EventGraph at CASE 2021 Task 1: A General Graph-based Approach to Protest Event Extraction

Huiling You,[1] **David Samuel,**[1] **Samia Touileb,**[2] and **Lilja Øvrelid**[1]
[1]University of Oslo
[2]University of Bergen
{huiliny, davisamu, liljao}@ifi.uio.no
samia.touileb@uib.no

Abstract

This paper presents our submission to the 2022 edition of the CASE 2021 shared task 1, subtask 4. The EventGraph system adapts an end-to-end, graph-based semantic parser to the task of Protest Event Extraction and more specifically subtask 4 on event trigger and argument extraction. We experiment with various graphs, encoding the events as either "labeled-edge" or "node-centric" graphs. We show that the "node-centric" approach yields best results overall, performing well across the three languages of the task, namely English, Spanish, and Portuguese. EventGraph is ranked 3rd for English and Portuguese, and 4th for Spanish. Our code is available at: https://github.com/huiling-y/eventgraph_at_case

1 Introduction

The automated extraction of socio-political event information from text constitutes an important NLP task, with a number of application areas for social scientists, policy makers, etc. The task involves analysis at different levels of granularity: document-level, sentence-level, and the fine-grained extraction of event triggers and arguments within a sentence. The CASE 2022 Shared Task 1 on Multilingual Protest Event Detection extends the 2021 shared task (Hürriyetoğlu et al., 2021a) with additional data in the evaluation phase and features four subtasks: (i) document classification, (ii) sentence classification, (iii) event sentence coreference, and (iv) event extraction.

The task of event extraction involves the detection of explicit event triggers and corresponding arguments in text. Current classification-based approaches to the task typically model the task as a pipeline of classifiers (Ji and Grishman, 2008; Li et al., 2013; Liu et al., 2020; Du and Cardie, 2020; Li et al., 2020) or using joint modeling approaches (Yang and Mitchell, 2016; Nguyen et al., 2016; Liu et al., 2018; Wadden et al., 2019; Lin et al., 2020).

In this paper, we present the EventGraph system and its application to Task 1 Subtask 4 in the 2022 edition of the CASE 2021 shared task. EventGraph is a joint framework for event extraction, which encodes events as graphs and solves event extraction as semantic graph parsing. We show that it is beneficial to model the relation between event triggers and arguments and approach event extraction via structured prediction instead of sequence labelling. Our system performs well on the three languages, achieving competitive results and consistently ranked among the top four systems.

In the following, we briefly describe the data supplied by the shared task organizers and present Subtask 4 in some more detail. We then go on to present an overview of the EventGraph system focusing on the encoding of the data to semantic graphs and the model architecture. We experiment with several different graph encodings and provide a more detailed analysis of the results.

2 Data and task

Our contribution is to subtask 4, which falls under shared task 1 – the detection and extraction of socio-political and crisis events. While most subtasks of shared task 1 have sentence-level annotations, subtask 4 has been annotated at the token-level while providing the annotators the document-level contexts. Subtask 4 focuses on the extraction of event triggers and event arguments related to contentious politics and riots (Hürriyetoğlu et al., 2021a). This subtask has been previously approached as a sequence labeling problem combining various methods of fine-tuning pre-trained language models (Hürriyetoğlu et al., 2021a).

The data supplied for Subtask 4 is identical to that of the 2021 edition of the task, as presented in Hürriyetoğlu et al. (2021a). The data is part of the multilingual extension of the GLOCON dataset (Hürriyetoğlu et al., 2021b) with data from English, Portuguese, and Spanish. The source of the

Proceedings of the 5th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE), pages 155 - 160

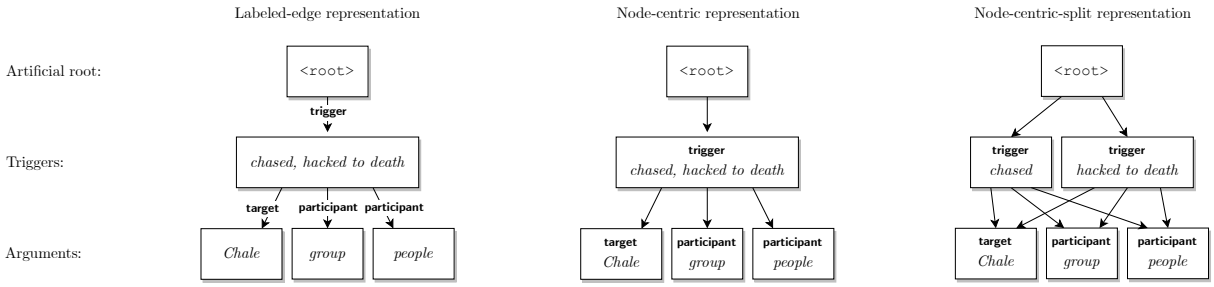

Figure 1: Graph representations of sentence "*Chale was allegedly chased by a group of about 30 people and was hacked to death with pangas, axes and spears.*"

data is protest event coverage in news articles from specific countries: China and South Africa (English), Brazil (Portuguese), and Argentina (Spanish). The data has been doubly annotated by graduate students in political science with token-level information regarding event triggers and arguments. Hürriyetoğlu et al. (2021a) reports the token level inter-annotator agreement to be between 0.35 and 0.60. Disagreements between annotators were subsequently resolved by an annotation supervisor. Table 1 shows the number of news articles for each of the languages in the task, distributed over the training and test sets. This clearly shows that the majority of the data is in English with only a fraction of articles in Portuguese and Spanish.

Relevant statistics for the different event component annotations for Subtask 4 are presented in Table 1 detailing the number of triggers, participants, and various other types of argument components, such as place, target, organizer, etc. Once again, the table also illustrates the comparative imbalance in data across the three languages.

	English	Portuguese	Spanish
train	732 (2,925)	29 (78)	29 (91)
dev	76 (323)	4 (9)	1 (15)
test	179 (311)	50 (190)	50 (192)
trigger	4,595	122	157
participant	2,663	73	88
place	1,570	61	15
target	1,470	32	64
organizer	1,261	19	25
etime	1,209	41	40
fname	1,201	48	49

Table 1: **Top**: Number of articles (sentences) for the different languages in Subtask 4 (Hürriyetoğlu et al., 2021a). About 10 percent (in terms of sentences) of the official training data is used as the development split. **Bottom**: Counts for the different event components in Subtask 4 training data for English, Portuguese, and Spanish (Hürriyetoğlu et al., 2021a).

3 System overview

We use our system, EventGraph, that adapts an end-to-end graph-based semantic parser to solve the task of extracting socio-political events. In what follows, we give more details about the graph representation and the model architecture of our system.

3.1 Graph representations

We represent each sentence as an event graph, which contains event trigger(s) and arguments as nodes. In an event graph, edges are constrained between the trigger(s) and the corresponding arguments. However, since our system can take as input graphs in a general sense the precise graph representation that works best for this task must be determined empirically. We here explore two different graph encoding methods, where the labels for triggers and arguments are represented either as edge labels or node labels, namely "labeled-edge" and "node-centric". Since sentences in the data may contain information about several events with arguments shared across these, we also experiment with a version of the "node-centric" approach where multiple triggers give rise to separate nodes in the graph. The intuition behind this is that it is easier for the model to predict a node anchoring to a single span than to several disjoint spans.

- **Labeled-edge**: labels for event trigger(s) and arguments are represented as edge labels; multiple triggers are merged into one node, as shown by the first graph of Figure 1.
- **Node-centric**: labels for event trigger(s) and arguments are represented as node labels;

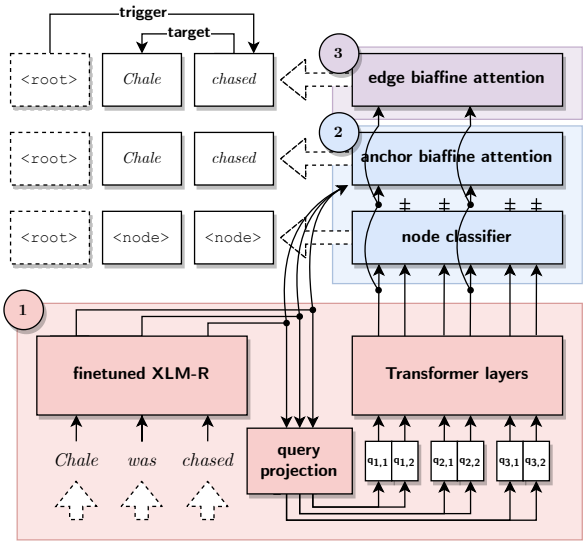

Figure 2: EventGraph architecture. 1) the input gets a contextualized representation from **the sentence encoding** module, 2) graph nodes are decoded by **the node prediction** module and 3) connected by **the edge prediction** module. The given example is for "label-edge" event graph parsing.

there is always a single node for trigger(s), as shown by the second graph of Figure 1.

- **Node-centric-split**: node labels denote trigger(s) and argument roles; multiple triggers are represented in different nodes, as shown by the third graph of Figure 1.

3.2 Model architecture

Our model is built upon a winning framework (Samuel and Straka, 2020) from a previous meaning representation parsing shared task (Oepen et al., 2020). The model contains customizable components for predicting nodes and edges, thus generating event graphs for different graph representations. We introduce each component of the model as following (Figure 2):

Sentence encoding Each token of an input sentence obtains a contextualized embedding from a pretrained language model, the large version of XLM-R (Conneau et al., 2020) in our implementation. These embeddings are mapped onto latent queries by a linear transformation layer, and processed by a stack of Transformer layers (Vaswani et al., 2017) to model the dependencies between queries.

Node prediction A node-presence classifier processes the queries and predicts nodes by classifying each query. An anchor biaffine classifier (Dozat and Manning, 2017) creates anchors from the nodes to surface strings via deep biaffine attention between the queries and the contextual embeddings.

Edge prediction With predicted nodes, two biaffine classifiers are used to construct the edges between nodes: one classifier predicts the presence of edge between a pair of nodes and the other predicts the corresponding edge label.

The graph generated for each input sentence contains the extracted event components. We then convert the labels to BIO format.

4 Experimental setup

Data We use all the official training data to train our final model, without using any additional data. During development time, we set aside about 10 percent of the training data for development. A breakdown of the number of articles and sentences in train and dev are provided in Table 1.

Joint training We train our model on the training data of all three languages and test on the official test data. As shown in Table 1, the training data for Portuguese and Spanish makes only a small portion of all training data, which leads to few-shot learning for these two languages.

Implementation details We use the large version of XLM-R via HuggingFace `transformers` library (Wolf et al., 2020). All models were trained with a single Nvidia RTX3090 GPU.

Evaluation metrics The evaluation metric is a macro F_1 score for individual languages. The predicted event-annotated texts are in BIO format, and the scores are calculated with a python implementation[1] of the `conlleval` evaluation script used in CoNLL-2000 Shared Task (Tjong Kim Sang and Buchholz, 2000), where precision, recall and F_1 scores are calculated for predicted spans against the gold spans and there is no dependency between event arguments and triggers.

Submitted systems We submitted three models as listed in Table 3.

5 Results and discussion

We summarize the results of our systems on the official test data in Table 3. All scores are obtained by submitting our test predictions to the shared

[1] https://github.com/sighsmile/conlleval

Language	System	trigger	target	Place	Participant	Organizer	fname	etime	all
		457	*134*	*118*	*293*	*131*	*129*	*121*	
En	Label-edge	82.48	56.29	75.44	74.62	74.52	50.42	77.06	73.46
	Node-centric	84.21	62.09	74.89	76.42	75.46	54.31	81.22	75.85
	Node-centric-split	84.62	52.88	75.11	73.75	74.91	52.28	78.97	73.92
		28	*5*	*5*	*7*	*4*	*7*	*5*	
Es	Label-edge	66.67	60.00	100.00	100.00	66.67	71.43	80.00	73.85
	Node-centric	65.62	72.73	100.00	100.00	80.00	76.92	80.00	75.76
	Node-centric-split	71.19	54.55	100.00	100.00	66.67	85.71	60	75.59
		11	*7*	*3*	*5*	*2*	*2*	*5*	
Pr	Labeled-edge	83.33	71.43	75.00	90.91	66.67	100.00	66.67	78.87
	Node-centric	88.00	61.54	66.67	90.91	100.00	100.00	66.67	79.45
	Node-centric-split	91.67	71.43	50	90.91	100.00	66.67	100.00	83.78

Table 2: Detailed F_1 scores of our systems on the development data with different graph representations. We also add the number of each event component to better compare the distribution of components against the scores.

System	Language	Macro F_1
Labeled-edge	English	73.12
	Spanish	64.02
	Portuguese	69.62
Node-centric	English	74.02
	Spanish	64.16
	Portuguese	70.73
Node-centric-split	English	74.76_3
	Spanish	64.49_4
	Portuguese	71.72_3
Winning systems	English	77.46_1
	Spanish	69.87_1
	Portuguese	74.57_1

Table 3: Results of our systems on the official test data with different graph representations. We also include the winning system results from the shared task leaderboard. Subscripts indicate the ranking on the leaderboard, so we only add corresponding ranking to our best-performing system.

task.[2] Results show that "node-centric" systems generate better results than "label-edge" systems, and it is more beneficial to keep multiple event triggers as separate nodes. In terms of languages, all models perform best on English, which is unsurprising, since the training data consists mostly of English. However, the results on Portuguese are consistently better than those of Spanish, signaling English might be a better transfer language for Portuguese than for Spanish.

Compared with other participating systems, in particular the winning systems,[2] as shown in Table 3, our results are still competitive. We rank 3rd for English and Portuguese, and 4th for Spanish; our best results are achieved by a single system. For English and Portuguese, our results are very close to the winning results, which are achieved by different participating systems.

[2] https://codalab.lisn.upsaclay.fr/competitions/7126, accessed on September 29, 2022.

Argument	System	P	R	$F1$
fname	Labeled-edge	47.62	53.57	50.42
	Node-centric	52.50	56.25	54.31
	Node-centric-split	48.84	56.25	52.28
target	Labeled-edge	60.28	52.80	56.29
	Node-centric	65.52	59.01	62.09
	Node-centric-split	58.21	48.45	52.88

Table 4: Detailed Precision, Recall, and $F1$ scores of `fname` and `target` arguments for English developmentset.

5.1 Error analysis on development data

Since the gold data for the test set is not available to task participants, we are not able to perform more detailed error analysis. Hence, to have more insights into our models' performance, we provide some error analysis on the development data (as described in Table 1). As previously mentioned, during our model development phase, we did not use all the official training data for training, but set aside small set for validation (about 10%).

As shown in Table 2, over all event components, `target` and `fname` arguments are more difficult to extract than others, with the scores substantially lower across different languages and models. In general, our models perform best in `trigger` extraction, partly because the number of triggers is much larger than event arguments for all datasets.

We further look at `target` and `fname` prediction scores of the English development set. As shown in Table 4, for `fname`, our systems tend to over-predict, with consistently lower precision scores; by manually going through our systems' predictions, we find many labeled chunks of `fname` are actually non-event components. For `target`, our systems tend to under-predict, with consistently higher precision scores; we also find that our systems would predict a longer span, for instance "former diplomat" as opposed to "diplomat", which is the gold span, and sometimes our systems confuse `organizer` and `participant` with `target`, by wrongly labelling the corresponding span as `target`.

6 Conclusion

In this paper we have presented the EventGraph system for event extraction and its application to the CASE 2022 shared task on Multilingual Protest Event Detection. EventGraph solves the task as a graph parsing problem hence we experiment with different ways of encoding the event data as general graphs, contrasting a so-called "labeled-edge" and "node-centric" approach. Our results indicate that the "node-centric" approach is beneficial for this task and furthermore that the separation in the graph of nodes belonging to different events in the same sentence proves useful. A more detailed analysis of the development results indicates that our system performs well in trigger identification, however struggles in the identification of `target` and `fname` arguments.

Acknowledgments

This research was supported by industry partners and the Research Council of Norway with funding to *MediaFutures: Research Centre for Responsible Media Technology and Innovation*, through the Centres for Research-based Innovation scheme, project number 309339.

References

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. 3

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*. 3

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics. 1

Ali Hürriyetoğlu, Osman Mutlu, Erdem Yörük, Farhana Ferdousi Liza, 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)*, pages 79–91, Online. Association for Computational Linguistics. 1, 2

Ali Hürriyetoğlu, Erdem Yörük, Osman Mutlu, Fırat Duruşan, Çağrı Yoltar, Deniz Yüret, and Burak Gürel. 2021b. Cross-context news corpus for protest event-related knowledge base construction. *Data Intelligence*, 3(2):308–335. 1

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics. 1

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 829–838, Online. Association for Computational Linguistics. 1

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, Sofia, Bulgaria. Association for Computational Linguistics. 1

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics. 1

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics. 1

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural*

Language Processing, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics. 1

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, San Diego, California. Association for Computational Linguistics. 1

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The second shared task on cross-framework and cross-lingual meaning representation parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1–22, Online. Association for Computational Linguistics. 3

David Samuel and Milan Straka. 2020. ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53–64, Online. Association for Computational Linguistics. 3

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task chunking. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*. 3

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30. 3

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. 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 5784–5789, Hong Kong, China. Association for Computational Linguistics. 1

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. 3

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics. 1