Event Sentence Detection Task Using Attention Model

ProtestNews Lab at CLEF 2019

Ali Safaya

Sakarya University, Adapazarı, Sakarya 54055, Turkey
alisafaya@gmail.com

Abstract. This paper describes and evaluates a model for event sentence detection in news articles using Attention Models with Bidirectional Gated Recurrent Network (GRU) and Word Embeddings. This model was developed for event sentence detection task in the competition that was organized by ProtestNews lab at CLEF 2019. We also evaluated the generalizability of NLP tools by training our model on data from one country and testing it on data from another country. The model was developed for this task was shown to have the highest score in the organized competition with average F1-score of 0.6547.

Keywords: Information extraction, Natural language processing, Sequence classification, Event sentence detection

1 Introduction

This task aims to identifying and labeling sentences that contain protest events in news articles. It follows the document labeling task which identifies news articles that contain protest events as identified in the Event Labeling Annotation Manual [1]. Once the news reports are classified as containing a protest event, what remains is to identify where in the article the relevant event information is presented. In terms of this task, we will analyze the sentences of the protest news articles one by one and classify them as event-sentence vs. non-event-sentence.

Event sentences, those that are labeled as 1, should contain an explicit reference to any protest event that makes the document eligible for being classified as a protest article. Such reference can be any word or phrase which denotes the said event. They can be direct expressions of the event or the pronouns which stand for the event. The sentence must clearly indicate that the event in question has definitely happened in the past or is an ongoing event.

Non-event sentences, i.e. those that are labeled as 0, are the ones which does not contain any event reference in the past, the present or the future.

The main goal for this task is to set a baseline in evaluating generalizability of the NLP tools. The setting was proposed facilitates testing and improving state-of-the-art methods for text classification and information extraction on English news article texts from India and China. The direction of ProtestNews lab work is towards developing generalizable information systems that perform comparatively well on texts from multiple countries [2].

2 Data Collection and Methodology

ProtestNews lab organizing committee have collected online English news articles from India and China. The annotation process started by labeling articles in a sample of news articles as containing protest or not which will be used for Task 1. Sentences of these positively marked documents are then labeled as containing protest information or not. These sentences should contain either an event trigger or a reference to an event trigger in order to be labeled as positive [2].

Table 1. Distribution of collected data samples

Data set	Negative	Positive	All
Training-India	4897	988	5885
Validation-India	525	138	663
Test-India	*	*	1107
Test-China	*	*	1235

(*) Was kept hidden by the commission of the lab.

Our deep learning based model was trained using Training-India set and Validated on Validation-India set, the data retrieved from China was not involved in training at all, so when the model was evaluated on the test sets we could obtain independent and generalized results.

F1-score metric (1), was used in the evaluation process for this kind of tasks where there is non-equal number of negative and positive samples.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (1)$$

2.1 Preprocessing and Tokenization

Before feeding data into the model, our text samples which are sentences taken from articles were to be cleaned and parsed into lower case words.

Because word embeddings were used in the classification process, a word index had to be created according to the embeddings set in use and once word sequences were

obtained, some of irrelevant words had to be dropped and those words were determined by the word index.

Also the sequences had to have fixed length of tokens, so sequence length was limited to 35 tokens. Longer sequences where truncated and shorter sequences were pre-padded with 0 indexed token in order to reach 35 tokens.

2.2 Word Embeddings

To feed those word sequences to our deep learning model every token had to be represented by some value or vector. In this model word based representations were used, so every word was replaced by embedding vector. This embeddings vector set was obtained from Google's pretrained set [3]. Which was trained using word2vec [4] algorithm on part of Google News database (100 billion words) and contains 300 dimensional vectors for 3 million English words. In this work only the most frequent 400000 vocabulary were used in the word index.

Every token was replaced by 300 dimensional vector and maximum sequence length was limited to 35 token. So every sentence was represented by matrix of shape (300, 35).

3 Bidirectional Gated Recurrent Unit (GRU)

In sequence classification tasks, Recurrent Neural Networks (RNN) and its variations had always been the state-of-art tool. After obtaining an embedding for each sample, the main approach will be using bidirectional GRU [5]. As Fig. 1 shows, every layer of Bidirectional GRU, contents of GRU cells for each direction.

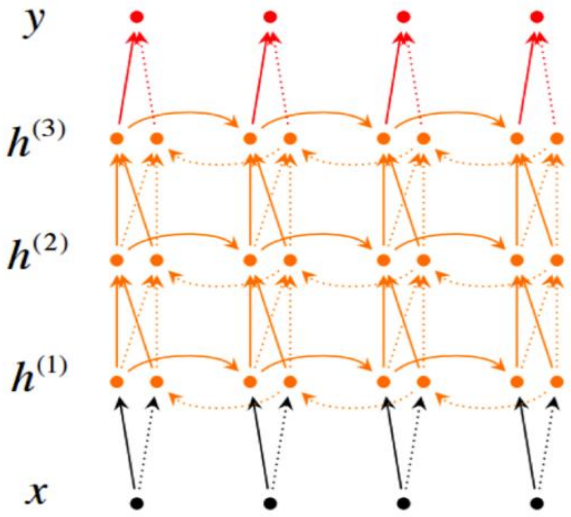
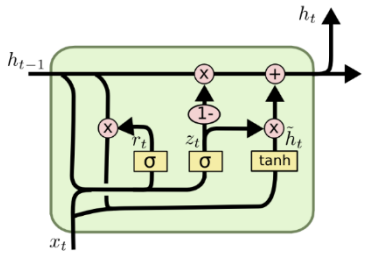


Fig. 1. Bidirectional GRU network model

Every cell has two gates; an update gate z_t and reset gate r_t (Fig. 2). Sigma representations demonstrate these gates: which allows a GRU to carry information over many time periods to influence a future time zone.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Fig. 2. GRU Cell structure[1]

4 Attention Models

Attention Models were firstly represented in 2015 by Dzmitry Bahdanau et al [5]. Past conventional methods used to find features from the text by doing a keyword extraction and some words are more helpful in determining the category of a text than others. However, in this method the sequential structure of the text is not fully used. With deep learning methods, while we can take care of the sequence structure, the ability to give higher weight to more important words is lost.

The firstly proposed model was meant for Machine Translation purposes, while using Attention Models mechanism for text classification tasks was proposed in the paper written jointly by CMU and Microsoft in 2016 [6]. In author's words:

Not all words contribute equally to the representation of the sentence meaning. Hence, we introduce attention mechanism to extract such words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence vector

In this model (see Fig. 3) the Attention layer is added after the last GRU layer. So the Attention Models output is the dot product of Attention Similarity Vector s_i and GRU cells output a_i.

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/ last accessed on June 2019

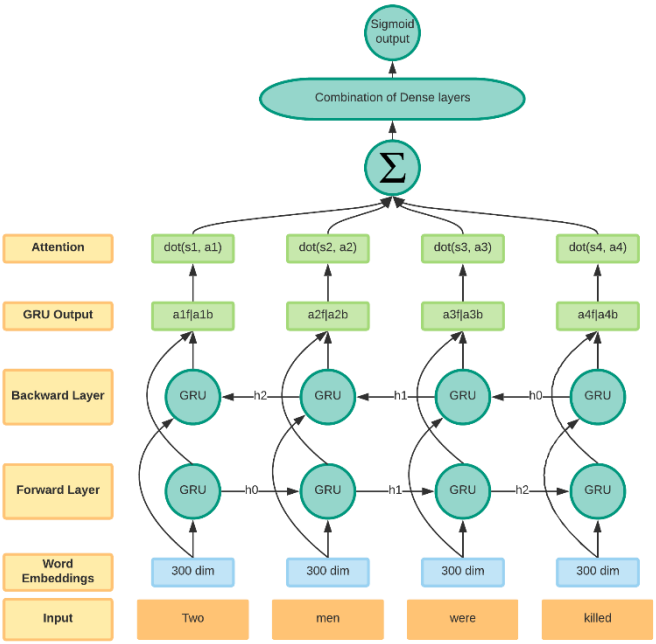


Fig. 3. Bidirectional GRU with Attention Model

The main goal is to create scores (s_i) for every word in the text, which is the attention similarity score for a word. Here in Fig. 4, we could see how those scores are calculated.

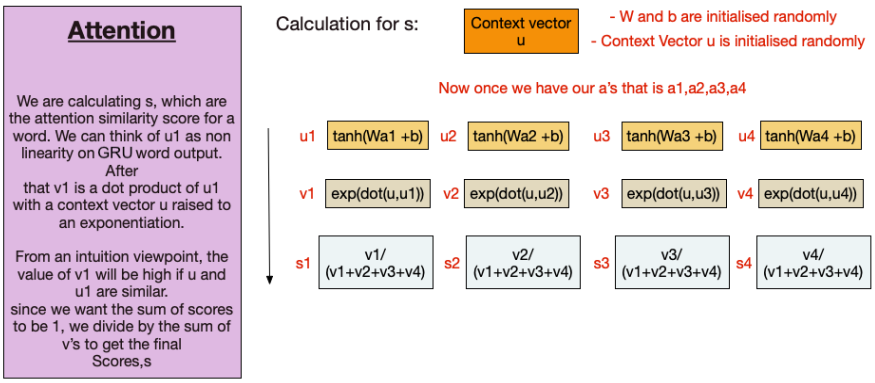


Fig. 4. Context vector calculation method[2]

[2] https://mlwhiz.com/blog/2019/03/09/deeplearning_architectures_text_classification/ last accessed on June 2019

These final scores are then multiplied by GRU output for words to weight them according to their importance. After which the outputs are summed and sent through dense layers and then to the last output function [7].

5 Modeling The Network and Evaluation

Machine learning model is shown in Fig. 4. After Embedding layer two Bidirectional GRU layers are introduced, with 128, 64 cells respectively. On the top of them an Attention with Context layer was added and followed by dense layer of 64 nodes with ReLU as their activation function. Finally an output layer was added with one node containing sigmoid function for binary classification output.

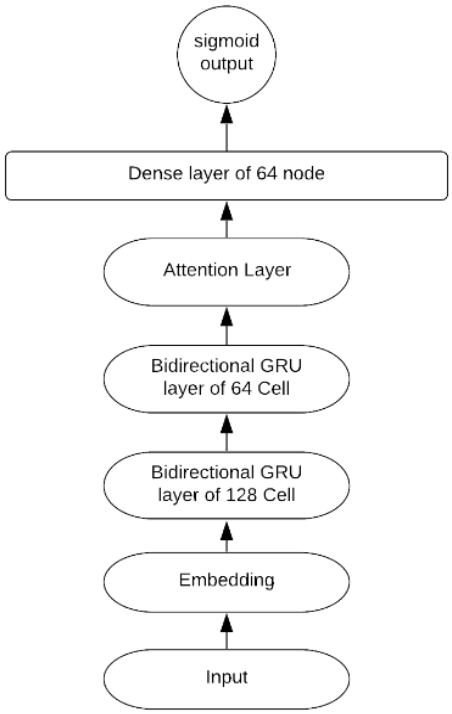


Fig. 5. Our deep learning models structure

The model was trained on Training-India dataset (see Table 1) for 8 epochs using Nadam [8] as optimizer function and validated through Validation-India dataset.

While testing the model on test datasets it could be observed (as in Table 2) that performance (F1-score) dropped from 0.70 on Test-India dataset which is the same source that Training data was obtained, to 0.60 on Test-China dataset.

Table 2. Evaluation of trained model on different datasets

Metric	Training-India	Validation-India	Test-India	Test-China
F1	0.7984	0.7094	0.7055	0.6039
Precision	0.7137	0.7402	*	*
Recall	0.9059	0.6812	*	*

(*) Was kept hidden by the commission of the lab.

6 Conclusion

In this task we worked on deep learning model which tries to classify and detect event sentences in news articles. The proposed model uses Bidirectional GRU with Attention Models. The results obtained from this model were the highest in the competition which had been organized by ProtestNews Lab.

With this experiment we could observe the effect of local data on NLP tools, our test results on datasets from the same source of training sets were noticeably higher than those on datasets from other sources.

For further work, we could evaluate POS based features of the words in the sentences by adding one more input layer in parallel with embedding layer.

References

1. ProtestNews lab Homepage, https://emw.ku.edu.tr/clef-protestnews-2019/, last accessed 23.05.2019.
2. Hürriyetoğlu, A., Yörük, E., Yüret, D., Yoltar, Ç., Gürel, B., Duruşan, F., and Mutlu, O. (2019, April). A Task Set Proposal for Automatic Protest Information Collection Across Multiple Countries. In European Conference on Information Retrieval (pp. 316-323).
3. Google Code word2vec page, https://code.google.com/archive/p/word2vec/, last accessed 24.05.2019.
4. Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013). Efficient estimation of word representations in vector space. In Proceedings of Workshop at ICLR, 2013.
5. Bahdanau, D., Cho, K. and Bengio, Y. (2015). Neural Machine Translation By Jointly Learning To Align And Translate. In Proceedings of ICLR, 2015.
6. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A. and Hovy, E. Hierarchical Attention Networks for Document Classification. Carnegie Mellon University, Microsoft Research, Redmond
7. MLWhiz, NLP Learning Series: Part 3-Attention, CNN and what not for Text Classification, https://mlwhiz.com/blog/2019/03/09/deeplearning_architectures_text_classification/ , last accessed 23.05.2019
8. Dozat, T., Incorporating Nesterov Momentum into Adam, http://cs229.stanford.edu/proj2015/054_report.pdf, last accessed 24.05.2019