IBM MNLP IE at CASE 2021 Task 2: NLI Reranking for Zero-Shot Text Classification

Ken Barker, Parul Awasthy, Jian Ni, and Radu Florian
IBM Research AI
Yorktown Heights, NY 10598
{kjbarker, awasthyp, nij, raduf}@us.ibm.com

Abstract

Supervised models can achieve very high accuracy for fine-grained text classification. In practice, however, training data may be abundant for some types but scarce or even non-existent for others. We propose a hybrid architecture that uses as much labeled data as available for fine-tuning classification models, while also allowing for types with little (few-shot) or no (zero-shot) labeled data. In particular, we pair a supervised text classification model with a Natural Language Inference (NLI) reranking model. The NLI reranker uses a textual representation of target types that allows it to score the strength with which a type is implied by a text, without requiring training data for the types. Experiments show that the NLI model is very sensitive to the choice of textual representation, but can be effective for classifying unseen types. It can also improve classification accuracy for the known types of an already highly accurate supervised model.[1]

1 Task 2: Fine-grained Classification of Socio-political Events

Fine-grained text classification assigns a type label to a text passage from an extended set of specific types. The types are often domain-specific and more focused than generic, coarse-grained topics. Creating an exhaustive list of such types for a domain or task *a priori* is challenging. Fine-grained type systems for a particular domain and task often evolve, with new, previously unseen types emerging. So some types may have many labeled examples available, some types may have few or none. In such a scenario, a flexible text classifier should be able to use training data when available, but also employ few-shot and zero-shot techniques when training data is limited or absent.

The Fine-grained Classification of Socio-political Events task (Haneczok et al., 2021) at the CASE 2021 workshop (Hürriyetoğlu et al., 2021) simulates the scenario of text classification with an evolving, fine-grained type system. There are 25 core, fine-grained event types capturing political violence, demonstrations, and other politically important events. The 25 types are from the ACLED (Armed Conflict Location & Event Data Project) event taxonomy (Raleigh et al., 2010). Copious training data exists for these types. Subtask 1 evaluates classification to these "seen" or "known" types only. Beyond the core types, subtask 2 identifies three new "unseen" types with no training data. Definitions of these three types were provided to task participants early, allowing exploration of zero-shot techniques on the types, but also few-shot techniques since the development window would allow enough time to annotate a handful of examples. Finally, subtask 3 introduces two additional unseen types revealed only at evaluation time. Subtask 3 evaluates true zero-shot techniques. Table 1 lists the type labels and names for the three subtasks.

For the task evaluation, organizers shared an evaluation set of 1,019 short texts (mostly one or two sentences each) with index numbers. 829 of the texts had gold labels from the 25 ACLED event types; 118 had gold labels from the three subtask2 types; 72 had gold labels from the two unseen subtask3 types. These numbers were not known at submission time. Submissions consisted of pairs of an index number with a single event type prediction for the text associated with the index. For scoring, the organizers removed entries whose gold event type was not among those being tested for the given subtask, and computed micro, macro, and weighted Precision, Recall, and F1-score. Weighted scores are the average of the per-type scores (like macro averages), but with the type scores weighted by the number of gold instances for the type.

[1]Distribution Statement "A" (Approved for Public Release, Distribution Unlimited)

Type Label	Type Name
subtask 1	
ABDUCT_DISSAP	abduction
AGREEMENT	agreement
AIR_STRIKE	air strike
ARMED_CLASH	armed clash
ARREST	arrest
ART_MISS_ATTACK	artillery, missile attack
ATTACK	attack
CHANGE_TO_GRO...	change to group activity
CHEM_WEAP	chemical weapon
DISR_WEAP	disrupted weapons use
FORCE_AGAINST...	excessive force against protesters
GOV_REGAINS_TER...	government regains territory
GRENADE	grenade
HQ_ESTABLISHED	headquarters established
MOB_VIOL	mob violence
NON_STATE_ACT...	non-state actor overtakes territory
NON_VIOL_TERR...	non-violent transfer of territory
PEACE_PROTEST	peaceful protest
PROPERTY_DISTR...	property destruction
PROTEST_WITH_I...	protest with intervention
REM_EXPLOS	remote explosive
SEX_VIOL	sexual violence
SUIC_BOMB	suicide bomb
VIOL_DEMONSTR	violent demonstration
subtask 2	
ORG_CRIME	organized crime
NATURAL_DISAST...	natural disaster
MAN_MADE_DISAS...	man-made disaster
subtask 3	
ATTRIB	attribution of responsibility
DIPLO	diplomatic event

Table 1: Type labels (some truncated) and names for the 25 known types of subtask 1, the three unseen types of subtask 2, and the two unseen types of subtask 3

Our approach to fine-grained text classification mirrors the evolving type system scenario: a hybrid system that is fine-tuned with labeled data when available, but one that can also classify text with types having little or no labeled data. Our approach is first to apply a supervised text classification model to produce a ranked list of predicted, known types. The highest scoring types from the classification model are combined with any unseen types and passed to a Natural Language Inference (NLI) reranking model. The NLI reranker rescores the types on the extent to which they are implied by the input text.

2 System Architecture

We experimented with many different combinations of supervised, few-shot, and zero-shot techniques and submitted multiple such combinations for each of the Case Task 2 subtasks. Despite their differences, all submissions are built on the same cascaded architecture of a supervised neural classification model followed by a neural NLI-based reranking model. The submissions differ on the exact combination of classification model and reranking model.

For each sentence, the classification model produces a ranked list of predicted types with scores, but only for the types the model was trained on. If the score of the top-ranked predicted type is below threshold, or if the top-ranked predicted type is OTHER, the top N predicted types P_N are passed to the reranking model along with all unseen types U. The reranker independently scores each of the types in $P_N \cup U$. The highest scoring type is the final prediction for the sentence.

For each of the known and unseen types $P_N \cup U$ to be submitted to the reranker, we generate a textual representation, based only on the definition of the type (not labeled examples). See section 4.2.1 for details on how we generate textual representations. The NLI reranking model scores each of these representations on the extent to which they are implied by the input text. Figure 1 illustrates the general architecture.

3 Supervised Sequence Classification for Seen Types

For the text classifier we used a standard transformer-based sequence classification model (Vaswani et al., 2017) with a pre-trained language model. Based on previous experience with such text classification systems, we chose RoBERTa (Liu et al., 2019). Specifically, we started with the `roberta-large` model from Hugging Face (Wolf et al., 2020).

3.1 Data

For the 25 known ACLED event types, we used the `ACLED-C-III` dataset derived from the ACLED source data by (Piskorski et al., 2020). This dataset contains 588,940 short text passages each labeled with exactly one of the 25 ACLED event types. The dataset is organized in four folds, where each fold is a different random split of the 588,940 instances into 80% training and 20% test sets. For our 25-type base classifier we fine-tuned `roberta-large` on the training subset (471,152 instances) of fold 1 of the Piskorski dataset. For development experiments to arrive at our final architectures and parameters, we used subsets of the fold 1 test subset (117,788 instances). Piskorski also provides smaller partitions of the dataset used in their learning curve experiments. In our smaller

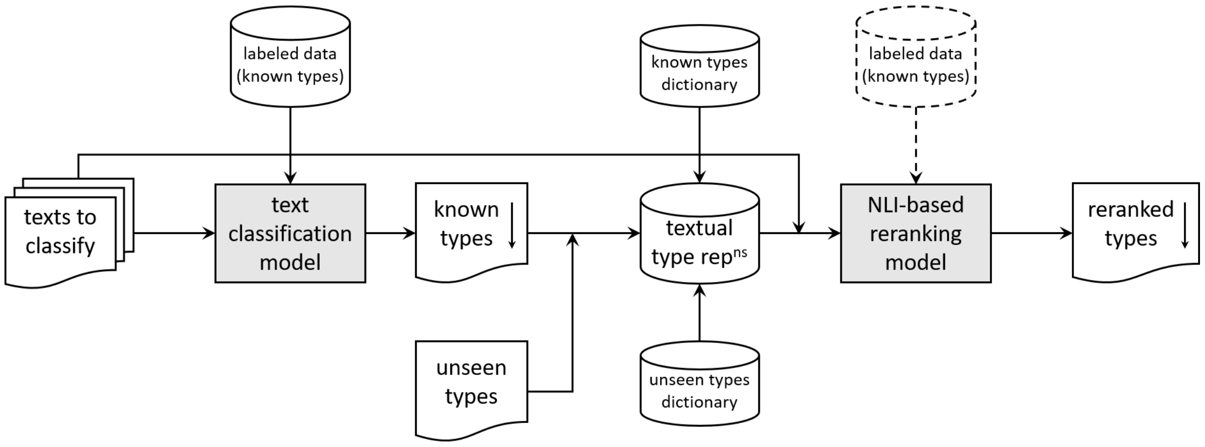

Figure 1: Cascaded Text Classification + NLI Reranking architecture. The classification model is trained on known types only and predicts a ranked list of known types. Any unseen types are added to the top N predicted known types for reranking. The reranking model may be fine-tuned for known types, in which case it is supervised (or few-shot supervised) for those types. The architecture is zero-shot for unseen types.

models (including `roberta-FT28`, see section 3.2), we used the 10% partition of the fold 1 training data.

In addition to the base classifier, we experimented with few-shot supervision for classifying the three unseen event types of subtask 2. We manually created a small training corpus of short texts (one or two sentences each) from Wikipedia and Wikinews. We found relevant documents by browsing the Wikipedia/Wikinews Categories hierarchies (for example, `https://en.wikipedia.org/wiki/Category:Natural_disasters` and `https://en.wikinews.org/wiki/Category:Crime_and_law`). Within relevant documents, we chose short text passages that described a single event of the target type. These passages were often simply the first one or two sentences in the document. An example appears in Figure 2. We originally collected 142 texts total: 56 texts for MAN_MADE_DISASTER; 55 texts for NATURAL_DISASTER; 31 texts for ORG_CRIME. For tuning experiments we created an additional test corpus of 20 short texts each for the three types. For the classification model used for final submissions on the challenge evaluation data, we combined both small corpora (along with an additional 20 texts for ORG_CRIME) for training on a total of 222 texts balanced among the three types.

3.2 Models

We fine-tuned `roberta-large` for sequence classification (with a linear classification layer) on the data described in section 3.1 to produce two classification models:

- `roberta-FT25`: fine-tuned on the 471,152 instances of ACLED training data for the 25 base ACLED event types
- `roberta-FT28`: fine-tuned on the 10% partition of the ACLED training data (47,115 instances) plus additional instances for the three unseen types of subtask 2 (142 instances during development, 222 for the final models)

Given the disparity in the number of training instances, we consider `roberta-FT28` to be supervised for the 25 types and *few-shot* supervised for the three unseen types.

Clock time for training `roberta-FT25` was 30 hours on four v100 GPUs. The time to train `roberta-FT28` was 4.6 hours on four v100 GPUs.

Experiments in (Piskorski et al., 2020) also trained models on the `ACLED-C-III` data, with

Search and rescue workers in Arkansas continue to search the Little Missouri and Caddo Rivers for survivors of Friday's flash flood. At least nineteen people were killed when the flood swept through the Albert Pike Recreation Area campground in the Ouachita National Forest in the southwestern portion of the state.

Figure 2: A short NATURAL_DISASTER text consisting of the first two sentences of a Wikinews article under the Category `https://en.wikinews.org/wiki/Category:Disasters_and_accidents`.

model	trdata	μF	macF	wF
Piskorski	100%	94.3	86.0	94.2
roberta-FT25	100%	94.7	87.3	94.7
Piskorski	10%	>90	>70	>90
roberta-FT28	10%	93.5	77.1	93.5

Table 2: Comparison of micro F_1 score, macro F_1 score, and weighted F_1 score on test fold 1 of the `ACLED-C-III` dataset. Piskorkski refers to the `bert-base` model fine-tuned by Piskorski et al. (2020), which was the top performer on test fold 1 in their experiments. Test scores for the Piskorski model trained on 10% of the training data are estimated from the graphs in Piskorski et al. (2020).

metrics reported on test fold 1. So those results can be compared directly with our models. Performance of our model is consistent with their results fine-tuning `bert-base` (see Table 2).

4 NLI for Unseen Types

Pretrained language models (PLMs) have proven to be very powerful for downstream NLP tasks when they are fine-tuned on data specific to the task. Recently, the research community has begun to observe that PLMs fine-tuned on large amounts of data for complex end-to-end tasks can often be leveraged for new tasks without further fine-tuning. Fine-tuning PLMs on complex tasks such as Question Answering (QA) and Natural Language Inference (NLI) infuses models with high-level knowledge useful for other tasks. By choosing an appropriate task representation, QA models and NLI models can be used as “pre-tuned” models for few-shot (Schick and Schütze, 2021) or even zero-shot (Yin et al., 2019) text classification.

Typically, an NLI model takes two texts (sentence1 and sentence2) and predicts whether sentence1 implies sentence2, with a given confidence score. To re-purpose an NLI model for zero-shot text classification, sentence1 is the text to be classified and sentence2 is some textual representation of a type. The classification score for each type is the NLI score, which represents the extent to which the textual representation of the type is implied by sentence1. Determining implication is not just based on surface lexical overlap between the sentences. In training, the models learn encodings for both sentences, supervised by a large corpus of hand-labeled textual entailment pairs (such as the 433k sentence pairs in the multi-genre RepEval corpus (Williams et al., 2018)).

For the current work, we explored using large, pre-tuned NLI models for few-shot and zero-shot classification. We experimented with NLI extensions to both BART (Lewis et al., 2020) and RoBERTa (Liu et al., 2019) Language Models. For both of these, large models fine-tuned for the NLI task are available from Hugging Face: `bart-large-mnli` and `roberta-large-mnli`. Our experiments tended to favor RoBERTa (see section 4.3.3).

We also experimented with further fine-tuning the NLI models for the 25 subtask 1 types and the three subtask 2 types (sections 4.3.1 and 4.3.2).

4.1 Type Representations

A crucial design choice when using NLI for zero-shot text classification is the choice of representation of the types. We experimented with full English descriptions, keywords, and type names. Examples of each representation for a sample sentence appear in Table 3.

The full English descriptions are the type definitions taken verbatim from source documentation. For the original 25 types, these were taken from ACLED directly. for the five unseen types, definitions were provided by the organizers. The type names were taken from the same source documentation. The keywords were extracted manually from the definitions, with no editing other than deleting text from the definition. The disjoint keywords are exactly the same as the keywords, the difference being how they are submitted to the NLI model.

Table 4 shows how critical the choice of type representation is. The superiority of the name representation over definition and keywords was initially surprising, since it contains so much less information. We hypothesized that having multiple distinct terms covering more of the breadth of a type could help the NLI model, but that specific terms irrelevant to a given text were more harmful in confusing the model than the underspecificity of a single generic phrase. For example, the presence of terms such as “pandemic”, “volcano”, and “wildfire” would be distracting to a model when trying to determine whether a sentence about avalanches implies a natural disaster. To test the hypothesis, we considered a fourth type representation: *disjunctive keywords*. Rather than a single type representation in which all keywords appear together, with disjunctive keywords, each keyword for a type is considered an independent representation of the

sentence1	*A hacker group called DarkSide is behind the cyberattack on Colonial Pipeline that shut down a major oil pipeline over the weekend.*
Type Rep'n	**sentence2**
definition	*Organized crime: This event type covers incidents related to activities of criminal groups, excluding conflict between such groups: smuggling, human trafficking, counterfeit products, property crime, cyber crime, assassination (for criminal purposes), corruption, etc. (list is non-exhaustive).*
name	*organized crime*
keywords	*organized crime, smuggling, human trafficking, counterfeit products, property crime, cyber crime, assassination, corruption*
disj-kw	*organized crime | smuggling | human trafficking | counterfeit products | property crime | cyber crime | assassination | corruption*

Table 3: Example type representations for a sentence of the type ORG_CRIME. For the disj-kw representation, each phrase in the pipe-delimited list is given separately to the NLI model as sentence2.

Type rep'n	**μAcc**	**macAcc**
definition	0.5461	0.5672
keywords	0.4610	0.5243
name	0.8723	0.8456
disj-kw	0.8936	0.9048

Table 4: Effect of different type representations on zero-shot accuracy for the 142-example dev set for the three unseen types.

type. The extent to which a text implies a type t is the maximum score produced by the NLI model for any of t's keywords $kw(t)_i$.

4.2 Data

The original 25 ACLED event types include the type OTHER, which indicates a closed-world assumption: any event that is not one of the 24 main event types is of type OTHER. In the evolving type system scenario, the closed-world assumption does not hold. Texts labeled OTHER in the classifier training data may describe events of the unseen types. For this reason, we trained our classifiers (section 3) on all known types (including OTHER), but remove OTHER from the top N types submitted to the reranker. We also exclude OTHER instances from any training data used to fine-tune the reranking models.

To compare zero-shot reranking and reranking fine-tuned to known types, we prepared two new datasets.

For the 24 ACLED event types (ignoring OTHER), we created a training dataset derived from a subset of 10% of Piskorki's fold 1 training set (section 3.1). This initial data gave 46,886 positive instances. We added two negative examples for each positive example giving 140,658 total instances. The process for generating both positive and negative examples is described below in section 4.2.1.

To fine tune a few-shot supervised NLI model for the three unseen types of subtask 2, we created a dataset derived from the small dataset described in section 3.1. The 222 instances from that dataset provided the positive examples, to which we again added two negatives each, giving 666 total instances.

4.2.1 Labeling NLI Data

The labeled data for classification (section 3.1) consists of sentences with gold event type labels (REM_EXPLOS, ORG_CRIME, etc.). We need to adapt the data in two ways to use it for fine-tuning NLI models.

First, the labels must be replaced by a textual representation of the class, which will be used as sentence2 by the NLI model. We chose a disjunctive keyword representation as described in section 4.1. To create positive examples, we paired each sentence with each of the disjunctive keywords of the gold label. We then used the untuned `roberta-large-mnli` model to score each of these pairs to determine which of the disjunctive keywords was most strongly implied by the sentence. This gave us the best single keyword to use as sentence 2 for each positive training sentence.

Second, fine-tuning the model requires negative examples. For each positive example, we created two negative examples by replacing the gold type

positive	**s1**	*A hacker group called DarkSide is behind the cyberattack on Colonial Pipeline that shut down a major oil pipeline over the weekend.*
	s2	*cyber crime*
random negative	**s1**	*A hacker group called DarkSide is behind the cyberattack on Colonial Pipeline that shut down a major oil pipeline over the weekend.*
	s2	*forced disappearance*
imposter negative	**s1**	*A hacker group called DarkSide is behind the cyberattack on Colonial Pipeline that shut down a major oil pipeline over the weekend.*
	s2	*attack*

Table 5: Example positive and negative instances generated from a sentence whose gold event type label is ORG_CRIME. Sentence 2 (s2) for the positive instance is the keyword from the gold event type keywords scored highest by `roberta-large-mnli`. Sentence 2 for the imposter instance is the highest-scoring keyword from the non-gold event types.

representation with the type representation of a different event type: one random and one imposter. The random negative example is the original sentence paired with a random keyword selected from all of the keywords of types other than the gold type. The imposter is the keyword most likely to confuse the model. We paired each sentence with each of the disjunctive keywords for all of the types other than the gold type. We used the top scoring pair (the most strongly implied incorrect type representation) as the imposter example.

Table 5 shows positive and negative training instances created for an example sentence.

4.3 Development Experiments and Results

4.3.1 Classification + Reranking

Using an NLI model as described in section 4 is needed for unseen types, since the classification model cannot predict types it was not trained on. But NLI reranking might also improve predictions on the known types. To explore this possibility, we fine-tuned the `bart-large-mnli` model for the 24 non-OTHER ACLED event types using the 140,658 NLI training pairs from the dataset described in section 4.2. We then ran our fine-tuned RoBERTa classifier (`roberta-FT25`) on the 117,788 instances of the ACLED development set and collected its ranked list of predictions for all of the instances where its top-scoring prediction was wrong (6,002 instances, accuracy = 0.0000).

For the 6,002 instances, the average classifier score for the top (wrong) prediction was 0.9644. The average position of the correct type within the list ranked by classifier score was 3.1. When the classifier predicted the correct type, the average classifier top score was 0.9960. When the classifier score was above 0.99, its accuracy was 0.9562. Below that threshold, accuracy was 0.4851.

To explore the benefit of reranking even for known types, we passed the top N classifier predictions for the 6,002 incorrectly classified instances to our fine-tuned BART NLI model for reranking. We first set N to the position of the correct type for each sentence, guaranteeing that the correct type was in the list to be reranked (we refer to this as *setting N to "gold"*). The fine-tuned NLI model predicted the correct type for 2,271 instances (37.8% of the incorrectly classified instances). Setting N to 5 (meaning for some sentences the correct type would not be among the top N), the fine-tuned NLI model predicted 2,027 instances correctly (33.7%). See Table 6.

Surprisingly, using `bart-large-mnli` without fine tuning performed even better on the incorrectly classified instances when the correct type is always in the top N, recovering 2,600 of the correct types (43.3%). When N was 5, however, the untuned model did not perform as well as the tuned model (29.6% recovered). As with other experiments, the untuned `roberta-large-mnli` model performs slightly better than the BART model.

Based on this experiment, it makes sense to pair a fine-tuned classifier with an NLI reranker (fine-tuned or not) even for known types. In practice, we invoke the reranker with the top 5 predictions from the classifier when the classifier's top prediction score is less than 0.99.

4.3.2 Zero-Shot vs. Fine-Tuned Reranking on Unseen Types

We also compared a fine-tuned NLI model to an untuned model on unseen types. To simu-

Model	N	Accuracy
`bart-FT24`	gold	0.3784
`bart-FT24`	5	0.3377
`bart-large-mnli`	gold	0.4332
`bart-large-mnli`	5	0.2956
`roberta-large-mnli`	gold	0.4347
`roberta-large-mnli`	5	0.3121

Table 6: Accuracy of NLI models on the top N predictions from the `roberta-FT25` classifier when its top prediction is wrong (6,002 instances). N=gold means that N is the position of the correct type in the ranked list, guaranteeing that it is available to the reranker.

Model	Tune	μAcc	macAcc
`bart-large-mnli`	none	0.9104	0.7977
`bart-mnli-FT24`	24	0.9806	0.9708
`bart-mnli-FT21`	21	0.2396	0.2540

Table 7: Comparison of three NLI models tested on data for three held-out types: an untuned model, a model tuned on all types (including the held-out types), and a model tuned on all types *except* the held-out types.

late the unseen types scenario, we randomly selected three of the original ACLED types and removed their instances (6,080) from the NLI fine-tuning dataset. We then fine-tuned the `bart-large-mnli` model on the instances for the remaining 21 types. Table 7 shows that fine-tuning an NLI model significantly improves accuracy for types it was tuned on over an untuned NLI model. Fine-tuning an NLI model on known types and then applying to unseen types performs catastrophically worse than even the untuned model. The fine-tuned model is clearly overfitting the 21 types in a way that significantly degrades performance on unseen types. We tried reducing the size of the fine-tuning set and balancing it to 1,000 instances per type to avoid overfitting. The resulting model performed worse by 5-6 micro-averaged accuracy points in all experiments.

Based on this experiment, we conclude that using a fine-tuned NLI model improves reranking for the types on which it was fine-tuned, but for unseen types, it is preferable to use an NLI reranking model not fine-tuned on other types.

4.3.3 BART vs. RoBERTa

We conducted two additional experiments to compare `bart-large-mnli` and `roberta-large-mnli` (no fine tuning), using keywords as the textual representation of the types. The first dataset was 457 examples from the ACLED test set, converted to the format needed for the NLI models (section 4.2.1). The 457 examples cover all 24 non-OTHER ACLED types, making this a particularly challenging task for reranking. The second dataset was the 6,080 instances covering the three held out types described in the previous section (4.3.2). Both experiments show a preference for RoBERTa. The experiments also show that the models are much better at distinguishing among smaller numbers of types. This supports the approach of using NLI models as rerankers on the top N classifier predictions (for small N) instead of using them as classifiers themselves on the full inventory of types.

Model	457-all	3heldout
`bart-large-mnli`	0.2998	0.9104
`roberta-large-mnli`	0.3129	0.9533

Table 8: Accuracy of BART vs. RoBERTa NLI models with no fine-tuning on two ACLED datasets: 457 random instances covering all event types and 6,080 instances covering three event types.

4.4 Models

Based on the lessons learned from our development experiments, we ultimately fine-tuned two NLI models on the data described in section 4.2

- `roberta-mnli-FT24`: `roberta-large-mnli` fine-tuned on the 140,658 instance training set for the base ACLED event types
- `roberta-mnli-FT27`: `roberta-large-mnli` fine-tuned on the 140,658 ACLED instances plus 666 instances for the three unseen types of subtask 2

Using the `roberta-mnli-FT24` fine-tuned reranker in a system configuration makes that system supervised on the 24 ACLED types of subtask 1. Using `roberta-mnli-FT27` makes a system supervised on the ACLED types and few-shot supervised on the three unseen types of subtask 2.

5 Challenge Submissions and Results

Official task results appear in Table 10. The table shows how little difference there was between the top scoring team submissions. Full results and analysis are presented in Haneczok et al. (2021).

Sub#	Classif model	Rerank model	25 classif	25 rerank	3 classif	3 rerank	2 rerank
1.1	roberta-FT25	*none*	sup				
1.2	roberta-FT25	rob-large-mnli	sup	zero			
1.3	roberta-FT25	rob-mnli-FT24	sup	sup			
2.1	roberta-FT28	*none*	sup		few		
2.2	roberta-FT28	rob-large-mnli	sup	zero	few	zero	
2.3	roberta-FT28	rob-mnli-FT27	sup	sup	few	few	
2.4	sub 1.3 output	rob-large-mnli	sup	sup		zero	
3.1	roberta-FT25	rob-large-mnli	sup	zero		zero	zero
3.2	roberta-FT28	rob-large-mnli	sup	zero	few	zero	zero
3.3	sub 1.3 output	rob-large-mnli	sup	sup		zero	zero
3.4	sub 2.2 output	rob-large-mnli	sup	zero	few	zero	zero

Table 9: System configurations for each of the eleven submissions. The combination of classification model and reranking model determines whether classification and reranking are supervised (*sup*), few-shot (*few*), or zero-shot (*zero*) for each category of event types (the 25 seen types, the 3 unseen types of subtask 2, or the 2 unseen types of subtask 3).

task	Our Score	Best Other Score	Our Rank
1	**0.839**	0.832	1
2	**0.797 (0.785)**	0.782	1
3	0.756 (0.746)	**0.771**	2

Table 10: Official challenge results (weighted F_1 scores). Top score is boldface. For subtasks 2 and 3, our highest scoring submission was few-shot fine-tuned for the subtask 2 event types and zero-shot for the subtask 3 event types. The score for our best true zero-shot submission appears in parentheses. Best Other Score is the highest scoring submission from another team.

We now turn to a more detailed discussion of our submissions and more detailed scores.

Combining classifiers and rerankers, we arrived at eleven system configurations for submission to the Challenge evaluation. Table 9 lists these configurations, grouped by Challenge subtask. The table specifies which classification model and which reranking model were used for each submission, as well as an indication of whether the configuration was supervised, few-shot, or zero-shot for each of the subsets of event types.

In every case, the classification model for the known 25 ACLED event types was supervised. For the first three submissions of subtask 2, classification of the three unseen types was few-shot, trained on our small corpus (see Section 3.1). For the fourth subtask 2 submission, we used the output of submission 1.3 as the "classifier". Since submission 1.3 was tuned on data for the 25 known types only, submission 2.4 is zero-shot for the three unseen types. No training data was used for the two new unseen types of subtask 3, so those types were always introduced during the reranking stage only (no classification). These were always zero-shot.

Reranking of the 25 original types was supervised when using the RoBERTa NLI model fine-tuned on the ACLED data (`rob-mnli-FT24` in the table. When using `roberta-large-mnli` off-the-shelf, reranking the 25 was zero-shot. For the 3 unseen types of subtask 2 (and subtask 3), only submission 2.3 reranked using the NLI model fine-tuned on the small amount of data for these types, and is considered few-shot. Otherwise, all reranking of the 3 unseen types and the 2 unseen types of subtask 3 were zero-shot.

6 Observations and Discussion

Table 11 shows results on the official, Task 2 evaluation. For subtask 1, fine-tuning the reranker did better (submission 1.3) than using an untuned reranker (1.2). This is the same result that we saw when passing the top 5 classification predictions to the reranker in the development experiments.

The best performing configuration for subtask 2 overall was supervised classification for all 28 types with untuned (zero-shot) reranking. In particular, the zero-shot reranking (submission 2.2) outperformed the reranker tuned on the three unseen types. This runs counter to what we saw in the development experiments. The configuration that was most successful on the original 25 types was the one whose classifier was the 1.3 submission (which had a fine-tuned reranker). Isolating

		All types			25 types		3 types		2 types	
Sub#	**Sup**	μ**F**	**macF**	**wF**	μ**F**	**macF**	μ**F**	**macF**	μ**F**	**macF**
1.1	sup	0.830	0.792	0.828	0.830	0.792				
1.2	sup	0.834	0.810	0.835	0.834	0.810				
1.3	sup	**0.838**	**0.814**	**0.839**	**0.838**	**0.814**				
2.1	few	0.782	0.752	0.779	0.790	0.785	0.712	0.703		
2.2	few	**0.797**	**0.773**	**0.797**	0.800	0.786	**0.779**	**0.756**		
2.3	few	0.794	0.764	0.790	0.799	0.785	0.749	0.746		
2.4	zero	0.786	0.758	0.785	**0.804**	**0.787**	0.621	0.595		
3.1	zero	0.744	0.720	0.746	**0.783**	**0.780**	0.591	0.572	0.362	0.379
3.2	few	**0.755**	**0.728**	**0.756**	0.776	0.765	**0.766**	**0.745**	**0.424**	**0.432**
3.3	zero	0.744	0.720	0.746	**0.783**	**0.780**	0.591	0.572	0.362	0.379
3.4	few	**0.755**	**0.728**	**0.756**	0.776	0.765	**0.766**	**0.745**	**0.424**	**0.432**

Table 11: Detailed scores for the eleven submissions. The "Sup" column denotes whether the submission overall should be considered supervised, few-shot, or zero-shot. μF is micro-averaged F_1 score; macF is macro-averaged F_1 score; wF is the weighted average F_1 score (see section 1). The highest scores for a given subtask and type subset are shown boldface.

the scores on the three unseen types versus the 25 known types shows strong performance in the few-shot case, but significantly weaker performance with zero-shot (2.4).

For subtask 3, submissions 3.1 and 3.3 produced identical predictions (not just scores), as did submissions 3.2 and 3.4. The configurations themselves are not equivalent, with the input to the 3.3 and 3.4 rerankers having already been reranked by the 1.3 and 2.2 rerankers. Interestingly, the configurations built on zero-shot rerankers only performed best, again suggesting NLI models can be used without fine-tuning for reranking classification for both known and unseen types. Performance on the two unseen types of subtask 3 (zero-shot) is significantly weaker than the zero-shot scenario of subtask 2 (2.4). It is possible that the two new types are inherently more difficult to recognize. But we suspect that tweaks to the textual representations for these two types might improve performance. Given the extreme differences that different representations produce (section 4.1), we expect that more carefully chosen representations would help.

7 Conclusion

The CASE 2021 Task 2 challenge accurately simulates a realistic, fine-grained, text classification scenario in which many types in the type inventory have abundant labeled data, some types are recently new and may have a small amount of labeled data, and some types are completely new and have no labeled data. Within these constraints, we proposed a hybrid system that combines supervised classification with NLI-based reranking that can be used in supervised, few-shot, and zero-shot settings. Our results show strong performance on known types with weaker results on unseen types. Nevertheless, the experiments for this challenge have produced some interesting conclusions. First, we confirm that NLI models are useful for zero-shot text classification, but only when distinguishing between a small number of target types. Second, even in a fully supervised scenario, where ample training data can produce classification models with extremely high accuracy, untuned NLI-based reranking can improve classification performance on known types. Third, the choice of textual representation to transform a classification problem into one amenable to an untuned NLI model greatly affects performance. In future work we hope to explore more rigorously what makes a good representation for NLI-based zero-shot text classification, and how to generate these representations more automatically.

Acknowledgments and Disclaimer

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-19-C-0206. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

Jacek Haneczok, Guillaume Jacquet, Jakub Piskorski, and Nicolas Stefanovitch. 2021. Fine-grained event classification in news-like text snippets - shared task 2, CASE 2021. In *Proceedings of the Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, online. Association for Computational Linguistics (ACL).

Ali Hürriyetoğlu, Hristo Tanev, Vanni Zavarella, Jakub Piskorski, Reyyan Yeniterzi, and Erdem Yörük. 2021. Challenges and applications of automated extraction of socio-political events from text (CASE 2021): Workshop and shared task report. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, online. Association for Computational Linguistics (ACL).

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

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

Jakub Piskorski, Jacek Haneczok, and Guillaume Jacquet. 2020. New benchmark corpus and models for fine-grained event classification: To BERT or not to BERT? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6663–6678, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Clionadh Raleigh, Andrew Linke, Håvard Hegre, and Joakim Karlsen. 2010. Introducing acled: an armed conflict location and event dataset: special data feature. *Journal of peace research*, 47(5):651–660.

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: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

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

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 3914–3923, Hong Kong, China. Association for Computational Linguistics.