things: first, the availability of a dataset which contains event descriptions that need to be labeled. Second, the domain experts should have decided upon event types of their liking (e.g. kidnapping, killings,...). Now, the first step is to come up with an initial set of templates and entailed answer candidates. For each event type, the domain expert is asked to draft a canonical event description. For example: the event type "kidnapping" could be exemplified by "Two men were kidnapped by rebels.". Then using PR-ENT, the domain expert is presented a list of answer candidates (e.g. "This event involves [kidnapping].", "This event involves [rebels].”...).

As a second step, domain experts select some of the entailed answer candidates provided by the model. If no entailed answer candidate is informative to classify the event, it is possible to group multiple entailed answer candidates with an `AND` condition. For example, "Riot" event types can be coded with the two following templates: "This event involves [protest]." `AND` "This event involves [violence].". The tool also offers the possibility of excluding certain answer candidates.

On-the-Go Validation. Validating the interpretability of the codebook and the answer candidates is a subjective task that we leave to the domain experts. The coding tools offers however guidance for the validation of accuracy, despite not having access to ground truth event type labels. Using the current state of the codebook and PR-ENT, randomly selected events are automatically coded into event types. Domain experts can then accept or reject the event type suggestions provided by the model. This creates a labeled dataset "on the go", which allows computing a per-class accuracy score. Repeated rounds of validation allow for a human-in-the-loop fine-tuning of the codebook by adding or removing more entailed answer candidates.

Codebook Use. The tool offers interoperability by enabling the download of the codebook and the labeled dataset in standard JSON format. The former can then be used to code a full dataset of event descriptions into event types. The codebook can still be modified if more event types are required.

7 Discussion

Is this few-shot, unsupervised tagging? While we have evaluated accuracy, efficiency and flexibility, it is up for discussion and definition whether our approach should be considered few-shot, unsupervised or tagging-based. In some cases, the language model copies tokens verbatim from the input, which could be seen as a form of "event tagging". In other cases, the answer candidates are abstract tokens outperforming purely tagging-based approaches. In cases where the answer candidates map directly to an event type without an additional shallow classifier §4.2, our approach may be considered unsupervised and zero-shot. On the contrary, the template is designed in an iterative trial and error fashion. Thus, it is tuned to observed data instances which arguably violates the zero-shot setting and should be framed few-shot instead.

Entailment-Only Approach. The presented approach PR-ENT relies on textual entailment to select entailed answer candidates from prompts as motivated in §3.2. However, textual entailment could have been considered for classification by itself (Wang et al., 2021; Barker et al., 2021). In this setting: a predefined set of hypotheses is created for each event type and is tested against each event description. However, this reduces flexibility as we need to define a broad set of hypotheses in advance. Our prompting-based approach relies on large language models which do not require labeled training data for training. As a consequence, they are more frequently updated and trained on larger amounts of data.

Extensions and Applications. Our approach can be used to filter and search events in a dataset of full-text event descriptions. An example of this use case is described in §4.2 where we classify lethal and non-lethal events in an unsupervised way via the "killed" token. Promising extension are the coding of source and target actors in addition to event types as presented in App. B.1 as well as the extraction of victim counts (Zhong et al., 2023).

8 Related Work

Similar to our prompting-based approach, existing work evaluates off-the-shelf QA (Halterman et al., 2021) and NLI (Barker et al., 2021) models for event coding. The prompting approach shares similarities with Shin et al. (2021), who build a semantic parser to map natural text to canonical utterances. Their training set is constructed by prompting a language model in a human-in-the-loop fashion. Sainz et al. (2021) uses NLI to extract relationship between two given entities based on a predefined hypothesis template. Schick et al. (2020) present an approach to identify words that can serve as high-accuracy labels for text classification. However, they are not focusing on interpretability and a particular application domain such as political event coding. There also exist methods for automating prompt generation and selective incorporation of examples in the prompt (Shin et al., 2020; Gao et al., 2021). Existing work in prompt-based classification focuses on sentiment, topic or intent (Yin et al., 2019; Liu et al., 2021; Schick and Schütze, 2021).

Within the field of event coding, we distinguish work on event detection, event type ontologies, and automated event coding tools. Our work falls into the latter two. The World Event/Interaction Survey (WEIS) project (McClelland, 1984) was pioneering in event data collection and event ontology design. The WEIS successor CAMEO (Schrodt, 2012) is one of the most popular event ontologies until today and used by ICEWS (Boschee et al., 2015) and NAVCO (Lewis et al., 2016) among others. VRA-Reader (King and Lowe, 2003) is among the first to automatize event coding based on matching string patterns. Its successors BBN ACCENT (Boschee et al., 2015), Tabari and Petrarch2 (Norris et al., 2017) rely on lambda calculus-based semantic parsing. Recent event coding systems rely on supervised machine learning (Hürriyetoğlu, 2021; Stoehr et al., 2021, 2022, 2023), word embedding- (Kutuzov et al., 2017; Piskorski and Jacquet, 2020) and transformer-based models (Olsson et al., 2020; Re et al., 2021; Hu et al., 2022; Skorupa Parolin et al., 2022).

9 Conclusion

We proposed a method to select answer candidates from prompts using textual entailment. This combined usage of state-of-the-art tools is motivated by a real-world use case that benefits humanitarian aid efforts with scarce resources.


https://github.com/Clement-Lef/pr-ent

Acknowledgments

This work was funded by the ETH4D Humanitarian Action Challenge and grew out of a collaboration with Roberto Castello, Silvia Quarteroni, Daniel Gatica-Perez and Sandro Saitta. We would like to thank Fiona Terry, Francesca Grandi and Chiara Debenedetti from the International Committee of the Red Cross (ICRC) for feedback and discussions that motivated this research project. Niklas Stoehr is supported by a scholarship from the Swiss Data Science Center (SDSC).

Limitations

We explore potential failure modes and the impact of bias in pre-trained (cloze) language models in §5. Erroneous event coding can be further mitigated through incorporation of confidence score. In §7, we discuss definitional caveats and model limitations. We make our code and interactive dashboard available for replication and scrutiny by the scientific community. We provide hyperparameter set-

tings, training times and details on the computing infrastructure in the appendix (App. A). Since we are only considering off-the-shelf models, mostly without further fine-tuning, our experiments can be reproduced with limited computing resources. Our experiments are limited to English language, but can be extended by considering models pre-trained on other language data.

Impact Statement

As explained in §1, our approach is aimed at helping low-resource organizations to analyze large amounts of text data efficiently. We do not foresee risk of misuse beyond the risks already introduced by conventional event coding pipelines. However, we would like to emphasize that the intended use of our approach is to gain additional, empirical insights for research and monitoring purposes, rather than for completely automatized decision-making. Application cases such as filtering event datasets are described in §7 and App. B.1 .

References

Ken Barker, Parul Awasthy, Jian Ni, and Radu Florian. 2021. IBM MNLP IE at CASE 2021 Task 2: NLI Reranking for Zero-Shot Text Classification. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 193–202, Online. Association for Computational Linguistics.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS Coded Event Data.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Yaroslav Fyodorov, Yoad Winter, and Nissim Francez. 2000. A Natural Logic Inference System.

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*, pages 3816–3830.

Joshua Goldstein. 1992. A conflict-cooperation scale for WEIS events data. *The Journal of Conflict Resolution*, 36(2):369–385.

Andrew Halterman, Katherine Keith, Sheikh Sarwar, and Brendan O'Connor. 2021. Corpus-Level Evaluation for Event QA: The IndiaPoliceEvents Corpus Covering the 2002 Gujarat Violence. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP*, pages 4240–4253, Online.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653.

Yibo Hu, MohammadSaleh Hosseini, Erick Skorupa Parolin, Javier Osorio, Latifur Khan, Patrick T. Brandt, and Vito J. D'Orazio. 2022. ConfliBERT: A pre-trained language model for political conflict and violence. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*.

Ali Hürriyetoğlu, editor. 2021. *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*. Association for Computational Linguistics.

Gary King and Will Lowe. 2003. An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57(3):617–642.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. 2017. Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop*, pages 31–36.

Gary LaFree and Laura Dugan. 2007. Introducing the Global Terrorism Database. *Terrorism and Political Violence*, 19(2):181–204. Publisher: Routledge.

Raphael Lepuschitz and Niklas Stoehr. 2021. SeismographAPI: Visualising temporal-spatial crisis data. *KDD Workshop on Data-Driven Humanitarian Mapping*, 2107.12443(arXiv).

Orion A. Lewis, Erica Chenoweth, and Jonathan Pinckney. 2016. Nonviolent and violent campaigns and outcomes 3.0: Effects of tactical choices on strategic outcomes codebook.

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*, 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*, 1907.11692.

Charles McClelland. 1984. World event/interaction survey (WEIS) project, 1966-1978: Archival version.

Clayton Norris, Philip Schrodt, and John Beieler. 2017. Petrarch2: Another event coding program. *The Journal of Open Source Software*, 2.

Fredrik Olsson, Magnus Sahlgren, Fehmi ben Abdesslem, Ariel Ekgren, and Kristine Eck. 2020. Text Categorization for Conflict Event Annotation. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Jakub Piskorski and Guillaume Jacquet. 2020. TF-IDF Character N-grams versus Word Embedding-based Models for Fine-grained Event Classification: A Preliminary Study. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Clionadh Raleigh, Andrew Linke, Håvard Hegre, and Joakim Karlsen. 2010. Introducing ACLED-Armed Conflict Location and Event Data. *Journal of Peace Research*, 47(5):651–660.

Francesco Re, Daniel Vegh, Dennis Atzenhofer, and Niklas Stoehr. 2021. Team "DaDeFrNi" at CASE 2021 Task 1: Document and sentence classification for protest event detection. In *Challenges and Applications of Automated Extraction of Socio-political Events from Text*, pages 171–178.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero- and few-shot relation extraction. *CoRR*, abs/2109.03659.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv*, 1910.01108.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically Identifying Words That Can Serve as Labels for Few-Shot Text Classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578.

Timo Schick and Hinrich Schütze. 2021. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*.

Philip Schrodt. 2012. CAMEO: Conflict and mediation event observations event and actor codebook. *Parus Analytics*.

Richard Shin, Christopher Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained Language Models Yield Few-Shot Semantic Parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Erick Skorupa Parolin, MohammadSaleh Hosseini, Yibo Hu, Latifur Khan, Patrick T. Brandt, Javier Osorio, and Vito D'Orazio. 2022. Multi-CoPED: A Multilingual Multi-Task Approach for Coding Political Event Data on Conflict and Mediation Domain. In *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, pages 700–711, Oxford United Kingdom. ACM.

Niklas Stoehr, Lucas Torroba Hennigen, Samin Ahbab, Robert West, and Ryan Cotterell. 2021. Classifying dyads for militarized conflict analysis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Niklas Stoehr, Lucas Torroba Hennigen, Josef Valvoda, Robert West, Ryan Cotterell, and Aaron Schein. 2022. An ordinal latent variable model of conflict intensity. In *arXiv*, volume 2210.03971.

Niklas Stoehr, Benjamin J. Radford, Ryan Cotterell, and Aaron Schein. 2023. The Ordered Matrix Dirichlet for state-space models. In *AISTATS*.

Umair ul Hassan, Sean O'Riain, and Edward Curry. 2013. SLUA: Towards Semantic Linking of Users with Actions in Crowdsourcing. In *CEUR Workshop Proceedings*, volume 1030.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as Few-Shot Learner. *CoRR*, abs/2104.14690. ArXiv: 2104.14690.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment 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 3912–3921, Hong Kong, China. Association for Computational Linguistics.

Mian Zhong, Shehzaad Dhuliawala, and Niklas Stoehr. 2023. Extracting victim counts from text. In *European Chapter of the ACL (EACL)*.

A Reproducibility Criteria

A.1 Experimental Results

1. A clear description of the mathematical setting, algorithm, and/or model
   - See Section §3
2. Submission of a zip file containing source code, with specification of all dependencies, including external libraries, or a link to such resources (while still anonymized)
   - Provided in the submission
3. Description of computing infrastructure used
   - PR-ENT inference: Dell Latitude 7490 laptop - Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz / 16 GB RAM
   - DistilBERT finetuning: Macbook Pro M1 Max - M1 Max / 32 GB RAM
   - Dashboard: 8 CPU Cores / 16 GB RAM
4. The average runtime for each model or algorithm (e.g., training, inference, etc.), or estimated energy cost
   - Training:
     - No training done for PR-ENT
     - For comparison purposes, a DistilBERT model was fine-tuned on 3000 samples. It took several minutes on a laptop.
   - Inference:
     - PR-ENT: 1-10secs per text depending on text length on a laptop
5. Number of parameters in each model:
   - DistilBERT-base-uncased (https://huggingface.co/distilbert-base-uncased): 65M
   - RoBERTa-large-mnli (https://huggingface.co/roberta-large-mnli): 125M
   - RoBERTA-large-squad2 (https://huggingface.co/deepset/roberta-large-squad2): 125M
   - PR-ENT: Top K, Entailment Threshold
6. Corresponding validation performance for each reported test result
   - Not applicable
7. Explanation of evaluation metrics used, with links to code
   - F1 Score, Scikit-learn
   - Precision, Scikit-learn
   - Recall, Scikit-learn
   - Accuracy, Scikit-learn
   - Jensen Shannon Distance, Scipy

A.2 Hyperparameter Search

Not applicable

A.3 Datasets

1. Relevant details such as languages, and number of examples and label distributions
   - ACLED: See section §2
   - GTD: See section §2
2. Details of train/validation/test splits
   - ACLED: 3000 train sample / 1000 test sample
   - GTD: 100 train sample / 1000 test sample
3. Explanation of any data that were excluded, and all pre-processing steps
   - See section §2
4. A zip file containing data or link to a downloadable version of the data
   - ACLED: Data is not open source. We provide a json file containing the event ID used in train and test set.
   - GTD: Data is available on GTD Website : We provide a json file containing the event ID used in train and test set
   - Provided in the submission
5. For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.
   - Not applicable

B Additional Material

B.1 Actor and Target Coding.

Until now, we studied how to code event types, which can be seen as actions or predicates of an event. We propose an extension to extract the actor and target of an event using question answering models similar to Halterman et al. (2021). In He et al. (2015), questions are constructed around a known action performed in an event. Given the example "Military injured two civilians.", PR-ENT yields "injured" as an action. Using this action, we can construct the questions "Who was injured?" and "Who injured people?" which are then fed to a QA model RoBERTa-base-squad2 (Rajpurkar et al., 2016). We present examples of extracted "who-did-what-to-whom" patterns in Tab. 6. Actor-target coding is even harder to evaluate, as there can be multiple actions / targets / actors in an event description and the abstract mapping between manually annotated entity types (e.g. civilians) and verbatim mentions (e.g. demonstrators) is not known.

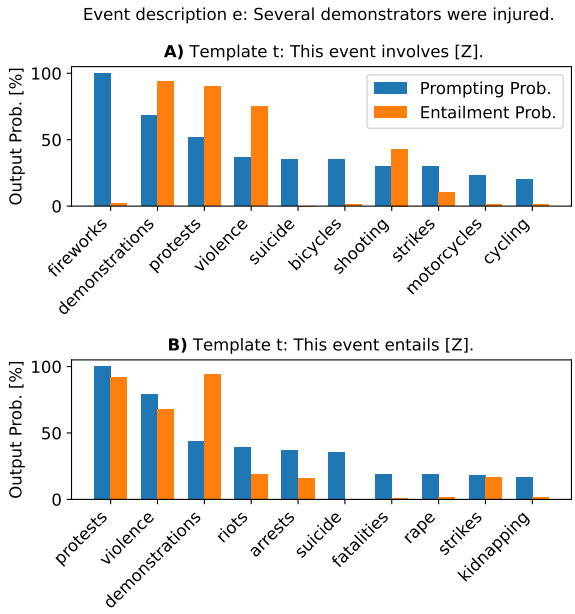

Figure 7: Given the event description *"Several demonstrators were injured."*, and the two similar templates **(A)** and **(B)**, we get drastically different answer candidates as shown by the top 10 outputs of the prompt model (blue bar). However, in both cases we obtain the same 3 answer candidates if they are filtered through an additional entailment step (orange bar).

Event Description + Extracted Actor-Target	Action
Arrests: **[WHO (31%): [LOC] police]** captured **[WHOM (90%): [NAME]]**, a senior [ORG] in [LOC]	arrested
On 3 January 2020, **[WHO (17%): [LOC] Armed Forces]** regained [LOC], [LOC], [LOC], [LOC] and [LOC] from [ORG]. In the operations 6 [ORG] fighters were arrested and **[WHOM (67%): 461 kidnapped civilians]** were rescued.	rescued
On 12 March 2020, **[WHO (40%): police and military intelligence officers]** raided the home of retired **[WHOM (15%, 6%): Lt. Gen [NAME]]**. The candidate was arrested and charged with treason in relation to remarks he made during a [WHO (29%): TV] interview; his staff of 18, as well as the MP for [ORG] as well as his son have all been arrested.	**arrested**; interviewed

Table 6: Actor-target coding based on our pipeline augmented with an additional extractive question-answering (QA) model. The first example represents a clear "who-did-what-to-whom" pattern. In the second example, actor and target are separated into two sentences. Finally, the third example shows an event with two *ARG0-V-ARG1* patterns (bolded and underlined). The confidence of the QA model is displayed for each answer.

Event Type	Template	Entailed Answer Candidate
Arrest	People were [Z].	arrested `AND NOT` kidnapped
Killing	This event involves [Z].	killing
	People were [Z].	killed
Looting	This event involves [Z].	looting `OR` theft `OR` robbery
Sexual Violence	This event involves [Z].	rape
	People were [Z].	abused `OR` raped
Kidnapping	This event involves [Z].	kidnapping
	People were [Z].	kidnapped `OR` abducted
Protest	This event involves [Z].	protest `OR` demonstration
	People were [Z].	protesting

Table 7: Example of an event ontology designed by means of our approach of entailment-based prompt selection PR-ENT. The final ontology is defined in terms of templates and expected entailed answer candidates. We use the event type "Killing" versus all others to classify "lethal" versus "non-lethal" events in Tab. 3. It's also used to compute results of Fig. 6 and Fig. 10.

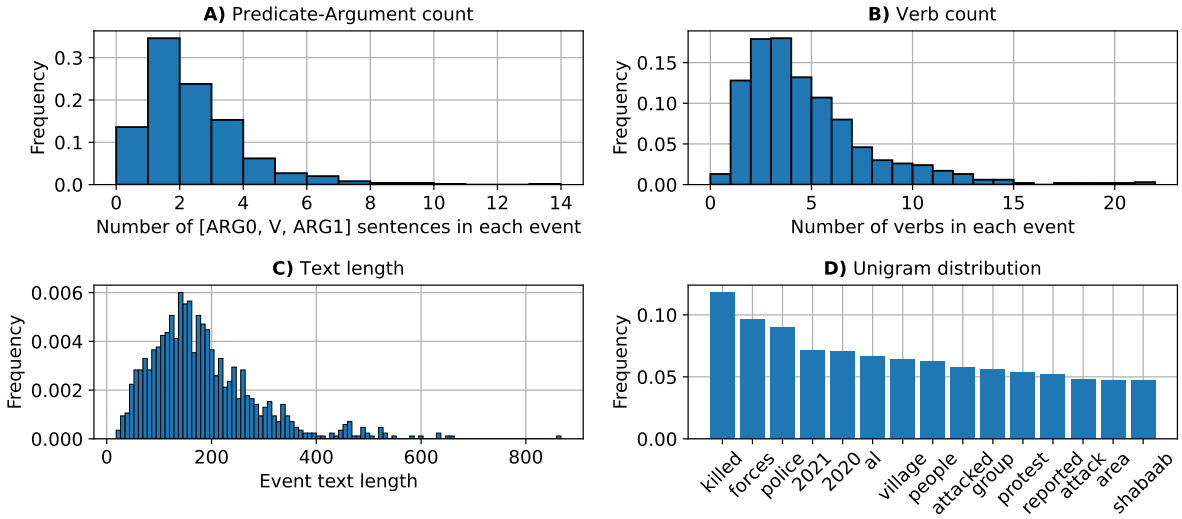

Figure 8: Statistics over a sample of 1000 ACLED event descriptions; **(A)** encountering many predicate-argument structures per event description can be an indication of difficult event coding; **(B)** number of verbs (actions) per event description; **(C)** length distribution of event descriptions; **(D)** unigram distribution over dataset.

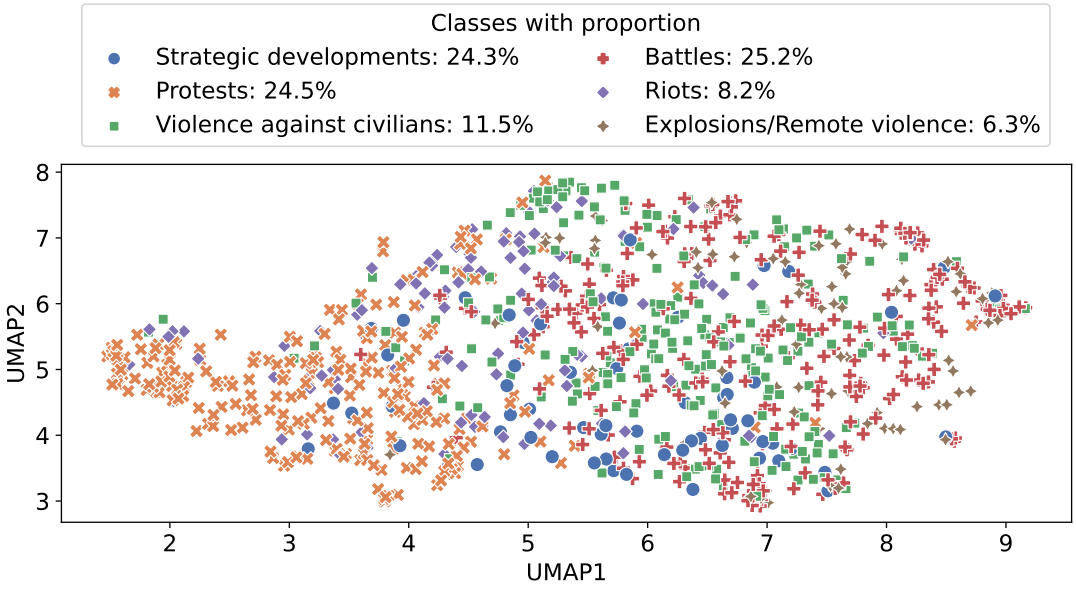

Figure 9: Event type distribution as visualized using UMAP over GloVe embeddings of the event descriptions. While some event types are easily distinguishable from each other (e.g. *Protests* and *Battles*), others are harder to tell apart (e.g. *Protests* and *Riots*). We also show the proportion of each event type in the legend.

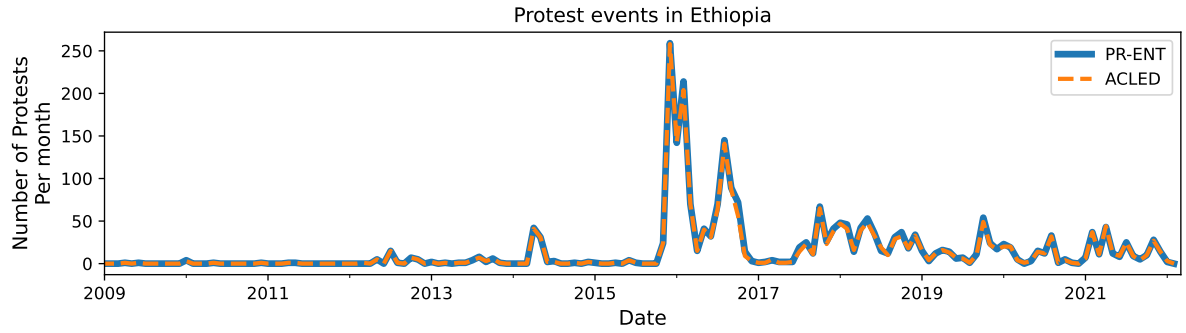

Figure 10: Time series of the number of protest events per month in Ethiopia between 2009-2021. The dashed line corresponds to all protest events coded by ACLED annotators. The blue line corresponds to all protest events coded by PR-ENT. Despite PR-ENT codings being machine-automated, they are very similar to ACLED's codings. Both clearly detect the high intensity violence periods in 2016.

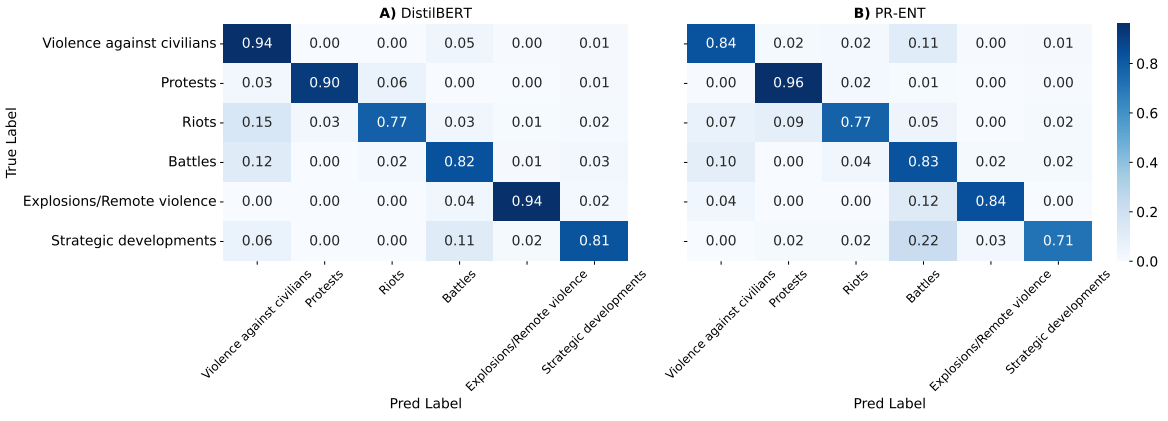

Figure 11: Confusion matrices of DistilBERT and PR-ENT + LR on the test set.