Proceedings of AESPEN 2020, pages 63–68
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Event Clustering within News Articles

Faik Kerem Örs*, Süveyda Yeniterzi, Reyyan Yeniterzi***
*Sabancı University, **YAZI Information Technologies
İstanbul, Turkey
{fkerem, reyyan}@sabanciuniv.edu

Abstract

This paper summarizes our group's efforts in the event sentence coreference identification shared task, which is organized as part of the Automated Extraction of Socio-Political Events from News (AESPEN) Workshop. Our main approach consists of three steps. We initially use a transformer based model to predict whether a pair of sentences refer to the same event or not. Later, we use these predictions as the initial scores and recalculate the pair scores by considering the relation of sentences in a pair with respect to other sentences. As the last step, final scores between these sentences are used to construct the clusters, starting with the pairs with the highest scores. Our proposed approach outperforms the baseline approach across all evaluation metrics.

Keywords: Clustering, Coreference Resolution

1. Introduction

In news articles, an event can be described together with some reference to prior events or some other relevant events in order to give more background information to the reader. Therefore, news articles do not solely consist of one event throughout the article. Event sentence coreference identification (ESCI) task aims to group event containing sentences within a news article into clusters based on the event they contain. Sentences that refer to the same event belong to the same cluster while sentences that are about different events are grouped into different clusters. A good clustering of these events can improve other event related tasks like event extraction, event timeline extraction or cause and effect relation of events.

ESCI task is very similar to other coreference resolution tasks. In the entity coreference resolution task, the goal is to identify entity mentions that refer to the same entity. There is also the event coreference resolution task in which the idea is to determine which event mentions refer to the same event (Lu and Ng, 2018). Similarly, in ESCI task, the goal is to identify sentences that refer to the same event. In this particular task, the sentence as a whole is considered as an event mention.

As a result of this similarity, our proposed approach is also similar to a well-known approach in coreference resolution tasks, known as the Mention-Pair model (Ng, 2010). In this model, a binary classification model is used to classify pair of mentions as referring to either the same entity or not. After this prediction step, the pairwise prediction decisions are used to determine the coreference relations by clustering them (Ng, 2010). In our proposed approach, in addition to the prediction and clustering steps, we also use an intermediate step to re-score the pairs in order to reward consistencies and penalize inconsistencies among them.

Our proposed approach consists of three steps. We initially predict whether any given two sentences are coreferent or not. For this binary classification part, we adapt a pre-trained transformer-based neural network and fine-tune it for our task. After retrieving the predictions, we analyze how the pair of sentences interact with other sentences outside this pair. If there is an agreement on predictions, we add a reward to the score of the pair. If there is a disagreement, we decrease the score. Finally, we use a greedy approach for the clustering of sentences using their scores. Starting with the sentence pairs with the maximum scores, we construct clusters by combining more likely pairs and iterate until some stopping conditions are satisfied.

The rest of the paper is organized as following: Section 2 describes the data and the preprocessing steps, Section 3 details the proposed approach. Section 4 presents the experimental results and finally Section 5 concludes the paper with future work.

2. Data

The provided data is a subset of the data created for extracting protests from news in a cross-context setting (Hürriyetoğlu et al., 2019). The data was collected from online local English news articles from India and the news articles are about protest related events. 404 news articles, with their gold-standard labels, were provided as the training data and another 100 news articles, without any labels, are provided closer to the submission deadline for test purposes.

The data is provided in JSON format. It does not contain the whole news article, but only the sentences which contain an event. An example is provided below:

```
{"url": "http://www.newindianexpress.
   com/states/odisha/2011/apr/10/
   maoist-banners-found-243277",
 "sentences": [
   "Maoist banners found 10th April
      2011 05:14 AM KORAPUT : MAOIST
      banners were found near the
      District Primary Education
      Project (DPEP) office today in
      which the ultras threatened to
      kill Shikhya Sahayak candidates,
       outsiders to the district, who
      have been selected to join the
      service here.",
    "Maoists, in the banners, have also
        demanded release of hardcore
        cadre Ghasi who was arrested by
        police earlier this week.",
    "Similar banners were also found
        between Sunki and Ampavalli
        where Maoists also blocked road
        by felling trees."
  ],
  "sentence_no": [1, 2, 3],
  "event_clusters": [[1, 2], [3]]
}
```

As seen above, the input to the task is the sentences with provided sentence numbers, and the output is the event clusters using these sentence numbers.

2.1. Pre-processing

Data is provided in processed format, as sentences were already segmented and ready to be tokenized. After some analysis, it has been observed that in some cases, the title of the news article together with some newspaper metadata and timestamp is concatenated to the first sentence of the news article. For example, in the above example the "*Maoist banners found*" is the title which is followed by "*10th April 2011 05:14 AM KORAPUT :*". These are followed the by first sentence of the news article.

As a pre-processing step, several regular expressions are used to clean such noise from the data. After removing the title, metadata and timestamp, the remaining part has been considered as the first sentence.

3. Approach

Our proposed approach consists of three steps. In the first step, we simplify the problem by focusing on any given two sentences and predict whether they refer to the same event or not. In the next step, we use our prediction outputs (either -1 or 1) as scores and update them by analyzing not only the sentences in pairs but also their interactions with other event containing sentences in the news article. Finally, we use these scores in a greedy approach to construct the event clusters.

3.1. Same Event Prediction

In this task, all event containing sentences in a news article are grouped into pairs. Given these sentence pairs as input, the task is to predict whether these sentences refer to the same event or not. In this binary classification task, we initially convert the provided training data of news articles into sentence pairs. For the example given above, 3 sentence pairs are constructed with following labels as shown in Table 1.

As seen in Table 1, each event-containing sentence in the news article is pairwise grouped with all the rest of the event containing sentences in the news article. We specifically use the sentence numbers while creating the pairs, and use the sentence with the lower indices as the first sentence, and the one with the higher indices as the second. Therefore, for a news article with n sentences, we end up with $\frac{n(n-1)}{2}$ sentence pairs.

Pair No	First Sent. No	Second Sent. No	Label
1	1	2	TRUE
2	1	3	FALSE
3	2	3	FALSE

Table 1: Sentence Pairs and Labels for a News Article (TRUE for prediction 1 (refer to the same event) and FALSE for prediction 0 (refer to different events))

In the provided training data, on average each news article has around 4.5 sentences which contain an event. Overall, for the given 404 training instances, we end up with 4834 pairs of sentences in total. For this prediction part, we explore the pre-trained transformer-based neural network architectures. We fine-tune the following pre-trained models for our binary classification task.

- BERT (Devlin et al., 2018): Uses bidirectional transformer architecture to learn about language representation in an unsupervised manner. We fine-tune the *BERT-Large Uncased*[1] model.
- ALBERT (Lan et al., 2019): This is an efficient (**A** **Lite** **BERT**) version of BERT which outperformed BERT in several benchmark data sets. In this paper, we experiment with the *ALBERT-xxlarge V2*[2] model.

BERT-like models encode the provided input using different types of embeddings for tokens, segments and positions. These embeddings were initially trained on large data sets and later on fine-tuned for specific tasks. Similarly, in our case, a pair of sentences, which were separated from each other by a separator token ([SEP]), is fed into the model during the fine-tuning phase. This fine-tuned model is used for predicting whether two sentences are event coreferent or not.

BERT and ALBERT return either 0 or 1 as the prediction output. The prediction 1 is interpreted as the pair of sentences refer to the same event and 0 as they refer to different events. In order to make a better distinction between these outputs, we use -1 instead of 0 to represent the pairs which are not coreferent.

3.2. Re-scoring Sentence Pairs

As a result of the same event prediction step, all pairs have scores either 1 (when they refer to the same event) or -1 (when they refer to different events). For each pair, in addition to using this score, we also consider how this pair of sentences are in relation to other sentences. For instance, assume that two sentences s_i and s_j are predicted to be referring to the same event; therefore, they have $Score(s_i, s_j) = 1$. However, the prediction result between s_i and s_k can be same or different than the prediction result between s_j and s_k. If they are both 1, we increase the $Score(s_i, s_j)$; otherwise, if they are different, we decrease the score.

[1] https://github.com/google-research/bert

[2] https://github.com/google-research/ALBERT

The main idea here is to calculate a score for a pair sentences not just based on the pair itself but using their agreements and disagreements with other sentences as well. For any pair of sentences, s_i and s_j, among the other sentences, s_k, if there are many of them where s_i and s_j have the same prediction, then the likelihood of putting s_i and s_j to the same cluster should be higher. If the number of disagreements is higher, then the likelihood of putting s_i and s_j to the same cluster should be lower.

The proposed re-scoring algorithm is described in Algorithm 1. BERT is used to represent our fine-tuned BERT and ALBERT models. It can be replaced with any other classification model.

Algorithm 1 Re-Scoring Pairs

```
$All\_Scores \leftarrow []$
$Sentences \leftarrow$sentences in the news article
**for** $s_i$ in $Sentences$ **do**
  **for** $s_j$ in $Sentences$ where $s_j \neq s_i$ **do**
    **if** BERT($s_i$, $s_j$) = 1 **then**
      $Score(s_i, s_j) \leftarrow 1$
    **else**
      $Score(s_i, s_j) \leftarrow -1$
    **end if**
    **for** $s_k$ in $Sentences$ where $s_k \neq (s_i$ **or** $s_j)$ **do**
      **if** BERT($s_i$, $s_k$) = 1 **and** BERT($s_j$, $s_k$) = 1 **then**
        $Score(s_i, s_j) \leftarrow Score(s_i, s_j) + reward$
      **else if** BERT($s_i$, $s_k$) $\neq$ BERT($s_j$, $s_k$) **then**
        $Score(s_i, s_j) \leftarrow Score(s_i, s_j) - penalty$
      **end if**
    **end for**
    INSERT $Score(s_i, s_j)$ into $All\_Scores$
  **end for**
**end for**
```

In Algorithm 1, $reward$ and $penalty$ can be set to different values between 0 and 1. The optimum values are identified using the validation data.

3.3. Constructing Event Clusters

After re-scoring the pairs, these updated scores are used to create the clusters, and for this clustering part, we use a greedy algorithm. Initially we assume that none of the sentences belongs to a cluster. Among all pairs of sentences, we only consider the ones where the score of the pair is higher than 0. For the rest, where score is 0 or less, we assume that they cannot belong to the same cluster; therefore we ignore those cases.

We sort all pairs with scores higher than 0 by their scores in descending order and, in case when there is a tie in the scores, we give priority to the sentences with lower indices. By giving that priority, we aim to start the event clustering from earlier sentences as that is how we expect the events are presented in the news articles as well. Therefore, the idea is that, in case of a tie, place the pair with the smallest sentence number before the other ones.

After sorting the pairs based on their scores and sentence indices, we begin to cluster the sentences starting with the pair with the maximum score. This merging continues until either (1) there are no more pairs of sentences left with score higher than 0, or (2) when every sentence is merged into some cluster already. In the first stopping condition, if there are any sentences left unclustered, we consider those as individual clusters. This clustering algorithm is summarized in Algorithm 2.

Our approach is similar to hierarchical clustering, as it creates clusters in a bottom-up fashion. Instead of using the minimum distance, we use the maximum score to decide the clusters.

Algorithm 2 Clustering

```
$Sentences \leftarrow$ sentences in the news article
$Groups$ $\leftarrow$group assignments for all $Sentences$, initially all are assigned to group 0
$All\_Scores \leftarrow$ scores retrieved from re-scoring sentence pairs
SORT ($All\_Scores$ by descending order of $scores$ and ascending order of $sentence\_ids$)
FILTER($All\_Scores$ by $scores > 0$)
$num\_of\_groups \leftarrow 0$
**for** $s_i, s_j$ in $All\_Scores$ **do**
  **if** $Groups(s_i)$ = 0 **and** $Groups(s_j)$ = 0 **then**
    $num\_of\_groups \leftarrow num\_of\_groups + 1$
    $Groups(s_i) \leftarrow num\_of\_groups$
    $Groups(s_j) \leftarrow num\_of\_groups$
  **else if** $Groups(s_i)$ = 0 **then**
    $Groups(s_i) \leftarrow Groups(s_j)$
  **else if** $Groups(s_j)$ = 0 **then**
    $Groups(s_j) \leftarrow Groups(s_i)$
  **end if**
**end for**
**for** $s$ in $Sentences$ **do**
  **if** $Groups(s)$ = 0 **then**
    $num\_of\_groups \leftarrow num\_of\_groups + 1$
    $Groups(s) \leftarrow num\_of\_groups$
  **end if**
**end for**
```

Source code of the proposed three steps approach is available online[3].

3.4. An Example

In order to show how the proposed algorithms perform with respect to a single news article, an example with 7 sentences and 2 clusters, is chosen from the training data. All three steps of the approach and their respective outputs are presented in Table 2.

The first two columns represent the constructed sentence pairs. For 7 sentences we construct 21 pairs in total. Column 3 presents the outputs of the coreference classifier for these 21 pairs. The output is 1 for sentences that are coreferent and -1 for sentences that are not. These are the scores before re-scoring. Column 4 displays the scores after re-scoring. Finally, the last column shows the filtered pairs (ones with score higher than 0), the order of pairs after sorting by score and indices and, finally step by step construction of the clusters.

[3] https://github.com/su-nlp/Event-Clustering-within-News-Articles

s_i	s_j	Scores Before Re-Scoring	Scores After Re-Scoring	Orders & Clusters
2	4	1	0	-
2	27	1	2	(2) [2,27,36]
2	36	1	2	(3) [2,27,36]
2	37	-1	2	(4) [2, 27, 36, 37]
2	40	-1	-3	-
2	43	-1	-3	-
4	27	1	2	(5) [2,4,27,36,37]
4	36	1	2	(6) [2,4,27,36,37]
4	37	1	0	-
4	40	1	-2	-
4	43	1	-2	-
27	36	1	4	(1) [27,36]
27	37	1	2	(7) [2,4,27,36,37]
27	40	-1	-4	-
27	43	-1	-4	-
36	37	1	2	(8) [2,4,27,36,37]
36	40	-1	-4	-
36	43	-1	-4	-
37	40	-1	-3	-
37	43	-1	-3	-
40	43	1	2	(9) [40,43]

Table 2: An Example for Re-Scoring and Clustering (Final clusters are the same as the actual clusters, which are [2,4,27,36,37] and [40,43])

Comparing columns 3 and 4 shows the impact of re-scoring. All 21 scores have changed, either increased or decreased. Among these, the most important ones are the ones in colored cells. In these three cases, the re-scoring does not only change the score but also the sign of the score, which directly affects the final clustering. If we use the initial scores without re-scoring, than all 7 sentences will be clustered into the same cluster. However, the re-scoring step corrects the wrong prediction between pairs 4-40 and 4-43, which at the end leads sentences 40 and 43 to end up in a different cluster than the rest of the sentences. Constructing the clusters with the re-scored pairs returns the same clusters as the actual golden standard clusters.

4. Experiments

During development, we divide the provided 404 news articles into two splits (80% for training and 20% for validation). After splitting the news articles, sentence pairs are constructed for the training and validation sets. At the end, 3758 pairs of sentences are constructed for training and 1076 pairs for the validation part. During the development phase, the validation data is used to compare the performance of models. In this section, we report some experimental results over this data.

For the final phase, we received another 100 test samples from the organizers. All 404 training instances are used to train our final model to be tested on this 100 samples. Our final best model's performance over this test sample is also reported in this section.

4.1. Evaluation Metrics

The evaluation script provided by the organizers is used to evaluate the models. Adjusted Rand Index and F1 measures are reported. Since the number of sentences in the input news article has direct impact on the size of the hypothesis space and the complexity of the problem; two different averaging mechanisms are used in evaluations.

- Macro: Averaging the scores despite the number of sentences in the news article. This measure weights all news articles equally likely; therefore, it is an unweighted approach.
- Micro: This weighted metric is calculated by multiplying the score of each news article by the number of event containing sentences it contains, and then dividing the sum with total sentence count across all news articles. In other words, news articles, which contain more event containing sentences, are weighted more. As a result, more complex test cases have higher impact on the final score.

4.2. Baseline System

The baseline system developed by the organizers is two fold, which is similar to Mention-Pair models.

- As the first step, they evaluated each possible sentence pair and predicted whether they are coreferent or not. The organizers used a multi-layered perceptron model for the prediction task but the details of the model are unknown at this point.
- As the second step, they used the Correlation Clustering algorithm (Bansal et al., 2004) to process and cluster the predicted pairs from the first step.

4.3. Same Event Prediction Experiments

Before analyzing the results of the event clustering, we initially compare the performances of BERT and ALBERT on the *same event prediction* task. Results of the experiments over the validation set are presented in Table 3.

Model	Accuracy	Precision	Recall	F1
BERT	0.741	0.739	0.741	0.734
ALBERT	0.784	0.784	0.784	0.780

Table 3: Results of the Same Event Prediction Part

As seen from the Table 3, ALBERT outperforms BERT in predicting whether sentences are referring to the same event or not. Due to its better performance, we continue working with the ALBERT model in the following experiments.

4.4. Cluster Construction Experiments

In order to compare how our proposed re-scoring and cluster construction algorithms compare with respect to Correlation Clustering (CC) algorithm used in the baseline, we apply all these approaches to the prediction outputs of sentence pairs. We perform two experiments in order to analyze the individual effects of our two proposed approaches, re-scoring and clustering.

In the first experiment, we skip the re-scoring phase and directly cluster the sentences based on their initial scores from the prediction model, which are either 1 or -1. This setting is referred as *w/oRS+C*. As the second experiment, we re-score the pairs and then cluster, which is referred as *w/RS+C*. In this one, during the re-scoring part both the *reward* and *penalty* are set to 1. Results of these experiments on our validation data are presented in Table 4.

	ARI		F1	
	Macro	Micro	Macro	Micro
Baseline CC	0.5359	0.3964	0.6914	0.6232
Ours *w/oRS+C*	0.6231	0.5277	0.6739	0.5866
Ours *w/RS+C*	0.6088	0.5293	0.7220	0.6831

Table 4: Evaluation Results over Train/Val Splitted Data

According to the results, our proposed approach outperforms the Correlation Clustering (CC) algorithm. Using the proposed clustering algorithm standalone without the re-scoring part provides significant improvements compared to the CC algorithm in ARI metric. When the proposed clustering is combined with the re-scoring phase, drastic improvements are also observed in the F1 Measure.

In our proposed approach, the main bottleneck in terms of running time comes from the re-scoring part which has a time complexity of $O(n^3)$, where n is the number of sentences. Overall, since the number of sentences containing event is limited (on average 4.5 sentences), this running time is acceptable given the improvement in the F1 Measure.

4.5. Effect of Training Size

In the initial data set, we were provided with 404 news articles, and among those, we use 80% for the training which makes a total of 3758 pairs of sentences. Unfortunately, this is still a limited data set for fine-tuning a model. In order to see whether using a larger training set would give a higher performance, we keep everything same, except for the training set size and train different models.

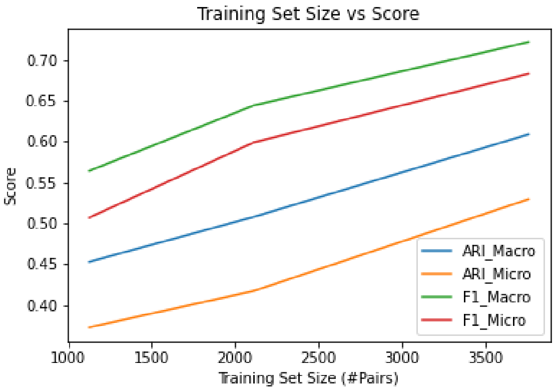

Figure 1: Performance Change with respect to Different Training Set Sizes

Testing these models on the same validation set returns the results in Figure 1. As seen from the figure, as the number of observations in training set increases, the performance consistently improves across all metrics. This indicates that even a transfered state-of-the-art pre-trained model may not be fine-tuned easily for different end-tasks. Increasing the training data would be definitely useful.

4.6. Fine-tuning *reward* and *penalty* Scores

In Table 4, the experiments are performed after setting both *reward* and *penalty* to 1. In such a case, for a pair of sentences, s_i and s_j, the *same event prediction's* result between these two sentences has the same effect as these two sentences being in agreement with other sentences. Normally agreement or disagreement with respect to other sentences may have lower effect on the final score compared to the pairwise prediction score of these sentences. Therefore, fine-tuning the values of *reward* and *penalty* may result in more effective re-scoring and clustering.

Values from 0.6 to 1 with an increase rate of 0.1 are used to fine-tune the *reward* and *penalty* scores over the validation set. Different optimum values are obtained for different metrics. Results for all 4 metrics are presented in Figure 2.

In Figure 2, the worst performance is obtained when both the *reward* and *penalty* is set to 1. *Penalty* equal to 1 performs poorly for the ARI metric, and similarly *reward* being set to 1 returns lower F1. Even though there is not a clear winner, based on the performances, both *reward* and *penalty* are set to 0.8 in the final model. The final results obtained with these values are presented in Table 5. As seen from Table 5, even a slight decrease in the *reward* and *penalty* rates leads to an important increase in the final results.

reward/penalty	ARI		F1	
	Macro	Micro	Macro	Micro
1.0 / 1.0	0.6088	0.5293	0.7220	0.6831
0.8 / 0.8	0.6500	0.5749	0.7440	0.7095

Table 5: Evaluation Results with varying *reward* and *penalty* values with *w/RS+C* approach

4.7. Experiments on Test Set

Finally, based on our experiments over the validation set, using ALBERT together with our proposed clustering approach with *reward* and *penalty* set to 0.8 is our best model. Retraining this same model over the whole training data and testing it over the test data set returns the following results in Table 6. As observed, our best model consistently outperforms the baseline model across all metrics.

	ARI		F1	
	Macro	Micro	Macro	Micro
Baseline Model	0.5077	0.4064	0.5560	0.4842
Our Submission	0.6006	0.4644	0.6736	0.5898

Table 6: Evaluation Results over Test Data

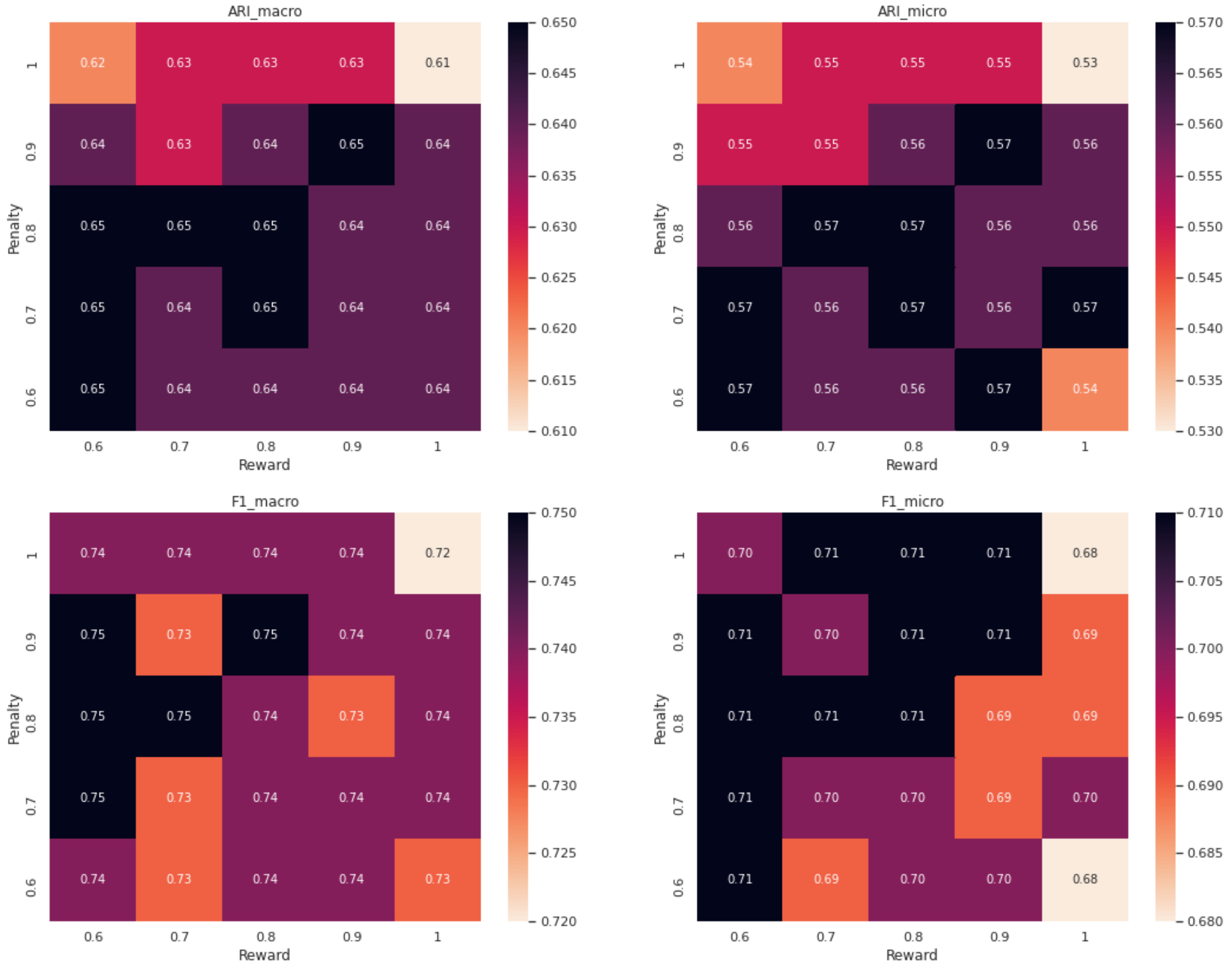

Figure 2: Performance heatmap for different values of *reward* and *penalty* scores

5. Conclusion

This paper summarizes our initial explorations on event sentence coreference identification within news articles. We propose a three-step approach, which is based on mention-pair model. Overall, these approaches independently and jointly work good enough to outperform the shared-task's baseline.

In future, we will perform detailed analysis of these approaches and continue improving these individual steps for our end task. An idea is to integrate the classifier's confidence levels to the scoring mechanism. Instead of using just the classification output as -1 or 1 at the initial scoring and re-scoring steps, we will analyze the effects of using classifier's confidence values directly.

6. Acknowledgements

We would like to thank to the organizers of the workshop and the shared task for organizing such an interesting challenge. We are also grateful to Dr. Ali Hürriyetoğlu for his timely and detailed responses to all of our questions during the challenge. We also thank to the anonymous reviewers for their useful and constructive feedbacks.

7. Bibliographical References

Bansal, N., Blum, A., and Chawla, S. (2004). Correlation clustering. *Machine learning*, 56(1-3):89–113.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hürriyetoğlu, A., Yörük, E., Yüret, D., Yoltar, Ç., Gürel, B., Duruşan, F., Mutlu, O., and Akdemir, A. (2019). Overview of clef 2019 lab protestnews: Extracting protests from news in a cross-context setting. In Fabio Crestani, et al., editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 425–432, Cham. Springer International Publishing.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Lu, J. and Ng, V. (2018). Event coreference resolution: A survey of two decades of research. In *IJCAI*, pages 5479–5486.

Ng, V. (2010). Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th ACL*, pages 1396–1411.