FKIE_itf_2021 at CASE 2021 Task 1: Using Small Densely Fully Connected Neural Nets for Event Detection and Clustering

Nils Becker
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg
nils.becker@
fkie.fraunhofer.de

Theresa Krumbiegel
Fraunhofer FKIE
Fraunhoferstraße 20
53343 Wachtberg
theresa.krumbiegel@
fkie.fraunhofer.de

Abstract

In this paper we present multiple approaches for event detection on document and sentence level, as well as a technique for event sentence co-reference resolution. The advantage of our co-reference resolution approach, which handles the task as a clustering problem, is that we use a single neural net to solve the task, which stands in contrast to other clustering algorithms that often are build on more complex models. This means that we can set our focus on the optimization of a *single* neural network instead of having to optimize numerous different parameters. We use small densely connected neural networks and pretrained multilingual transformer embeddings in all subtasks. We use either document or sentence embeddings, depending on the task, and refrain from using word embeddings, so that the implementation of complicated network structures and unfolding of RNNs, which can deal with input of different sizes, is not necessary. We achieved an average macro F1 of 0.65 in subtask 1 (i.e., document level classification), and a macro F1 of 0.70 in subtask 2 (i.e., sentence level classification). For the co-reference resolution subtask, we achieved an average CoNLL-2012 score across all languages of 0.83.

1 Introduction

Gathering information about current and past events is quite important since such information can help to detect, analyze, prevent and forecast dangerous social and political situations. An accumulation of protest events in a certain region may indicate massive discrepancies between two or more parties. Such situations can escalate and result in violence. Using modern systems and data including for example news articles, violent events can be forecast (Schrodt et al., 2013). Today, caused by a globally connected world, there exists an endless stream of news and information. To conquer this flood of data, much human effort is needed. Therefore, automation of information analysis can help to reduce the workload.

One task in this area is the detection of events in texts consisting of natural language, for example newspaper articles. It is an easy task for humans to read, understand and identify such events. For computers it is more difficult to process natural language and detect event mentions.

In this paper, we present our approaches for event detection in articles and sentences based on simple densely connected neural networks as part of task 1 (Hürriyetoğlu et al., 2021) of the Shared Task on Socio-Political and Crisis Events Detection at CASE @ ACL-IJCNLP 2021. The first task is split up into four different subtasks. We participated in the first three.

For the first subtask, we used an accumulation of trained neural nets with majority voting, where each net is a densely connected net consisting of only six layers including the in- and output layer. For the second subtask, we used a single net with the same specifications as in the first subtask. The third subtask aims at co-reference resolution of event sentences. We see this subtask as a typical clustering task. Therefore, we use a comparison based algorithm, which reduces the clustering problem mainly to the optimization of a single neural net. Co-reference resolution in our case, is based on the comparison of sentence pairs and will be described later in more detail.

All code used in this paper is publicly available [1].

The paper will proceed as follows: First, related work will be introduced. After that, the subtasks the we participated in will be described. The next chapter presents our methodology, including data

[1] https://github.com/s6nlbeck/FKIE_itf_Task1.git

Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE) , pages 113–119

preparation and system descriptions for all subtasks. Then, the results are depicted. In the end, we come to a conclusion and give an outlook for future work.

2 Related Work

Since this workshop is a follow up event of the CLEF ProtestNews 2019 and AESPEN at LREC 2020 Shared Task, many approaches were already made as mentioned by Hürriyetoğlu et al. (2019) and Hürriyetoğlu et al. (2020). Aside from these approaches, a variety of other experiments trying to solve the task of event detection can be found in the literature. In earlier years, pattern matching approaches as described by Riloff et al. (1993) were common and successful for the detection of events, but often required much human effort and domain knowledge for pattern construction. This lead to the idea propagated by Riloff and Shoen (1995) of the automatic construction of such patterns. With the rise of available and affordable computing power, these techniques were replaced by modern machine learning techniques and especially artificial neural networks. State of the art systems for event detection, see for example Cui et al. (2020), use a combination of different kinds of neural nets, like bidirectional LSTMs and modified graph convolutional networks. Other models, as presented by Nguyen and Grishman (2015), use convolutional neural networks and reduce the task to a multi class labeling problem. Event detection can also be seen as a question answering task, where one could ask if an event exists in the given text or not, as done by Liu et al. (2020).

What all of the systems have in common is that they need a representation of text that is understandable for a computer. Piskorski et al. (2020) showed that modern transformer embeddings are the best choice by comparing them to classic word embeddings and achieving superior results with them. Based on these findings, we decided to make use of them in our work too.

For subtask 3, common clustering algorithms could be used for co-reference resolution, when using suitable metrics. Co-reference resolution using mention pair models, such as those proposed by Ng (2010), Örs et al. (2020) and Radford (2020), could also be implemented.

3 Task Description

The first task of the workshop consists of four different subtasks. The different subtasks build upon each other, starting at document level (subtask 1) and go on to gradually focus on smaller instances (sentence level, word level). We provide three different models for the first three subtasks. The data for all three subtasks is provided in a JSON format.

3.1 Subtask 1

In the first subtask, the challenge is to identify if a news article contains a past or ongoing event. For training, data in three different languages, namely English, Spanish and Portuguese, was provided. Each training sample consists of an unique identifier, a news article as the text basis and a binary label which marks if the article contains an event or not. Label 0 means that no event is included, label 1 means that an event is present. In total, the dataset comprises 11811 entries and is described in detail in table 1.

	en	es	pr	total
1	1912	131	197	2240
0	7412	869	1290	9571
total	9324	1000	1487	11811
prop. 1	20.5%	13.1%	13.2%	19%

Table 1: Details of training data for subtask 1

A training instance of subtask 1 looks as follows:


```
{"id":100023,"text":"2 policemen
   suspended for torturing man\
   nHYDERABAD:The Ranga Reddy
   superintendent of police on
   Monday suspended a head
   constable and a constable for
   adopting 'heinous' methods in
   interrogating Jangaiah, an
   accused in a missing person
   case.\nTNN | Sep 3, 2001, 02.0
   8 AM IST\nhyderabad:the ranga
   reddy superintendent ","label"
   :0}
```


3.2 Subtask 2

The second subtask is quite similar to the first one, the only difference being that the event detection has to be done at sentence level. Thus, the goal is to decide for each sentence if it contains an event or not. Each entry in the training corpus contains a single sentence instead of a whole news article. The dataset is much larger than the set for subtask 1, containing 26748 instances, as shown in table 2.

	en	es	pr	total
1	4223	450	281	4954
0	18602	2291	901	21794
total	22825	2741	1182	26748
prop. 1	18.5%	16.4%	23.8%	18.5 %

Table 2: Details of training for subtask 2

In the following an example of the training data of subtask 2 is given:


```
{"id":66133,"label":0,"sentence":
   "He had also made headlines
   for kidnapping his 13-year-old
    brother and taking him to
   Syria."}
```


3.3 Subtask 3

The third subtask differs from both of the other subtasks. It aims at event sentence co-reference resolution. This means that it has to be decided which sentences are about the same event. In this case, co-reference resolution can be seen as a clustering task. Each example in the training data consist of an unique identifier, multiple sentences and their respective event cluster. An overview of the data distribution for subtask three is given in table 3.

	en	es	pr	total
instances	596	11	21	628

Table 3: Details of training data for subtask 3

An example of a shortened training instance is given below. Each instance has four fields. One field contains an array including the event sentences. The depicted example has a total of four sentences. Each sentence is further represented as a number. For example, the sentence beginning with "Around 30,000..." is represented by the number 4. The event clusters are given as arrays. Each array contains the numbers of the sentences of the respective cluster. We can see that in the given example, sentence 15 is a cluster by itself and the other three sentences, sentences 4, 5 and 11, build another cluster. The last field is the id field, which contains an unique identifier for the entry.


```
{"event_clusters":[[15],[4,5,11]]
   , "sentence_no":[4,5,11,15], "
   sentences":["Around 30,000..."
   ,"Several...", "RFEA chief
   ...","On Tuesday..."], "id":55
   666}
```


4 Methodology

4.1 Subtask 1 and 2 - Data Preparation

For our experiments for subtasks 1 and 2, we use the Flair framework (Akbik et al., 2019). The utilised document embeddings are generated using the pre-trained multilingual cased Bert model. The Bert model uses bidirectional LSTMs to create context sensitive embeddings (Devlin et al., 2019). Each embedding is represented by a 768-dimensional vector. We use the Bert model to generate the embeddings without any text preprocessing. For the first subtask, each news article is transformed into one vector, whereas in the second subtask every sentence is transformed into a same sized sentence embedding.

4.2 Subtask 1 and 2 - System Description

For the first subtask we use an accumulation of one hundred separately trained densely connected neural nets with one input layer of size 768, four hidden layers with 64 neurons and one output layer with one single unit. Each net is trained for 20 epochs with the adam optimizer and a learning rate of 0.001. As an activation function, we use the sigmoid function for each neuron. Since we are dealing with a binary classification task, we use binary crossentropy as a loss function. After each epoch the training data is shuffled. For the first subtask, a majority vote is used to decide if the article contains an event or not.

During the development phase, we also tested different structures of CNNs using the data from the shared tasks of 2019. The best result was gained with a small densely residual network like structure, as proposed by Huang et al. (2017), with a macro F1 score of 0.77. On the same data, our final approach reached a score of 0.81.

For the second subtask we use a single net with the same specifications as described for subtask 1.

4.3 Subtask 3 - Data preparation

The main part of our approach for subtask 3 is based on a neural network which is able to compare two sentences and determine if they belong to the same event cluster.

For each entry in the dataset a number of training instances are generated. A training instance is a

triple which includes the sentence embeddings of two different sentences and a binary label which shows if the two sentences belong to the same event cluster or not.

This means that for every instance of the dataset, first the needed embeddings are calculated in the same way as in the subtasks before. After that, the positive and negative sentence pairs are generated and the matching labels are added. The sentences in the negative sentence pairs do not belong to the same event cluster, the ones in the positive pairs do. This results in a set of triples containing all possible combinations of sentences with corresponding labels. The generated entries for each instance are merged into one big dataset.

4.4 Subtask 3 - System Description

Since the third subtask differs substantially from the other subtasks, we developed and used another model compared to subtasks 1 and 2. As modern sentence embeddings based on neural nets are quite powerful, we also considered to use neural nets for clustering. As mentioned in section 2, many different clustering algorithms are available. In the area of using neural networks for clustering, self organizing maps (Kohonen, 1990) and neural gases (Martinetz et al., 1993) can be considered. Neural techniques like these are mainly used for representing topological structures in the given data. To use them for clustering, time-consuming additional steps would be needed beforehand.

Popular clustering algorithms like DBSCAN (Ester et al., 1996) include numerous hyperparameters which have to be optimized before the models can be used sensibly. Additionally, for some models the amount of clusters must be specified in advance. An example for this is the k-Means algorithm (Hamerly and Elkan, 2004). This makes them unsuitable for our use case.

We argue that it would be desirable if one did not have to define a fixed amount of cluster or to optimize many different hyperparameters before using the model.

In the following we present a supervised clustering algorithm based on a neural network. This neural network needs to be trained in advance. The task of the trained net is to decide if two sentences belong to the same cluster or not. Our approach reduces the amount of work that has to be invested before using the model, as only the neural net needs to be optimized.

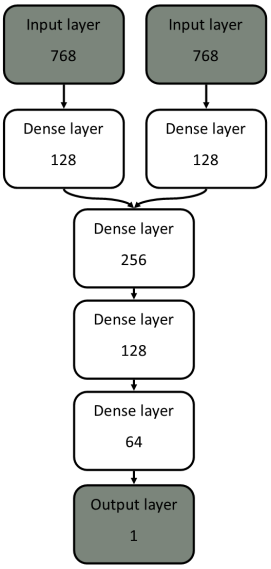

Figure 1: Structure of the used neural net.

The comparison of the event sentences is done by a neural network with two inputs and one output. Using the prepared data triples that were just mentioned, the net can be trained and optimized in a regular manner. The goal is to decide correctly for two sentences if they belong to the same event cluster. If this succeeds for all sentence pairs, we can in theory build perfect event clusters. The output generated by the neural net is needed for the final clustering which is implemented by using a graph.

The used neural network consists of two input layers with 768 neurons. To reduce the input size after both input layers, a layer of 128 neurons is used. To connect both size reduced inputs to each other, a 256 sized layer is used, followed by a 64 sized layer and an output layer with a single neuron like pictured in figure 1.

In total, the model has 238,081 trainable parameters, including the bias weights. Like in subtask 1 and 2 we use the same optimizer, loss and activation function and learning rate.

As mentioned before, the trained neural network is used as a comparison function, which determines if two sentences belong to the same event cluster or not. We use the results ot this comparison for building a graph $G = (V, E)$. The graph consists of a set of nodes $V = v_1, ..., v_n$ and a set of edges $E = \{\{v_x, v_y\} \mid v_x, v_y \in V \text{ and } v_x \neq v_y\}$. The sentences are represented by the nodes. If the network predicts that the two sentences belong to the same cluster, an edge is added in the graph between the corresponding nodes, otherwise no edge

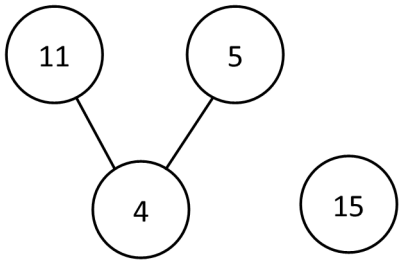

Figure 2: Example of a possible generated graph

is added. The resulting graph is analyzed with regard to disjoint subgraphs. Each individual subgraph represents an event cluster. Figure 2 shows a possible graph with two distinct clusters.

5 Results

5.1 Subtask 1

For both of the first subtasks, the macro F1 score is used for evaluation on the provided test set. In the first subtask we achieved a macro F1 score of 0.74 on the English documents, 0.68 on the Spanish documents, 0.62 on the Portuguese ones and 0.54 on the Hindi documents. Averaged over all test data, a score of 0.65 was achieved, which is slightly better in comparison to the results of a single net. Mostly, the use of multiple nets leads to a small increase in performance as can be seen in table 4. Only with regard to the Spanish data, the single net performed slightly better than the combination of multiple nets. However, this may be an outlier and requires further analysis.

	en	**es**	**pr**	**hi**	**avg**
100 nets	0.74	0.68	0.62	0.54	0.65
single net	0.72	0.70	0.60	0.50	0.63

Table 4: Result for subtask 1 using different amount of nets

We compare these results to the results that were achieved during development of the systems. For the preliminary evaluation we used 20 percent of the training set as a test set. The evaluation results for subtask 1 are shown in table 5. We reached a macro F1 score of 0.76 for English, 0.66 for Spanish and 0.68 for Portuguese. This lead to an average over all languages of 0.70.

We see that the results achieved on the self-compiled test set are similar to the ones achieved on the test set of the organizers. Only Portuguese stand out with a difference in performance of 0.06.

	en	**es**	**pr**	**avg**
macro F1	0.76	0.66	0.68	0.70

Table 5: Preliminary results for subtask 1

5.2 Subtask 2

Since the improvement using an accumulation of neural nets is only marginal for the classification at sentence level, we used a single net for the second subtask. We scored a macro F1 of 0.65 on the English data, 0.76 on the Spanish data and 0.70 on the Portuguese data, as specified in table 6.

	en	**es**	**pr**	**avg**
macro F1	0.65	0.76	0.70	0.70

Table 6: Result for subtask 2 on different languages

Considering the results of the first two subtasks, which both use very similarly constructed models, it is noticeable that in subtask 1 the best results are achieved on the English data, while in subtask 2 English constitutes the worst performing language class.

For subtask 2, a similar constructed test set as in subtask 1 was used during development. On this set we achieved an average score of 0.73 over all languages. Details for the different languages can be found in table 7. The results are slightly better than the ones for subtask 1.

	en	**es**	**pr**	**avg**
macro F1	0.78	0.73	0.68	0.73

Table 7: Preliminary results for subtask 2

Moreover, we find that the performance of our system declines notably with regard to English when using the test set provided by the organizers. Further analysis is needed to determine what causes this.

5.3 Subtask 3

For evaluating the system submitted for subtask 3, the CoNLL-2012 average score was used. The scores were calculated for each language separately. The amount of test data is quite low, as shown in table 8, the systems were tested on only 180 examples in total.

On the English data we achieved a score of 0.77 and on the Spanish data a score of 0.83. The best

	en	es	pr	total
instances	100	40	40	180

Table 8: Distribution of classes in test data for subtask 3

result with a score of 0.91 was reached on the Portuguese dataset. An overview is given in table 9.

	en	es	pr	avg
CoNLL-2012 avg	0.77	0.83	0.91	0.83

Table 9: Results for subtask 3 for different languages

During the development and testing phase using the training set, the overall score averaged over all three languages was 0.82. The basis for this result was a self compiled test set including 20 percent of the examples of each language included in the training set. The relatively good score for Portuguese on the final test set stands out, since very few data for training was available for this language. An analysis of the training and test data could be helpful to see if there are differences that cause this behaviour.

6 Conclusion

We presented three different approaches for the three different subtasks. The accumulation of several neural nets used in subtask 1 improved the results of the model just very slightly in comparison to a single densely connected neural net.

In general, we can see that working on word level is not mandatory. Sentence and document embeddings in combination with simple dense nets can lead to good results. This decreases the complexity of the task immensely. The results on the sentence level improve in comparison to the ones achieved on the document level, with exception of the results for the English data. The clear difference between the results obtained on the self-compiled test set and the test set of the organizers with regard to English serves as a good starting point for future work.

For subtask 3, we presented a simple solution for event sentence co-reference resolution, focusing on the optimization of a function for comparison by using a multi input neural network. Using this approach, we were able to solve the task in a way that does not require metrics, thresholds and other hyperparameters, which are often needed in clustering, and thus save time during the clustering process. For future work it would be interesting to use bidirectional LSTMs and other techniques to improve the results for co-reference resolution further.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2329–2339.

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.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231.

Greg Hamerly and Charles Elkan. 2004. Learning the k in k-means. *Advances in neural information processing systems*, 16:281–288.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

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).

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 *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 425–432. Springer.

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).

Teuvo Kohonen. 1990. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

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.

Thomas M Martinetz, Stanislav G Berkovich, and Klaus J Schulten. 1993. 'neural-gas' network for vector quantization and its application to time-series prediction. *IEEE transactions on neural networks*, 4(4):558–569.

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.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with 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 2: Short Papers)*, pages 365–371.

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.

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

Benjamin J Radford. 2020. Seeing the forest and the trees: Detection and cross-document coreference resolution of militarized interstate disputes. *arXiv preprint arXiv:2005.02966*.

Ellen Riloff and Jay Shoen. 1995. Automatically acquiring conceptual patterns without an annotated corpus. In *Third Workshop on Very Large Corpora*.

Ellen Riloff et al. 1993. Automatically constructing a dictionary for information extraction tasks. In *AAAI*, volume 1, pages 2–1. Citeseer.

Philip A Schrodt, James Yonamine, and Benjamin E Bagozzi. 2013. Data-based computational approaches to forecasting political violence. In *Handbook of computational approaches to counterterrorism*, pages 129–162. Springer.