SU-NLP at CASE 2021 Task 1: Protest News Detection for English

Furkan Çelik, Tuğberk Dalkılıç, Fatih Beyhan, Reyyan Yeniterzi
Sabancı University
İstanbul, Turkey
{fcelik, tdalkilic, fatihbeyhan, reyyan}@sabanciuniv.edu

Abstract

This paper summarizes our group's efforts in the multilingual protest news detection shared task, which is organized as a part of the Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) Workshop. We participated in all four subtasks in English. Especially in the identification of event containing sentences task, our proposed ensemble approach using RoBERTa and multichannel CNN-LexStem model yields higher performance. Similarly in the event extraction task, our transformer-LSTM-CRF architecture outperforms regular transformers significantly.

1 Introduction

Identifying events and extracting event related information from text is an important language understanding task which has been studied for quite some time. This challenging task has been studied in several steps or divided into some sub-tasks. The first step is identifying whether a document or a sentence contains an event or not. If it contains then the event co-reference resolution task analyses whether the context around it (such as other sentences) refer to the same event or not. Event related information such as the event trigger and its arguments are also extracted, which can be later on used to create event taxonomies.

These steps either alone or together have been studied for English extensively, similar to many other Natural Language Processing tasks. This year as part of the Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) Workshop, a shared task covering some of these sub-tasks has been organized not only for English but also for Portuguese, Spanish and Hindi (Hürriyetoğlu et al., 2021). The common theme was the identification of protest events from news articles.

The organizers specifically focus on the four subtasks. In the first and second sub-task, the aim is to predict whether a given document (subtask 1) or sentence (subtask 2) contains information about an event (either past or ongoing). The third subtask focuses on event sentence coreference and the participants are asked to predict whether the sentences containing an event are referring to the same event or not. In subtask 4, the goal is to identify event triggers and related arguments from sentences.

It is hard to choose among these interesting subtasks, therefore we participate in all four of them. Due to time constraints we only work on English and leave the rest of the languages as future work.

The first and the second subtask focus on predicting whether a content contains an event or not. For these tasks in addition to trying standard transformer based models, we explore ensemble models which combine the strengths of different models. Furthermore, the effect of stemming the context is also explored in these subtasks. The third subtask is related to the event coreference task. For this task, we explore the rescoring and clustering approach proposed by (Örs et al., 2020). Finally, the goal of subtask 4 is to extract event information from context. For this task, we exploit the transformer-LSTM-CRF architecture which has shown success in several NER tasks.

The rest of the paper is organized as following: Section 2 describes our proposed approach for identifying whether a content contains an event or not, and details our submissions for subtasks 1 and 2. Section 3 explains our submission to the event coreference resolution subtask. Section 4 presents the experimental results for event extraction subtask and finally Section 5 concludes the paper with future work.

2 Subtask 1 & 2: Event or Not

The goal of the first two subtasks is to predict whether the provided input context contains an

Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) , pages 131–137
August 5–6, 2021, ©2021 Association for Computational Linguistics

event (either past or ongoing) or not. Therefore, the task is a binary classification task. In these two subtasks the only difference is the input context. In subtask 1 the input is the whole news article while in subtask 2, it is only a sentence. The main difference between these two tasks is the length of the input. In subtask 1's dataset, even though most documents contain around 3 sentences, the maximum length in the data is almost 10 times larger than the maximum length in subtask 2 data. This makes subtask 1 slightly more challenging. One expects documents as longer input, to contain more clues about an event if there is; therefore more useful. However, there is also the risk of unrelated content causing mixed signals.

Even though this difference between the tasks, we mostly apply same approaches to both. For this binary classification problem, we use some simple neural network architectures as baselines and also investigate fine-tuning several pretrained transformer based models. The models applied are listed as follows:

- CNN: A single convolutional layer connected to a fully connected dense layer.
- LSTM: A unidirectional long short term memory model.
- GRU: A unidirectional gated recurrent unit model.
- BERT (Devlin et al., 2019): Uses bidirectional transformer architecture for language modeling. We fine-tune the BERT-base-cased [1] model.
- Albert (Lan et al., 2019): An efficient (A Lite BERT) version of BERT which outperformed BERT in several benchmark data sets. We fine-tune the Albert-base-v2 model [2] in this paper.
- RoBERTa (Liu et al., 2019): A robustly optimized version of BERT which outperformed BERT in GLUE benchmark. We fine-tune the RoBERTa-base model [3] in our experiments.

For neural networks like CNN and RNN, several pretrained word embeddings, like Google News Word2Vec[4] (Mikolov et al., 2013), NNLM (Bengio et al., 2003) model trained on Google News dataset [5] and GloVe (Pennington et al., 2014) 6B Wikipedia embeddings [6], have been tried. Since the ratio of out-of-vocabulary words were very small, character-based embeddings have not been explored. We have seen that using different embeddings resulted in minor changes, and rather fine-tuning the embedding layer or not, does not have any significant effect on the performance of models in terms of overfitting resistance or achieved scores.

NNLM and GloVe return slightly better performance compared to Word2Vec, when used in standalone CNN or RNN models. However, as we try ensembling approaches (to be described in the upcoming sections), NNLM outperforms GloVe with its high Precision score. Therefore, NNLM embedding is used in all reported experiments in this section.

2.1 Baseline Experiments

In all these subtasks, the data collections were gathered from news articles about socio-political and crisis conflicts. For the document classification task, we are provided with an imbalance training data of 9324 news articles with 7407 of them without any events and the rest as containing event. Similarly in subtask 2, among the provided 22825 sentences, only 4210 of them contain an event while the rest of them do not.

For both tasks, 20% of the provided data is used for validation purposes and rest for model training. During the training process, several balancing approaches were applied to decrease any possible negative effects caused by the imbalance data problem. But overall they did not provide any significant improvements in F1 score; therefore data is used in its original ratio without any balancing.

The experimental results of the baseline approaches are displayed in Tables 1 and 2. In subtask 1, except for RNNs, all methods listed above were tested. RNNs were not tested due to limited time and prioritization of computational resources for other more advance models. Only a single layer CNN is used in the experiments, since adding more

[1] https://huggingface.co/bert-base-cased

[2] https://huggingface.co/albert-base-v2

[3] https://huggingface.co/roberta-base

[4] https://radimrehurek.com/gensim/auto_examples/howtos/run_downloader_api.html

[5] https://tfhub.dev/google/nnlm-en-dim128/2

[6] https://nlp.stanford.edu/projects/glove/

layers caused over-fitting.

Model	Validation Set	Test Set
CNN	0.82	0.77
BERT	0.84	0.80
Albert	0.84	0.81
RoBERTa	0.86	0.81

Table 1: Subtask 1 Baseline Approaches F1 Scores

Model	Validation Set	Test Set
CNN	0.80	0.70
LSTM	0.82	0.68
GRU	0.83	0.64
BERT	0.87	0.81
Albert	0.86	0.81
RoBERTa	0.88	0.82

Table 2: Subtask 2 Baseline Approaches F1 Scores

Based on the results, transformer based approaches outperform classical neural network based approaches in both tasks. In traditional neural network based models, RNN based ones, both LSTM and GRU, suffer from serious overfitting even though all the efforts of regularization and dropout. Regarding the transformer-based models, in both subtasks, RoBERTa outperforms both BERT and Albert with close margin.

2.2 LexStem Model

In the task definition, it is mentioned that the labeled events can be either from past or continuous. This suggests various types of tense use in the context. This variety may cause model to miss some events. In order to deal with this variety, in addition to the lexical forms of the words, their stemmed versions are also included to CNN model as additional channel in the network. WordNetLemmatizer [7] is used as the stemmer. In this proposed model, which is named as *LexStem* model, one channel is used for the original form of the sentence and another channel for the stemmed version.

In order to make a fair comparison of the LexStem model, additional CNN multi-channel models are trained as well.

- CNN-LexLex: A two channels model with original form of the words are used in both channels. This one is developed to see the effect of two channels compared to one.
- CNN-StemStem: A two channels model with stemmed version of the words are used in both channels. This one is developed to see the individual effect of stem information.
- CNN-LexStem: The proposed two channel model with one channel for lexical form of the word and the other for stemmed version.

The experimental results of these models are displayed in Table 3. In the table, the first two rows are from subtask 1 and the rest of them are from subtask 2. The proposed LexStem model does not provide any significant improvements in subtask 1, therefore other multi-channel models are not tested with this task.

ST	Model	#CH	Val.	Test
1	CNN	1	0.82	0.77
1	CNN-LexStem	2	0.82	0.78
2	CNN	1	0.80	0.70
2	CNN-LexLex	2	0.82	0.69
2	CNN-StemStem	2	0.83	0.68
2	CNN-LexStem	2	0.85	0.71

Table 3: Subtask 1 & 2 Stemming Experiments F1 Scores. ST: Subtask and CH: Channel

Unlike subtask 1, for subtask 2 the LexStem model provides drastic improvements with validation data, but only slight improvement on test data. A similar improvement on test set is also observed at subtask 1. Using multi-channel architecture and therefore using more parameters probably increases model's likelihood of overfitting. This is more observable with CNN-LexLex and CNN-StemStem models. Even though with this increased overfitting possibility, CNN-LexStem model returns small yet consistent increase on test set. The possible reasons of this improvement will be explored more in the future.

2.3 Ensemble Models

RoBERTa model outperforms all other models, therefore we specifically analyze its performance and its confidence of its predictions on the validation set. Figure 1 displays how the average F1 score changes with respect to model's confidence values. In the figure, 0.05-0.95 means RoBERTa's predictions which are lower than 0.05 or higher than 0.95.

According to the Figure 1, confidence scores lower than 10% and higher than 90% achieve the

[7] https://www.nltk.org/_modules/nltk/stem/wordnet.html

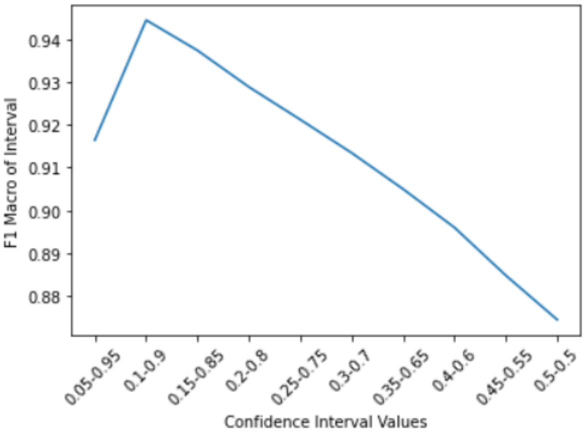

Figure 1: Subtask 2: Confidence Intervals and Their Respective Macro F1 Scores Calculated over Validation Set

Model	Validation	Test
LSTM	0.82	0.68
GRU	0.83	0.64
CNN-LexStem	0.85	0.71
RoBERTa	0.88	0.82
RoBERTa+RNN	0.89	0.83
RoBERTa+LexStem	0.88	0.84

Table 4: Subtask 2 - Ensemble Models F1 Macro Scores

highest Macro F1 score of 94% and after this, as confidence values go below 90% or above 10%, the F1 score consistently decreases. This means that as RoBERTa gets more unsure of its predictions, it is making more mistakes as expected. In order to prevent these errors, ensemble models are explored.

A weighted ensemble model is applied for any case in which RoBERTa is not confident. After trying several threshold values, 0.1 and 0.9 is chosen. Cases where RoBERTa's output are higher than 0.9 or lower than 0.1, are accepted as they are. For anything in between, an ensemble model is used. In order to find the right models to ensemble, a grid search is applied. RoBERTa is assumed to be the permanent model in this ensemble. Therefore, the search is performed over other models as either individual or in groups of two. The following models and weights return the highest performance for subtask 2:

- RoBERTa-RNN: 0.4 RoBERTa + 0.15 LSTM + 0.45 GRU
- RoBERTa-LexStem: 0.45 RoBERTa + 0.55 CNN-LexStem

The performance of these ensembles together with individual model performances are presented in Table 4. The ensemble model is only applied for subtask 2. As for subtask 1, we don't have any RNN model to ensemble or the CNN-LexStem did not provide any improvement on the validation set.

According to Table 4, both ensembles outperform RoBERTa both in the validation and test sets. This indicates that different types of neural networks have different powers, and in case when a model is not confident; using a weighted voting and combining these powers can be useful.

In conclusion, for subtask 1 RoBERTa is the top performing model based on the validation set and it is ranked the 3rd place in the public leaderboard. For subtask 2, our ensemble models receive the 3rd rank in the leaderboard.

3 Subtask 3: Event Sentence Coreference Identification

In event sentence coreference task, event containing sentences in a document are analyzed to see whether they refer to the same event or not. This task is slightly different than other ones as it does not only consist of a classification step, but also requires clustering afterwards. This two step procedure is known as the Mention-Pair model (Ng, 2010) in coreference resolution tasks. The first step includes a binary classification model to classify pairs of mentions and the second step uses these predictions to determine the coreference relations by clustering them (Ng, 2010). In this paper, we also use the two step approach, and first perform pairwise classification of sentences and then cluster them.

3.1 Two-Step Approach

For the classification part, similar to previous subtasks, base models of BERT, ALBERT and RoBERTa are fine-tuned. Additionally, an ensemble model which is a probabilistic average of these three models, is developed. In all these four binary classification models, instead of using the regular 0.5 boundary, 0.6 boundary is used to identify the positive labels, since 0.6 threshold returned better performance in our experiments.

For the clustering step, (Örs et al., 2020)'s clustering approach together with their proposed rescoring algorithm is used. Their rescoring algorithm calculates an updated score for a pair of sentences by using how sentences within the pair interact with other sentences in the document. For instance, the following pair of sentences, s_1 and s_2, has positive label predicted. If the predicted label between s_1 and s_3 is same as the prediction between s_2 and s_3, then a reward is given to s_1 and s_2 pair. But if the labels are different, then a penalty is applied. After the scores are updated, a greedy agglomerative algorithm is applied to construct the clusters (Örs et al., 2020). The same rescoring and clustering approach is used in this paper as well.

3.2 Experimental Setting

The main evaluation metric for this subtask is different than the other three. CoNLL metric, which is widely used on event/entity coreference tasks, is used in this task for the final system rankings. CoNLL is the average of MUC score (Vilain et al., 1995), B^3 score (Bagga and Baldwin, 1998) and $CEAF_e$ score (Luo, 2005).

The provided English dataset consists of 596 documents with their event containing sentences and gold clusters. This dataset is divided into training (80%) and validation (20%) sets. Unlike other tasks, this data split is performed more carefully to make sure that various types of clusters are observed in both training and validation sets. While creating these splits, two ratios are calculated and observed. The first one is the *single cluster ratio* which is calculated by dividing the number of documents with only one cluster to the total number of documents. The second one is referred to as *positive class ratio* which is calculated by dividing the number of sentence pairs with positive labels into total number of sentence pairs.

Having training and validation splits with very different *single cluster ratio* may affect the performance of clustering step. Similarly having a different *positive class ratio* may affect the classification performance. Hence, we tried different seeds for random splitting to find the splits which are similar to each other in terms of both of these ratios. The statistics of the constructed splits are presented in Table 5.

In addition to the provided training data, we also explore an external dataset from a similar shared task which was organized in 2020. AESPEN'20[8] shared task also focused on event sentence coreference identification and publicly shared a training data of 404 English news articles with their gold-standard labels. We explore the effects of using this dataset as an extension to the existing one. In our experiments this year's provided dataset is referred to as *RAW*, and the extended version which contains data from both CASE and AESPEN is called *EXT*.

	Train	Validation
# Documents	476	120
# Sentences	2041	538
# Sentence Pairs	4918	1323
Positive Class Ratio	68%	69%
Single Clusters Ratio	61%	64%

Table 5: Statistics of the Training and Validation Sets

3.3 Experiments

Classification results of our models on validation set can be seen in Table 6. As expected, all models perform much better with the extended dataset. In general, BERT performs slightly better than the others. The Ensemble model cannot outperform BERT, but it is the second best, therefore we keep using it.

Model	RAW Data	EXT Data
BERT	86.98	92.42
ALBERT	85.74	91.57
RoBERTa	86.68	90.13
Ensemble	86.49	92.14

Table 6: Subtask 3: F1 Macro Scores of Classification Step over Validation Set

Errors of the classification step will unfortunately propagate to the next step, which is clustering. Since some of the pairwise sentences' labels are wrong, the constructed clusters will likely be wrong as well. In order to decrease the effect of this error propagation, we use the best two models from the classification step in this clustering part. The results of the BERT and the Ensemble models are summarized in Table 7.

Model	Data	Validation	Test
BERT	RAW	77.70	74.83
Ensemble	RAW	79.01	74.27
BERT	EXT	80.54	78.45
Ensemble	EXT	80.03	78.66

Table 7: Subtask 3: CONLL Scores after Clustering

As expected, models trained on the extended

[8] https://emw.ku.edu.tr/aespen-2020/

(larger) dataset return consistently higher scores. Between the BERT and the Ensemble model, there isn't a clear winner. However, in test set the highest score is retrieved with the Ensemble model which is ranked the 5th in the public leaderboard.

4 Subtask 4: Event Extraction

The goal of the final subtask is to identify the event triggers and its arguments from the sentence. The training dataset consists of 808 sentences which contain IOB type token-based labels of 7 different labels. Similar to previous tasks, 20% of this data is used for validation and the rest for training purposes.

In many sequence modeling tasks, the bidirectional transformer models outperform other machine learning architectures; therefore, BERT and RoBERTa are used as strong baselines in this task. As a further development, the transformer model is connected with a BiLSTM and a CRF layer as our second architecture. Connecting BiLSTM and CRF to a transformer has shown success in several Named Entity Recognition tasks (Jiang et al., 2019; Dai et al., 2019). The performance of these models over both validation and test sets are presented in Table 8.

Model Name	Validation	Test
BERT	0.70	0.69
RoBERTa	0.72	0.74
BERT-BiLSTM-CRF	0.76	0.75
RoBERTa-BiLSTM-CRF	0.76	0.76

Table 8: Subtask 4: F1 Macro Scores

According to Table 8, RoBERTa outperforms BERT in both validation and test sets. Combining these with BiLSTM-CRF improves both of them. The performance difference between test and validation sets also decreases with this addition.

Even though we achieved good performance, due to a minor format issue at our test submission file, our submissions were not correctly evaluated. Based on our scores at Table 8, with our best model RoBERTa-BiLSTM-CRF, we would have ranked second in the public leaderboard.

Analyzing the individual tag performances revealed that model is doing a better job at identifying the triggers compared to its arguments. This is expected as trigger tag is the second most popular tag at the data after the O tag. Trigger is closely followed by event time, which is easier to predict due to its smaller vocabulary variance and common language patterns, even though its lower presence in the training data.

In order to analyze the weak points of the models, the confusion table of the top performing RoBERTa-BiLSTM-CRF model over the validation data is shown in Figure 2. The confusion matrix specifically focuses on the event trigger and arguments tags.

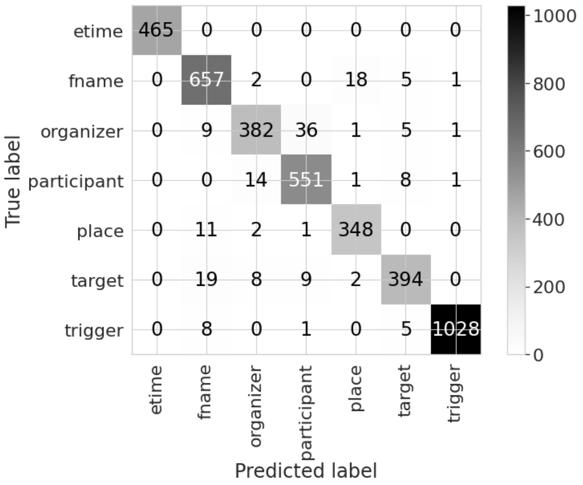

Figure 2: Confusion Table for Event Trigger and Arguments Tags

Based on Figure 2, the *etime* (event time) is the tag which has not been mistaken with any other event specific tags. On the other hand, the highest confusion is between the *organizer* and *participant* tags. That is followed by *place* and *fname* (facility name) which is expected due to use of similar wordings and context around.

5 Conclusion

In this paper, we mainly focus on English, and try to improve the current state-of-the-art on event specific NLP tasks. Source codes of all of our models are available online [9]. Additional details of our models, like hyper-parameters, are also summarized in the Github. As future work, we will focus on other languages and see whether the trends observed with English, exist in those other languages as well.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. pages 563–566.

[9] https://github.com/furkan-celik/CASE21-SuNLP-Models

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The journal of machine learning research*, 3:1137–1155.

Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, and Xuming Bai. 2019. Named entity recognition using bert bilstm crf for chinese electronic health records. In *2019 12th international congress on image and signal processing, biomedical engineering and informatics (cisp-bmei)*, pages 1–5. IEEE.

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. 2021. 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).

Shaohua Jiang, Shan Zhao, Kai Hou, Yang Liu, Li Zhang, et al. 2019. A bert-bilstm-crf model for chinese electronic medical records named entity recognition. In *2019 12th International Conference on Intelligent Computation Technology and Automation (ICICTA)*, pages 166–169. IEEE.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

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.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. pages 6–8.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Vincent Ng. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1396–1411.

Faik Kerem Örs, Süveyda Yeniterzi, and Reyyan Yeniterzi. 2020. Event clustering within news articles. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 63–68, Marseille, France. European Language Resources Association (ELRA).

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. pages 45–52.