IBM MNLP IE at CASE 2021 Task 1: Multigranular and Multilingual Event Detection on Protest News

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

Abstract

In this paper, we present the event detection models and systems we have developed for Multilingual Protest News Detection - Shared Task 1 at CASE 2021. [1] The shared task has 4 subtasks which cover event detection at different granularity levels (from document level to token level) and across multiple languages (English, Hindi, Portuguese and Spanish). To handle data from multiple languages, we use a multilingual transformer-based language model (XLM-R) as the input text encoder. We apply a variety of techniques and build several transformer-based models that perform consistently well across all the subtasks and languages. Our systems achieve an average F_1 score of 81.2. Out of thirteen subtask-language tracks, our submissions rank 1^{st} in nine and 2^{nd} in four tracks.

1 Introduction

Event detection aims to detect and extract useful information about certain types of events from text. It is an important information extraction task that discovers and gathers knowledge about past and ongoing events hidden in huge amounts of textual data.

The CASE 2021 workshop (Hürriyetoğlu et al., 2021b) focuses on socio-political and crisis event detection. The workshop defines 3 shared tasks. In this paper we describe our models and systems developed for "Multilingual Protest News Detection - Shared Task 1" (Hürriyetoğlu et al., 2021a). Shared task 1 in turn has 4 subtasks:

- *Subtask 1 - Document Classification*: determine whether a news article (document) contains information about a past or ongoing event.
- *Subtask 2 - Sentence Classification*: determine whether a sentence expresses information about a past or ongoing event.
- *Subtask 3 - Event Sentence Coreference Identification*: determine which event sentences refer to the same event.
- *Subtask 4 - Event Extraction*: extract event triggers and the associated arguments from event sentences.

Event extraction on news has long been popular, and benchmarks such as ACE (Walker et al., 2006) and ERE (Song et al., 2015) annotate event triggers, arguments and coreference. Most previous work has addressed these tasks separately. Hürriyetoğlu et al. (2020) also focused on detecting social-political events, but CASE 2021 has added more subtasks and languages.

CASE 2021 addresses event information extraction at different granularity levels, from the coarsest-grained document level to the finest-grained token level. The workshop enables participants to build models for these subtasks and compare similar methods across the subtasks.

The task is multilingual, making it even more challenging. In a globally-connected era, information about events is available in many different languages, so it is important to develop models that can operate across the language barriers. The common languages for all CASE Task 1 subtasks are English, Spanish, and Portuguese. Hindi is an additional language for subtask 1. Some of these languages are zero-shot (Hindi), or low resource (Portuguese and Spanish) for certain subtasks.

In this paper, we describe our multilingual transformer-based models and systems for each of the subtasks. We describe the data for the subtasks in section 2. We use XLM-R (Conneau et al.,

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

Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) , pages 138–146
August 5–6, 2021, ©2021 Association for Computational Linguistics

Task	Language	Train	Dev	Test
1	English (en)	8392	932	2971
	Spanish (es)	800	200	250
	Portuguese (pt)	1190	297	372
	Hindi (hi)	-	-	268
2	English (en)	20543	2282	1290
	Spanish (es)	2193	548	686
	Portuguese (pt)	946	236	1445
3	English (en)	476	120	100
	Spanish (es)	-	11	40
	Portuguese (pt)	-	21	40
4	English (en)	2565	681	311
	Spanish (es)	106	-	190
	Portuguese (pt)	87	-	192

Table 1: Number of examples in the train/dev/test sets. Subtasks 1 and 3 counts show number of documents, and subtasks 2 and 4 counts show number of sentences.

2020) as the input text encoder, described in section 3. For subtasks 1 (document classification) and 2 (sentence classification), we apply multilingual and monolingual text classifiers with different window sizes (Sections 4 and 5). For subtask 3 (event sentence coreference identification), we use a system with two modules: a classification module followed by a clustering module (section 6). For subtask 4 (event extraction), we apply a sequence labeling approach and build both multilingual and monolingual models (section 7). We present the final evaluation results in section 8. Our models have achieved consistently high performance scores across all the subtasks and languages.

2 Data

The data for this task has been created using the method described in Hürriyetoğlu et al. (2021). The task is multilingual but the data distribution across languages is not the same. In all subtasks there is significantly more data for English than for Portuguese and Spanish. There is no training data provided for Hindi.

As there are no official train and development splits, we have created our own splits. The details are summarized in Table 1. For most task-language pairs, we randomly select 80% or 90% of the provided data as the training data and keep the remaining as the development data. Since there is much less data for Spanish and Portuguese, for some subtasks, such as subtask 3, we use the Spanish and Portuguese data for development only; and for subtask 4, we use the entire Spanish and Portuguese data as training for the multilingual model.

For the final submissions, we use all the provided data, and train various types of models (multilingual, monolingual, weakly supervised, zero-shot) with details provided in the appropriate sections.

3 Multilingual Transformer-Based Framework

For all the subtasks we use transformer-based language models (Vaswani et al., 2017) as the input text encoder. Recent studies show that deep transformer-based language models, when pre-trained on a large text corpus, can achieve better generalization performance and attain state-of-the-art performance for many NLP tasks (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020). One key success of transformer-based models is a multi-head self-attention mechanism that can model global dependencies between tokens in input and output sequences.

Due to the multilingual nature of this shared task, we have applied several multilingual transformer-based language models, including multilingual BERT (mBERT) (Devlin et al., 2019), XLM-RoBERTa (XLM-R) (Conneau et al., 2020), and multilingual BART (mBART) (Liu et al., 2020). Our preliminary experiments showed that XLM-R based models achieved better accuracy than other models. Hence we decided to use XLM-R as the text encoder. We use HuggingFace's pytorch implementation of transformers (Wolf et al., 2019).

XLM-R was pre-trained with unlabeled Wikipedia text and the CommonCrawl Corpus of 100 languages. It uses the SentencePiece tokenizer (Kudo and Richardson, 2018) with a vocabulary size of 250,000. Since XLM-R does not use any cross-lingual resources, it belongs to the unsupervised representation learning framework. For this work, we fine-tune the pre-trained XLM-R model on a specific task by training all layers of the model.

4 Subtask 1: Document Classification

To detect protest events at the document level, the problem can be formulated as a binary text classification problem where a document is assigned label "1" if it contains one or more protest event(s) and label "0" otherwise. Various models have been developed for text classification in general and also for this particular task (Hürriyetoğlu et al., 2019).

Model	en-dev	es-dev	pt-dev
XLM-R (en)	91.7	72.1	82.3
XLM-R (es)	85.4	71.9	83.9
XLM-R (pt)	85.5	75.2	84.8
XLM-R (en+es+pt)	90.0	75.2	88.3

Table 2: Macro F_1 score on the development sets for subtask 1 (document classification).

In our approach we apply multilingual transformer-based text classification models.

4.1 XLM-R Based Text Classification Models

In our architecture, the input sequence (document) is mapped to subword embeddings, and the embeddings are passed to multiple transformer layers. A special token is added to the beginning of the input sequence. This BOS token is <s> for XLM-R. The final hidden state of this token, $\mathbf{h_s}$, is used as the summary representation of the whole sequence, which is passed to a softmax classification layer that returns a probability distribution over the possible labels:

$$\mathbf{p} = \text{softmax}(\mathbf{W}\mathbf{h}_s + \mathbf{b}) \qquad (1)$$

XLM-R has $L = 24$ transformer layers, with hidden state vector size $H = 1024$, number of attention heads $A = 16$, and 550M parameters. We learn the model parameters using Adam (Kingma and Ba, 2015), with a learning rate of $2e$-5. We train the models for 5 epochs. Clock time was 90 minutes to train a model with training data from all the languages on a single NVIDIA V100 GPU.

The evaluation of subtask 1 is based on macro-F_1 scores of the developed models on the test data in 4 languages: English, Spanish, Portuguese, and Hindi. We are provided with training data in English, Spanish and Portuguese, but not in Hindi.

The sizes of the train/dev/test sets are shown in Table 1. Note that English has much more training data ($\sim$10k examples) than Spanish or Portuguese ($\sim$1k examples), while Hindi has no training data.

We build two types of XLM-R based text classification models:

- **multilingual model**: a model is trained with data from all three languages, denoted by XLM-R (en+es+pt);
- **monolingual models**: a separate model is trained with data from each of the three languages, denoted by XLM-R (en), XLM-R (es), and XLM-R (pt).

The results of various models on the development sets are shown in Table 2. We observe that:

- A monolingual XLM-R model trained with one language can achieve good zero-shot performance on other languages. For example, XLM-R (en), trained with English data only, achieves 72.1 and 82.3 F_1 score on Spanish and Portuguese development sets. This is consistent with our observations for other information extraction tasks such as relation extraction (Ni et al., 2020).
- Adding a small amount of training data from other languages, the multilingual model can further improve the performance for those languages. For example, with $\sim$1k additional training examples from Spanish and Portuguese, XLM-R (en+es+pt) improves the performance by 3.1 and 6.1 F_1 points on the Spanish and Portuguese development sets, compared with XLM-R (en).

4.2 Final Submissions

For English, Spanish and Portuguese, here are the three submissions we prepared for the evaluation:

S1: We trained five XLM-R based document classification models initialized with different random seeds using provided training data from all three languages (multilingual models). The final output for submission 1 is the majority vote of the outputs of the five multilingual models.

S2: For this submission we also trained five XLM-R based document classification models, but only using provided training data from the target language (monolingual models). The final output is the majority vote of the outputs of the five monolingual models.

S3: The final output of this submission is the majority vote of the outputs of the multilingual models built in (1) and the monolingual models built in (2).

For Hindi, there is no manually annotated training data provided. We used training data from English, Spanish and Portuguese, and augmented the

Model	en-dev	es-dev	pt-dev
XLM-R (en)	89.2	78.0	82.2
XLM-R (es)	84.4	86.4	80.1
XLM-R (pt)	83.2	82.2	85.1
XLM-R (en+es+pt)	89.4	86.2	85.6

Table 3: Macro F_1 score on the development sets for subtask 2 (sentence classification).

data with machine translated training data from English to Hindi ("weakly labeled" data). We trained nine XLM-R based Hindi document classification models with the weakly labeled data, and the final outputs are the majority votes of these models (S1/S2/S3 is the majority vote of 5/7/9 of the models, respectively).

5 Subtask 2: Sentence Classification

To detect protest events at the sentence level, one can also formulate the problem as a binary text classification problem where a sentence is assigned label "1" if it contains one or more protest event(s) and label "0" otherwise. As for document classification, we use XLM-R as the input text encoder. The difference is that for sentence classification, we set max_seq_length (a parameter of the model that specifies the maximum number of tokens in the input) to be 128; while for document classification where the input text is longer, we set max_seq_length to be 512 (for documents longer than 512 tokens, we truncate the documents and only keep the first 512 tokens). We train the models for 10 epochs, taking 80 minutes to train a model with training data from all the languages on a single NVIDIA V100 GPU.

For this subtask we are provided with training data in English, Spanish and Portuguese, and evaluation is on test data for all three languages. The sizes of the train/development/test sets are shown in Table 1.

As for document classification, we build two types of XLM-R based sentence classification models: a multilingual model and monolingual models. The results of these models on the development sets are shown in Table 3. The observations are similar to the document classification task. The multilingual model trained with data from all three languages achieves much better accuracy than a monolingual model on the development sets of other languages that the monolingual model is not trained on.

We prepared three submissions on the test data for each language (English, Spanish, Portuguese), similar to those described in section 4.2.

6 Subtask 3: Event Sentence Coreference Identification

Typically, for the task of event coreference resolution, events are defined by event triggers, and are usually marked in a sentence. Two event triggers are considered coreferent when they refer to the same event. In this task, however, the gold event triggers are not provided; the sentences are deemed coreferent, possibly, on the basis of any of the multiple triggers that occur in the sentences being coreferent, or if the sentences are about the same general event that is occurring. Given a document, this event coreference subtask aims to create clusters of coreferent sentences.

There is good variety in the research for coreference detection. Cattan et al. (2020) rely only on raw text without access to triggers or entity mentions to build coreference systems. Barhom et al. (2019) do joint entity and event extraction using a feature-based approach. Yu et al. (2020) use transformers to compute the event trigger and argument representation for the task.

Following the recent work on event coreference, our system is comprised of two parts: the classification module and the clustering module. The classification module uses a binary classifier to make pair-wise binary decisions on whether two sentences are coreferent. Once all sentence pairs have been classified as coreferent or not, the clustering module clusters the "closest" sentences with each other with agglomerative clustering, using a certain threshold, a common approach for coreference detection (Yang et al. (2015); Choubey and Huang (2017); Barhom et al. (2019)).

Agglomerative clustering is a popular technique for event or entity coreference resolution. At the beginning, all event mentions are assigned their own cluster. In each iteration, clusters are merged based on the average inter-cluster link similarity scores over all mentions in each cluster. The merging procedure stops when the average link similarity falls below a threshold.

Formally, given a document D with n sentences $\{s_1, s_2, ..., s_n\}$, our system follows the procedure outlined in Algorithm 1 while training. The input to the algorithm is a document, and the output is a list of clusters of coreferent event sentences.

Algorithm 1: Event Coreference Training

```
Input: $D = \{s_1, s_2, ..., s_n\}$, threshold $t$
Output: Clusters $\{c_1, c_2, ..., c_k\}$
Module Classify($D$):
    for $(s_i, s_j) \in D$ do
        Compute $sim_{i,j}$
        $SIM \leftarrow SIM \cup sim_{i,j}$
    return $SIM$

Module Cluster($D$, $SIM$,$t$):
    for $(s_i) \in D$ do
        Assign $s_i$ to cluster $c_i$
        Add $c_i \rightarrow C$
    moreClusters = True
    while moreClusters do
        moreClusters = False
        for $(c_i, c_j) \in C$ do
            score=0
            for $(s_k) \in c_i$ do
                for $(s_l) \in c_j$ do
                    score+=$sim_{k,l}$
            if $score > t$ then
                Merge $c_i$ and $c_j$
                Update $C$
                moreClusters = True
    return $C$
```

6.1 Experiments

The evaluation of the event coreference task is based on the CoNLL coref score (Pradhan et al., 2014), which is the unweighted average of the F-scores produced by the link-based MUC (Vilain et al., 1995), the mention-based B^3 (Bagga and Baldwin, 1998), and the entity-based CEAF$_e$ (Luo, 2005) metrics. As there is little Spanish and Portuguese data, we use it as a held out development set.

Our system uses XLM-R large pretrained model to obtain token and sentence representations. Pairs of sentences are concatenated to each other along with the special begin-of-sentence token and separator token as follows:

$$\text{BOS} < s_i > \text{SEP} < s_j >$$

We feed the BOS token representation to the binary classification layer to obtain a probabilistic score of the two sentences being coreferent. Once we have the score for all sentence pairs, we call the clustering module to create clusters using the coreference scores as clustering similarity scores.

Model	en-dev	es-dev	pt-dev
S1	83.4	93.3	80.4
S2	87.7	82.4	85.5
S3	88.8	81.7	91.7

Table 4: CoNLL F_1 score on the development sets for subtask 3: Event Coreference.

We use XLM-R large pre-trained models. We trained our system for 20 epochs with learning rate of 1e-5. We experimented with various thresholds and chose 0.65 as that gave the best performance on development set. It takes about 1 hour for the model to train on a single V100 GPU.

6.2 Final Submissions

For the final submission to the shared task we explore variations of the approach outlined in 6.1. They are:

S1: This is the multilingual model. To train this we translate the English training data to Spanish and Portuguese and train a model with original English, translated Spanish and translated Portuguese data. The original Spanish and Portuguese data is used as the development set for model selection.

S2: This is the English-only model, trained on English data. Spanish and Portuguese are zero-shot.

S3: This is an English-only coreference model where the event triggers and place and time arguments have been extracted using our subtask 4 models (section 7). These extracted tokens are then surrounded by markers of their type, such as <trigger>, <place>, etc. in the sentence. The binary classifier is fed the sentence representation.

The performance of these techniques on the development set is shown in table 4.

7 Subtask 4: Event Extraction

The event extraction subtask aims to extract event trigger words that pertain to demonstrations, protests, political rallies, group clashes or armed militancy, along with the participating arguments

Model	en-dev	es-dev	pt-dev
S1	80.57	-	-
S2	80.25	64.09	69.67
S3	80.87	-	-

Table 5: CoNLL F_1 score on the development sets for subtask 4: Event Extraction.

in such events. The arguments are to be extracted and classified as one of the following types: time, facility, organizer, participant, place or target of the event.

Formally the Event Extraction task can be summarized as follows: given a sentence $s = \{w_1, w_2, .., w_n\}$ and an event label set $T = \{t_1, t_2..., t_j\}$, identify contiguous phrases $(w_s, ..., w_e)$ such that $l(w_s, .., w_e) \in T$.

Most previous work (Chen et al. (2015); Nguyen et al. (2016); Nguyen and Grishman (2018)) for event extraction has treated event and argument extraction as separate tasks. But some systems (Li et al., 2013) treat the problem as structured prediction and train joint models for event triggers and arguments. Lin et al. (2020) built a joint system for many information extraction tasks including event trigger and arguments.

Following the work of M'hamdi et al. (2019); Awasthy et al. (2020), we treat event extraction as a sequence labeling task. Our models are based on the stdBERT baseline in Awasthy et al. (2020), though we extract triggers and arguments at the same time. We use the IOB2 encoding (Sang and Veenstra, 1999) to represent the triggers and the argument labels, where each token is labeled with its label and an indicator of whether it starts or continues a label, or is outside the label boundary by using *B-label, I-label* and *O* respectively.

The sentence tokens are converted to token-level contextualized embeddings $\{h_1, h_2, .., h_n\}$. We pass these through a classification block that is comprised of a dense linear hidden layer followed by a dropout layer, followed by a linear layer mapped to the task label space that produces labels for each token $\{l_1, l_2, .., l_n\}$.

The parameters of the model are trained via cross entropy loss, a standard approach for transformer-based sequence labeling models (Devlin et al., 2019). This is equivalent to minimizing the negative log-likelihood of the true labels,

$$L_t = -\sum_{i=1}^{n} \log(P(l_{w_i})) \quad (2)$$

7.1 Experiments

The evaluation of the event extraction task is the CoNLL macro-F_1 score. Since there is little Spanish and Portuguese data, we use it either as train in our multilingual model or as a held out development set for our English-only model.

For contextualized word embeddings, we use the XLM-R large pretrained model. The dense layer output size is same as its input size. We use the out-of-the-box pre-trained transformer models, and fine-tune them with the event data, updating all layers with the standard XLM-R hyperparameters. We ran 20 epochs with 5 seeds each, learning rate of $3 \cdot 10^{-5}$ or $5 \cdot 10^{-5}$, and training batch sizes of 20. We choose the best model based on the performance on the development set. The system took 30 minutes to train on a V100 GPU.

7.2 Final Submission

For the final submission to the shared task we explore the following variations:

S1: This is the multilingual model trained with all of the English, Spanish and Portuguese training data. The development set is English only.

S2: This is the English-only model, trained on English data. Spanish and Portuguese are zero-shot.

S3: This is an ensemble system that votes among the outputs of 5 different systems. The voting criterion is the most frequent class. For example, if three of the five systems agree on a label then that label is chosen as the final label.

The results on development data are shown in table 5. There is no score for S1 and S3 for *es* and *pt* as all provided data was used to train the S1 model.

8 Final Results and Discussion

The final results of our submissions and rankings are shown in Table 6. Our systems achieved consistently high scores across all subtasks and languages.

To recap, our S1 systems are multilingual models trained on all three languages. S2 are monolingual

Task - Language	Our Scores S1	S2	S3	Best Competitor Score	Our Rank
1 (Document Classification) - English	83.60	83.87	83.93	**84.55**	2
1 (Document Classification) - Portuguese	82.77	**84.00**	83.88	82.43	1
1 (Document Classification) - Spanish	73.86	**77.27**	74.46	73.01	1
1 (Document Classification) - Hindi	78.17	77.76	78.53	**78.77**	2
2 (Sentence Classification) - English	84.17	84.56	83.22	**85.32**	2
2 (Sentence Classification) - Portuguese	88.08	84.87	**88.47**	87.00	1
2 (Sentence Classification) - Spanish	**88.61**	87.59	88.37	85.17	1
3 (Event Coreference) - English	79.17	**84.44**	77.63	81.20	1
3 (Event Coreference) - Portuguese	89.77	92.84	90.33	**93.03**	2
3 (Event Coreference) - Spanish	82.81	**84.23**	81.89	83.15	1
4 (Event Extraction) - English	75.95	77.27	**78.11**	73.53	1
4 (Event Extraction) - Portuguese	**73.24**	69.21	71.5	68.14	1
4 (Event Extraction) - Spanish	**66.20**	62.02	66.05	62.21	1

Table 6: Final evaluation results and rankings across the subtasks and languages. Scores for subtasks 1 and 2 are macro-average F_1; subtask 3 are CoNLL average F_1; subtask 4 are CoNLL macro-F_1. The ranks and best scores are shared by the organizers. Bold score denotes the best score for the track.

models: for subtasks 1 and 2 they are language-specific, but for subtasks 3 and 4 they are English-only. S3 is an ensemble system with voting for subtasks 1, 2 and 4, and an extra-feature system for subtask 3. Among our three systems, the multilingual models achieved the best scores in three tracks, the monolingual models achieved the best scores in six tracks, and the ensemble models achieved the best scores in four tracks.

For subtask 1 (document-level classification), the language-specific monolingual model (S2) performs better than the multilingual model (S1) for English, Portuguese and Spanish; while for subtask 2 (sentence-level classification), the multilingual model outperforms the language-specific monolingual model for Portuguese and Spanish. This shows that building multilingual models could be better than building language-specific monolingual models for finer-grained tasks.

The monolingual English-only model (S2) performs best on all three languages for subtask 3. This could be because the multilingual model (S1) here was trained with machine translated data. Adding the trigger, time and place markers (S3) did not help, even when these features showed promise on the development sets.

The multilingual model (S1) does better for Spanish and Portuguese on subtask 4. This is consistent with our findings in Moon et al. (2019) where training multilingual models for Named Entity Recognition, also a token-level sequence labelling task, helps improve performance across languages. As there is much less training data for Spanish and Portuguese, pooling all languages helps.

9 Conclusion

In this paper, we presented the models and systems we developed for Multilingual Protest News Detection - Shared Task 1 at CASE 2021. We explored monolingual, multilingual, zero-shot and ensemble approaches and showed the results across the subtasks and languages chosen for this shared task. Our systems achieved an average F_1 score of 81.2, which is 2 F_1 points higher than best score of other participants on the shared task. Our submissions ranked 1^{st} in nine of the thirteen tracks, and ranked 2^{nd} in the remaining four tracks.

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

Parul Awasthy, Tahira Naseem, Jian Ni, Taesun Moon, and Radu Florian. 2020. Event presence prediction helps trigger detection across languages. *CoRR*, abs/2009.07188.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189, Florence, Italy. Association for Computational Linguistics.

Arie Cattan, Alon Eirew, Gabriel Stanovsky, Mandar Joshi, and Ido Dagan. 2020. Streamlining cross-document coreference resolution: Evaluation and modeling.

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

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2133, Copenhagen, Denmark. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ali Hürriyetoğlu, Osman Mutlu, Farhana Ferdousi Liza, Erdem Yörük, Ritesh Kumar, and Shyam Ratan. 2021a. Multilingual protest news detection - shared task 1, CASE 2021. 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).

Ali Hürriyetoğlu, Hristo Tanev, Vanni Zavarella, Jakub Piskorski, Reyyan Yeniterzi, and Erdem Yörük. 2021b. 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).

Ali Hürriyetoğlu, Erdem Yörük, Deniz Yüret, Çağrı Yoltar, Burak Gürel, Fırat Duruşan, Osman Mutlu, and Arda Akdemir. 2019. Overview of clef 2019 lab protestnews: Extracting protests from news in a cross-context setting. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 425–432, Cham. Springer International Publishing.

Ali Hürriyetoğlu, Vanni Zavarella, Hristo Tanev, Erdem Yörük, Ali Safaya, and Osman Mutlu. 2020. Automated extraction of socio-political events from news (AESPEN): Workshop and shared task report. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 1–6, Marseille, France. European Language Resources Association (ELRA).

Ali Hürriyetoğlu, Erdem Yörük, Osman Mutlu, Fırat Duruşan, Çağrı Yoltar, Deniz Yüret, and Burak Gürel. 2021. Cross-Context News Corpus for Protest Event-Related Knowledge Base Construction. *Data Intelligence*, pages 1–28.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, ICLR '15.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82.

Ying Lin, Heng Ji, F Huang, and L Wu. 2020. A joint neural model for information extraction with global features. In *Proc. The 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

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. *CoRR*, abs/1907.11692.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Meryem M'hamdi, Marjorie Freedman, and Jonathan May. 2019. Contextualized cross-lingual event trigger extraction with minimal resources. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 656–665, Hong Kong, China. Association for Computational Linguistics.

Taesun Moon, Parul Awasthy, Jian Ni, and Radu Florian. 2019. Towards lingua franca named entity recognition with BERT. *CoRR*, abs/1912.01389.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-second AAAI conference on artificial intelligence*.

Jian Ni, Taesun Moon, Parul Awasthy, and Radu Florian. 2020. Cross-lingual relation extraction with transformers. *CoRR*, abs/2010.08652.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 30–35, Baltimore, Maryland. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, page 173–179, USA. Association for Computational Linguistics.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the The 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. 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.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, page 45–52, USA. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A Hierarchical Distance-dependent Bayesian Model for Event Coreference Resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.

Xiaodong Yu, Wenpeng Yin, and Dan Roth. 2020. Paired representation learning for event and entity coreference.