Automated Extraction of Socio-political Events from Text

Dominik Stammbach
ETH Zürich
dominik.stammbach@gess.ethz.ch

Didem Durukan
University of Zurich
eminedidem.durukan@student.ethz.ch

He Liu
University of Zurich
heliuhe@student.ethz.ch

1 Introduction

The need for precise and high-quality information about a wide variety of events ranging from political violence, environmental catastrophes, conflict, to international economic and health crises has been increasing. Furthermore, the documentation of such conflicts, e.g. through traditional media, NGOs and social media also is at an all time high, the further the digital age proceeds. This leads to an ever-growing amount of conflicts being described in an abundance of data, and thus calls for automation of extracting such information at scale.The resulting data is important for preventing or resolving conflicts, or improving the lives of and protect citizens in a variety of ways. Researchers also have an interest, as this data can lead to valuable social science experiments and provide insights to prevent further conflicts. These reasons combined are the cause why governments, local and global NGOs and researchers might be highly interested in gathering such socio-political information about events.

We hope that Automated Extraction of Socio-political events provides the resources for large scale socio-political event information collection across sources, countries, and languages (Hürriyetoğlu et al., 2020). Hence our choice of tackling this problem and dataset. In this project, our goal is to extract these socio-political events from raw text. We use manually annotated data, namely the CASE dataset[1], which has been used to organize a shared task and a workshop event at ACL-IJCNLP 2021.

The dataset is described in more detail in (Hürriyetoğlu et al., 2020a). As mentioned in the proposal we have four subtasks that we want to work on (we only tackle the English version of the subtask – the dataset offers the same task across different langauges and provides another two tasks which we did not conisder because this might go beyond the scope of this project).

1. Document classification (Does a news article contain information about a past or ongoing event?)

2. Sentence classification (Does a sentence contain information about a past or ongoing event?)

3. Event sentence coreference identification (Which event sentences (subtask 2) are about the same event?)

4. Event extraction (What is the event trigger and its arguments?)

Given the annotated data, we tackle the task as a supervised learning problem, using recent methods in NLP, namely transformer-based architectures such as RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021) and BigBird (Zaheer et al., 2021).

The code of our project is available on GitHub[2], and we provide a recorded presentation about the project[3].

2 The Tasks in More Detail

2.1 Subtask 1

Subtask 1 is a straight-forward sequence classification problem. We have 9.3K annotated English documents (and 1K Spanish, 1.5K Portuguese documents). The task is to determine whether a document contains information about an event. The mean length of the documents is 171 tokens. Furthermore, the class distribution is imbalanced and

[1] https://emw.ku.edu.tr/case-2021/

[2] https://github.com/codingFerryman/computational_semantics_for_NLP

[3] https://www.dropbox.com/s/uo492nu2wwzomvz/zoom_0.mp4?dl=0

System	Task 1 (F1)	Task 2 (F1)
Random baseline	45.79	45.06
TF-IDF SVM	75.45	69.55
RoBERTa-base	85.97	88.91
BigBird-base	87.01	88.43
Roberta-base pretrained on Task 1	na	88.74
BigBird-base pretrained on Task 1	na	88.48
BigBird-large	87.38	n/a
Debertav2-xl-mnli	n/a	89.42

Table 1: First results

we only have 1.9K documents in which an event of interest is described (20% of all documents). An example of a document containing an event of interest is *"PAU staff's hunger strike continues - Indian Express Express News Service , Express News Service : Ludhiana, Tue Dec 22 2009, 04:49 hrs The chain hunger strike against the s"*. Judging from this example, we see that the data has probably been automatically preprocessed and is not entirely clean.

2.2 Subtask 2

Subtask 2 is similar in spirit. We are trying to answer the following question: Does a sentence contain information about a past or ongoing event, hence it is a sequence classification task again, but on shorter sequences of text. An example of a sentence of interest is *"There was no formal notice of strike and this has created a lot of hassle for the public."*.

2.3 Subtask 3

In this subtask, we are expected to determine the relationship between events and cluster the sentences in paragraphs based on the event categories they include. Each event will be numbered from 1 and on in accordance with the order in which they appear in the article, and the separation of event references will be based on difference in at least one of the following: event time, event place, facility (name or type), or participants. For example, in the article *"Whitfield said the part fortnight has seen highways blocked, 20 trucks burnt and a driver die of injuries sustained when his truck was petrol bombed near Touws River in the Western Cape.","He said an attack on the transport of goods was an attack on the economy.","South African truck drivers have embarked on a nationwide strike in protest at foreign drivers allegedly taking away their jobs."]*, the first two sentences are about attack so they in the same cluster, and the last sentences is about strike so it is in another cluster.

This task is a coreference resolution problem, and the analyzing process includes: (a) finding out the event tokens in sentences; (b) figuring out the coreference between event tokens in different sentences; (c) clustering sentences from the coreference.

2.4 Subtask 4

Finally, in subtask4 we plan to extract fine-graned arguments of interest of a specific event, e.g. "What is the event trigger and its arguments?" This again is a classification task, but we produce a classification for each token in a sequence.

To evaluate, we use the official macro-F1 for evaluating subtasks 1 and 2, and accuracy for subtask 4. We report these metrics for a random split of the train set (75%-25% train-test split) which we used to determine the architecture choice. And we also report results of our system's performance on the official blind testset[4] of the shared task.

3 Methods

In this work, we explore Transformer models to automatically extract socio-political events from text. Specifically, we use RoBERTa (Liu et al., 2019), DeBERTav2 (He et al., 2021) and BigBird (Zaheer et al., 2021) in subtask 1, 2, and 4. These all implement the Transformer encoder from (Vaswani et al., 2017), an architecture originally proposed for machine translation. Transformers let go of convolution or recurrence and only use attention mechanisms – and a Transformer consists of stacked transformer blocks implementing this self-attention followed by a feed-forward layer. For the equations

[4] https://competitions.codalab.org/competitions/31639#results

and the exact mechanism, we refer to the original paper.

Radford and Narasimhan (2018) have shown that combining a transformer encoder and pre-train it on language modeling on large amounts of text yields amazing tranferrable capabilities of such models, which then can be fine-tuned on any NLP task. This insight has (literally) transformed the whole NLP landscape. The last big gains have been made via introduction of bidirectional attention in pre-training, that is to mask tokens in a sequence instead of always predicting the last token (Devlin et al., 2019).

RoBERTa (Liu et al., 2019) just trains BERT on more text, while letting go of the next sentence prediction task in the original BERT paper. DeBERTa (He et al., 2021) slightly modifies the attention mechanism and disentangles the word embedding from the positional embedding, leading to modest improvements over RoBERTa on a large number of NLU tasks, e.g. in the GLUE benchmark. Larger models are better, this is one of the lame insights working in NLP these days – and it just so happens that the DeBERTav2-xl-mnli checkpoint fits nicely on a 32GB gpu, which is the whole motivation to also include this checkpoint in our experiments.

Lastly, BigBird (Zaheer et al., 2021) does not compute the whole attention matrix A anymore, e.g. the interaction of all tokens in a sequence with each other. The authors show that it suffices to only compute neighboring interactions (window attention), global interactions for some tokens, by default [CLS] and [SEP] token (global attention) and some randomly sampled interactions for each token (random attention). The BigBird attention pattern then is composed of these three attention types and achivees BERT-like capabilities on short sequences and makes vast improvements on a large number of longer sequences. Because e.g. documents in our training data are often longer than 512 subwords (the maximum amount of tokens allowed in BERT), we use BigBird in such cases.

Obviously, we are aware that there exist many more high-performance transformer architectures which we did not consider in our experiments. We believe we provided a compelling rationale for our final model choice – and given the results achieved and the scope of the project, we are under the assumption that this suffices as is.

In subtask 3, we explore the SpanBERT model (Joshi et al., 2020). SpanBERT does not mask tokens randomly, but adds masks to contiguous random spans. Training SpanBERT bases on the boundary of spans (known as Span Boundary Objective), which is useful for coreference resolution problems. The author shows that it can consistently outperforms BERT and their better-tuned baselines. Additionally, a new perspective of higher-order inference (HOI)(Xu and Choi, 2020), has a positive impact on SpanBERT for the task of coreference resolution. Cluster Merging (CM), the best-performed HOI method in this paper, performs sequential antecedent ranking combining both antecedent and entity information to gradually build up the entity clusters (Xu and Choi, 2020). Due to the lack of time, we will only focus on how to figure out the event mentions and feed the training data to the models.

We used the HuggingFace transformers library for all the experiments in subtask 1, 2, and 4. (Wolf et al., 2020).

4 Experiments and Development Set Results

4.1 Subtask 1 and 2

We fine-tuned two transformer models on subtask 1 (document classification) and subtask 2 (sentence classification). We find that both transformer approaches outperform a TF-IDF SVM by at least 10% points. BigBird works better on subtask 1, presumably because many of the documents exceed the maximum sequence length of 512 subwords for RoBERTA. BigBird has access to this additional information which seems to help for this task. For sentence classification (shorter sequences) RoBERTa and BigBird perform very similarly, which is to be expected since both have access to the whole sequence. For the sentence classification task, we find that larger models perform better, and we obtain the best results using a Debertav2-xl model which was previously fine-tuned on the MNLI dataset.

Additionally, we further fine-tuned our checkpoints obtained by training on subtask 1 on subtask 2, however we get close to identical results (slightly worse in fact). We speculate that subtask 2 is contained in itself and models converge to a decent solution in any case, without the need for transferring models fine-tuned on document classification first. We observe this pattern for both transformer models.

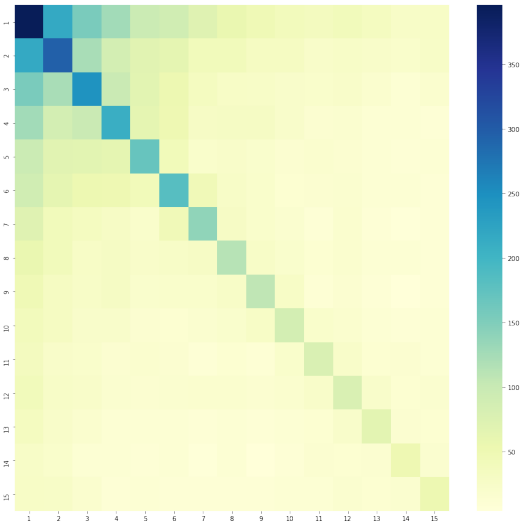

Figure 1: Subtask 3 Co-occurrence Matrix for *sentence_no* 1-15

4.2 Subtask 3

We explored the training data of this subtask and found that there are 62 different *sentence_no* in the training data's 596 entities and the larger the number the fewer the occurrence (Figure 1). The numbers are not consecutive and the numbers larger than 15 occur occasionally. Some numbers even exist in only one cluster or always be clustered standalone.

There are three baseline models in this subtask, and two of them are dummy models. The first dummy model assigns all event sentences to a single cluster all the time (MinC). The other is also a dummy model, which assigns each event sentence as a separate cluster in an article (MaxC). The last baseline model preprocesses the data into pairwise sentences and then builds a multilayer perceptron (MLP) model to predict if they are in the same cluster.

In this subtask we use similarity (Cai and Strube, 2010) (CEAF-e and CEAF-m) as the scorer. The evaluation results of those baseline models are in Table 2. The result shows that put more sentences into one cluster is more preferable to separating them. The MLP model cannot outperform MinC.

We evaluated a pre-trained SpanBERT model, but its performance is even worse than always clustering all sentences together. This model preferably outputs smaller clusters, which is not the case for training data.

Therefore, it is necessary to do transfer learning and fine-tuning. To find the real coreference entities, we use the data and the model trained in subtask 4 to fetch or predict the NER tags. We found that more than 97% traning articles have at least one *trigger* tag, and only 1.20% of the sentences have no tags in *trigger, participant, organizer,* and *target*, so we can assume that if two sentences' events coreference each other, the words in them with the same NER tags also coreference each other. We also assume that there is no coreference between other words. A sentence that does not have those tags (e.g., only has "*O*") will be put into its previous sentence's cluster. In the prediction, we will first predict if there is any coreference on word-level, then put the sentences having word-level coreference into one cluster. From then on, this events coreference problem is converted to be a normal coreference resolution task.

Since 61.58% articles in the training data have only one cluster, training a text classification model before the coreference resolution task is a possible improvement. Unfortunately, we don't have enough time to implement the coreference resolution model, so we are unsure if our ideas work.

Baselines	CEAF-e	CEAF-m
MaxC	38.53%	18.76%
MinC	56.41%	82.53%
MLP	53.76%	80.16%

Table 2: Subtask 3 baseline results

System	Task 4 (Acc)
bigbird-roberta-base, 3 epochs	28.34%
bigbird-roberta-base, 20 epochs	74.08%
bigbird-roberta-base + CRF, 20 epochs	78.56%

Table 3: Subtask 4 development results

4.3 Subtask 4

For subtask 4, we again fine-tune a bigbird checkpoint in NER-style fashion. We show development set results in Table 3. We first started with a RoBERTa-checkpoint, but found out that roughly 20% of the sequences are above 512 tokens, so we switched to bigbird to avoid sliding window approaches. We performed mostly experiments regarding the number of epochs, because training for too little doesn't yields poor performance. Additional improvements can be made by adding a CRF layer on top of the BigBird last hidden state, as kindly pointed out in the feedback of the progress report and motivated by (Lample et al., 2016).

We selected 25% train split to fine-tune number

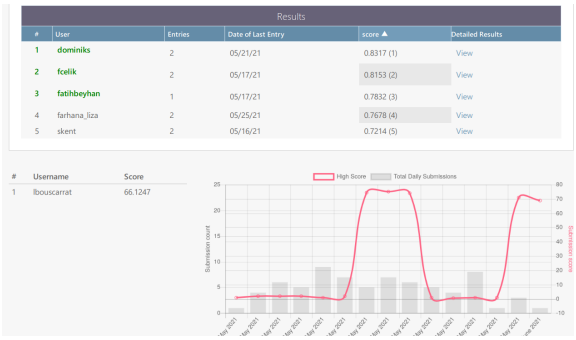

Figure 2: Subtask 1

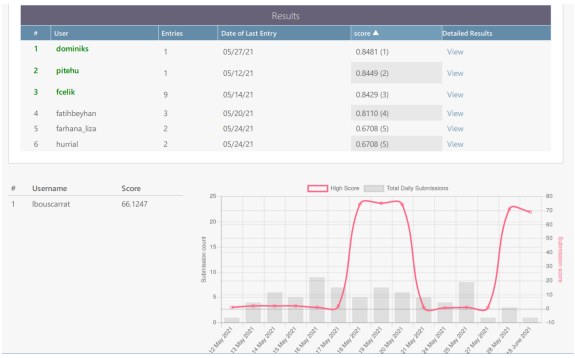

Figure 3: Subtask 2

of epochs and then trained a bigbird-roberta-large model using our best configuration (20 epochs) on all training data and used that to predict the test data. We submitted bigbird-large trained for 20 epochs and bigbird-large + CRF trained for 20 epochs, however bigbird-large without a CRF yielded slightly better results on the test set (68.8% vs 68.5%).

5 Test results

At the time of submission, our approaches yielded best results for subtask 1 and 2 and somewhat competitive results for subtask 4. We show the leaderboards in the following, our account for submitting is *dominiks*

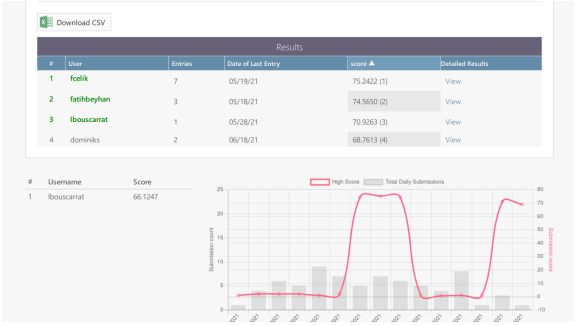

Figure 4: Subtask 4

6 Related Works

A similar study about extracting events from text has been covered in a shared task in 2019 (Hürriyetoğlu et al., 2020b), i.e. the competition was to solve Subtask 1,2 and 4, however on a smaller scale (3.5K training documents compared to the SCALE data with 9.3K English training documents). The winning system back then used GRUs to tackle Sbutask 1 and 2, achieving an average F1 of 0.69 (Hürriyetoğlu et al., 2020b).

The state-of-the-art in sequence classification has shifted a lot since then; These days, we use Transformer-based architectures pre-trained on language modeling tasks such as (Devlin et al., 2019; Liu et al., 2019), which are likely to work well for extracting event information.

For the event coreference task, many works approach it by training the embeddings to learn the implicit relations between coreferent events. (Krause et al., 2016) solve the problem by generating contextualized representations of event mention pairs. Another study in (Kenyon-Dean et al., 2018) also use word embeddings to embed the event mentions and their context, and model to maximize the cosine similarity between coreferent mentions.

A more recent work (Ahmed and Martin, 2021) shows the effectiveness of BERT for within-document coreference. It also uses the threshold of cosine similarity between the corresponding vectors of the mention pairs as the coreference criteria, but the representation is from a frozen BERT model in this study. Their best model can always achieve 100% true negatives by never clustering mentions, achieve 100% true positives by clustering all mentions in a single cluster, and achieve the state-of-the-art overall.

7 Conclusion

We applied Transformers to automatically extract Socio-political events from raw text and we conducted such experiments on the Case dataset. The dataset is part of a shared task and workshop at the ACL-IJCNLP 2021, but results and other participant's submission won't be public at the time of submission of these projects, so it is hard to put our work into perspective.

Nevertheless, our findings are: bigger models work better; and BigBird works better for document-level sequence classification compared to e.g. RoBERTa – while these are not necessarily groundbreaking from an NLP perspective, we

still managed to achieve the to-date best published results on the blind testset of the English version for 2 out of 4 subtasks (on sequence classification), competitive results on another subtask (token-level classification) and did not manage to beat a majority baseline in the last subtask (clustering).

References

Shafiuddin Rehan Ahmed and James H. Martin. 2021. Within-document event coreference with bert-based contextualized representations. *CoRR*, abs/2102.09600.

Jie Cai and Michael Strube. 2010. Evaluation metrics for end-to-end coreference resolution systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 28–36, Tokyo, Japan. 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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

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, Deniz Yüret, Osman Mutlu, Çağrı Yoltar, Fırat Duruşan, and Burak Gürel. 2020a. Cross-context news corpus for protest events related knowledge base construction.

Ali Hürriyetoğlu, Erdem Yörük, Deniz Yüret, Çağrı Yoltar, Burak Gürel, Fırat Duruşan, Osman Mutlu, and Arda Akdemir. 2020b. Overview of clef 2019 lab protestnews: Extracting protests from news in a cross-context setting.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 1–10, New Orleans, Louisiana. Association for Computational Linguistics.

Sebastian Krause, Feiyu Xu, Hans Uszkoreit, and Dirk Weissenborn. 2016. Event linking with sentential features from convolutional neural networks. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 239–249, Berlin, Germany. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

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.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Liyan Xu and Jinho D. Choi. 2020. Revealing the myth of higher-order inference in coreference resolution.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2021. Big bird: Transformers for longer sequences.