Design Event Extraction Model from Amharic Texts Using Deep Learning Approach

Amogne Andualem[1,2(✉)] and Tesfa Tegegne[1,2]

[1] Faculty of Computing, Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia
[2] ICT4D Research Center, Bahir Dar Institute of Technology, Bahir Dar University, Bahir Dar, Ethiopia

Abstract. Every day, a massive amount of information is reported in the form of video, audio, or text through various media such as television, radio, social media, and web blogs. As the number of unstructured documents on those media has grown, finding relevant information has become more difficult. As a result, extracting relevant events from large amounts of unstructured text data is essential. We proposed an event extraction model, which aims to detect, classify and extract various types of events along with their arguments from Amharic text documents. In this paper, the researchers first come up with Amharic language-specific issues and then proposed Bidirectional Long Short Memory (BiLSTM) with a Word2vec model to detect and classify Amharic events from unstructured documents. To achieve this research 9,050 Amharic documents were used for event detection and extraction purpose. In addition to event detection and classification, the model also extracts event arguments that contain additional information about events such as Time and Place. The experimental results showed that the Bidirectional long short-term memory approach with Word2vec word embedding shows a promising result in terms of Amharic event detection and event classification, with 94% and 89% accuracy, respectively.

Keywords: Natural language processing · Information extraction · Event extraction · Bidirectional long short-term memory · word2vec

1 Introduction

Natural language Processing (NLP) applications aids in the extraction of relevant information from large, unstructured text documents. Event extraction is one of the NLP applications that makes detecting and extracting events and their arguments, as well as classifying and tracking similar events from different texts. Events have many definitions among those the actual or contemplated fact of anything happening or occurrence at a specific time and place [1]. The ideal goal of Event extraction systems in the presence of huge volume digital text in the news, social media, and blogs are to produce the best possible extraction of text events with their arguments with minimal human intervention.

It can be done automatically using Natural Language Processing(NLP) and other machine learning algorithms to extract events with their arguments from text, including

M. L. Berihun (Ed.): ICAST 2021, LNICST 411, pp. 424–434, 2022.
https://doi.org/10.1007/978-3-030-93709-6_28

details on what happened, when it happened, and where it happened [2]. The most important sub tasks of event extraction in the NLP domain include extracting event arguments and identifying their roles, as well as classifying and tracking similar events from different texts. Another important aspect of event extraction is event arguments, which discuss what happened, when it happened, where it happened, and who took part during the occurrence of the event [3]. Event argument extraction is the process of locating incidents that occurred at a specific time and location, extracting a set of properties from those incidents, and converting unstructured texts into a structured representation of those incidents [4]. Currently, this task is performed manually by media analysts, digital news editors, collecting, interpreting, and presenting news from multiple news sources.

This manual task is tedious and time-consuming for journalists and news staff. Amharic event extraction researchers recently used a machine learning algorithm for binary classification and some rule-based techniques for identifying and extracting events in Amharic text documents [5, 6].

2 Related Studies

Due to the wide availability of resources in various European languages, event extraction research has gained popularity in the last decades. Because of the differences in the language's morphological structure, we cannot use the existing techniques and tools developed for other languages without modification for our work.

Until today, several event extractions studies have been developed using machine learning algorithms and deep learning with some limitations. Furthermore, to the best of our knowledge, no deep learning-based event extraction model for Amharic text events has been proposed previously. Besides this, Hordofa in [5] proposed a model for extracting events from Amharic News articles that uses traditional machine learning techniques as a binary classifier and an ontology technique for event detection. However, there are some limitations on this research. (1) some sub-tasks of Event Extraction, such as Argument Role classification and event classification, are not included; (2) handcrafted feature engineering is used as a feature extractor, resulting in incorrect event detection prediction probability. Similar approaches have also been adopted in [6], in this work was related to Amharic event extraction from unstructured Amharic texts using machine learning and a rule-based called hybrid approach. However, in this research, the researchers did not incorporate basic sub-tasks of event element extraction, event classification, and Argument Role classification.

Other than those studies, there has been no other work in Amharic event extraction until today. However, many works on event extraction have been proposed in other resourceful languages such as English, France, and others [7–10]. The majority of research has been focused on detecting, extracting, and matching event slots on pre-defined templates in domain-specific event extractions [11]. There are several traditional machine learning text classification models that have been employed for text events extraction tasks such as the K-Nearest Neighbor classification algorithm, Support Vector Machine algorithm, Bayesian classification algorithm, Decision Tree, and others have been used.

Before years, many researchers developed many event extraction systems using an unsupervised machine-learning algorithm for Event Detection, Event classification, Argument identification, and argument role classification tasks [15–17]. However, existing machine learning approaches, on the other hand, have a variety of drawbacks for text classification applications. The major flaw in the traditional machine learning approach is it extracts text features using handcrafted rules, which rely heavily on the designer's prior knowledge and make big data use nearly impossible.

Deep learning has a strong nonlinear mapping capability on different NLP application development and is capable of extracting abstract hierarchical features from complex text data [18]. It improves the accuracy of the event detector by checking whether every token in a given input sentence is an event trigger of some pre-defined event type or not using convolutional neural networks (CNN) and recurrent neural networks (RNN) [19].

Recently, many researchers have applied recurrent neural networks to automatically extract different text features such as distribution of words, capture word syntax, and semantics of words from natural language written texts, and improving the performance of event detector and event classifier models [20, 21].

3 Proposed Event Extraction Model

The proposed Amharic Event Extraction Model consists of many subtasks: Dataset Preparation, Preprocessing, Event detection, Event Classification, and Event argument Extraction. The following figure depicts the overall proposed Amharic event extraction model.

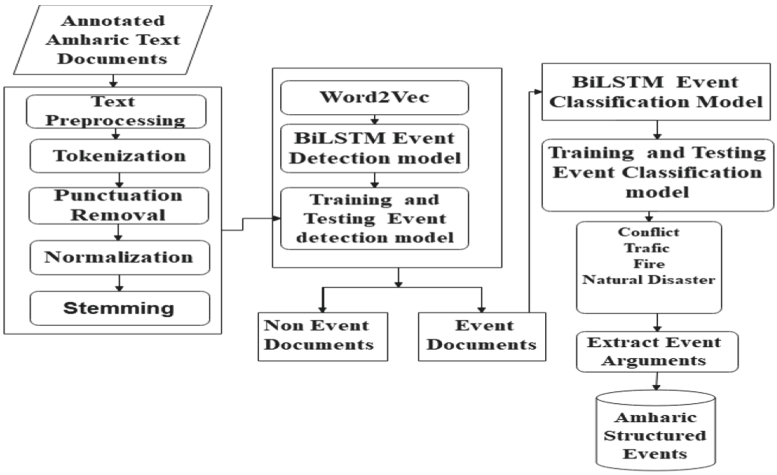

Fig. 1. Proposed Amharic event extraction model

3.1 Dataset Preparation

Data set for Natural Language Processing (NLP) tasks is crucial, specially to train and test the model using traditional machine learning and deep learning approaches. To determine the system's accuracy, event extraction requires a large corpus for training, validation, and testing. For event classification, we focus on disaster and accident data namely fire accident (የእሳት አደጋ), conflict (ግጭት), car accident (የመኪና አደጋ), Natural Disaster (የተፈጥሮ አደጋዎች). All collected Amharic documents were annotated by Amharic language experts using annotation guidelines prepared by the researchers.

3.2 Preprocessing

Data preprocessing is a crucial and fundamental step in the event extraction model development process. Because the Amharic texts collected have a variety of formats and characteristics. To produce structured Amharic texts documents, it is necessary to analyze the collected raw texts using various techniques. Under the preprocessing module the following sub tasks have been performed tokenization, punctuation removal, stop word removal, character normalization, and stemming.

Tokenization. It is the process of identifying all words in a document by using space as the primary separator factor. Furthermore, in the Amharic language writing system, not only space but also Amharic punctuation marks are used to separate words or phrases. To tokenize or convert an Amharic document into a sequence of words, the proposed algorithm used space and other special characters as delimiters.

Remove Punctuations. Amharic language writing system uses different punctuation marks for different purposes. For example, አራት ነጥብ ።(full stop) used for end of the sentence, የጥያቄ ምልክት(?) (question mark) used for interrogative sentences, ነጥላ ሰረዝ ፣(semicolon) used for separate the list object, etc. When the model is being trained, those punctuations in the sentence cause ambiguity. As a result, the developed punctuation removal algorithm removes punctuation from the Amharic text document.

Remove Stop Words. Stop words are extremely common words across document collections that have no discriminatory power from the collection. Amharic stop words have little semantic content in the document, high-frequency words, or highly occurred in the document collection. To avoid those words from the document we prepared 226 Amharic stop word lists. Following that, we proposed an algorithm for removing those words from the training and testing documents.

Character Normalization. There are characters in the Amharic language that have similar roles and redundant in similar words. Such as ኀ, ሀ, ሐand ሰ, ሠand አ, ዐand ጸ, ፀare superficial difference along with their sequences. Those character representations may have different meanings in NLP tasks including event extraction. As a result, such inflections must be normalized to a single common character. To avoid this type of inflection, we replaced a set of characters with the same meaning as the most common characters by using the previously proposed normalization algorithm.

Stemming. It is the process of reducing morphological variants to a single root word in natural language processing. It is language-dependent, which means that different

stemming techniques are required for each language due to morphological differences. Amharic words can take on a variety of morphologies by adding a prefix, suffix, or infix. As a result, the researchers proposed a rule-based stemmer algorithm by collecting different root words from various domains to address the aforementioned issue by finding their root from the collection and stem the prefix and suffix of a word.

3.3 Word Embedding

As shown in Fig. 1, after completing the text preprocessing module, each word must be numerically valued using appropriate embedding techniques. Words are essential units of letter formed language and sequence of characters like ‘በላ’ (he ate), ‘መጣ’ (he came), etc. To understand by machine learning algorithms, each word in a document must be represented as a real-valued vector.

As a result, word embeddings have been successfully used in a variety of natural language processing and it is a vector that not only represents syntactic but also semantic similarities between words in a document [22]. In our study after performed preprocessing steps, we change the word into vector form using Word2vec- Continuous Bag of a Word (CBOW) to use vectors as input for deep learning models called BiLSTM and classify Amharic text events effectively.

We have used 420,910 documents to proposed word2vec word embedding model with 200 Embedding dimensions and other important setups. Amharic Event Detection model

Event detection is a subcomponent of the event extraction model that identifies events in a text document. We proposed a Bidirectional Long Short-Term Memory (BiLSTM) recurrent neural network algorithm to extract context information from an Amharic text sequence. It is a special type of neural network that is proved to be extremely effective in capturing long and short-term dependencies in sequential data in a forward and backward direction.

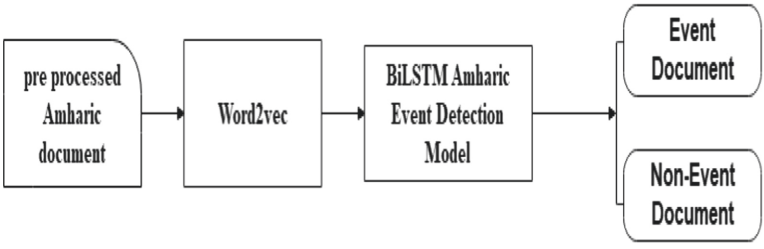

Fig. 2. Event detection model using Bi-LSTM techniques

As shown in Fig. 2, Bi-LSTM binary event classification component, the goal is to determine whether a given input Amharic text document contains an event or not called event detection.

3.4 Amharic Event Classification Model

The event and non-event documents are identified by the event detection model from the previously mentioned BiLSTM based event detection model. The outputs of the Bi-LSTM layer are merged and combined into a single matrix and passed to the fully connected layer. In Bi-LSTM independent event classification model takes events for the BiLSTM event detection model and classify events in the four predefined event type such as traffic (ትራፊክ), fire (እሳት), conflict (ግጭት), and natural disaster (የተፈጥሮ አደጋዎች).

3.5 Event and Event Argument Extraction

As we mentioned above Event extraction has many components including event detection, event classification, and event Argument identification. Event Arguments are an important component of event extraction, discussing what happened, and when the event happened. The event arguments, on the other hand, are another component of the event that provides additional information about the event. For our study, we created a predefined list of all possible values of a named entity, referred to as a gazetteer. Using Named Entity Recognition, the model uses some specific rules to identify and extract arguments from the input Amharic text.

3.6 Event Extraction Experiment and Results

The word embedding, event detection, and event classification models are all sub-components of the proposed EE model, and they were all evaluated using appropriate evaluation metrics. For the Word2vec word embedding model, both the Continuous Bag of Words and the Skip-gram models were evaluated, and the CBOW model performed better, so it was used from each embedding layer of the event detection and classification model. For the event detection and event classification model we used precision, Recall, F1-score, and Accuracy as evaluation metrics and compare the three popular deep learning text classifier algorithms such as convolutional neural network (CNN), Long short-term memory (LSTM), and Bidirectional long short-term memory (BiLSTM).

3.7 Event Detection Model Experimental Results

We used 9,050 Amharic data sets for Event detection training and testing phase. The total data set were preprocessed and split into training, validation, and test sets. The training set is composed of 7,318 (80%) sentences, the validation set consists of 813 (10%) sentences, and the test set contains 919 (10%). By using zero post padded techniques, each sentence has a sequence length of 50 tokens. In the embedding layer used designed CBOW word embedding with 200 dimensions. The detailed experiment result from the BiLSTM with Word2vec Event detection and Classification model is discussed in the following table (Tables 1 and 2) respectively.

Table 1. BiLSTM event detection model evaluation result

Document size		Class	Evaluation metrics with Result		
			Precision	Recall	F-score
Training data	7,318	Event	0.93	0.97	0.95
Validation data	813	None event	0.97	0.90	0.93
Testing data	919	Testing accuracy			**94%**

To achieve the best model performance, we have changed the hyperparameter values that have a significant impact on our models, such as dropout value (0.4), optimizer (Adam), and learning rate values (0.001), Batch Size (128), number of epoch (20) and those values of each hyperparameter achieve the optimal performance value of the proposed Amharic Event detection model. Finally, as shown in the graph below, we trained the model to achieve optimal training and validation accuracy. One of the challenges we faced when training the model is avoiding overfitting between training and validation accuracy.

To solve such a problem, we used the dropout regularization technique that prevents neural networks from overfitting problems by modifying the cost function of the dropout value.

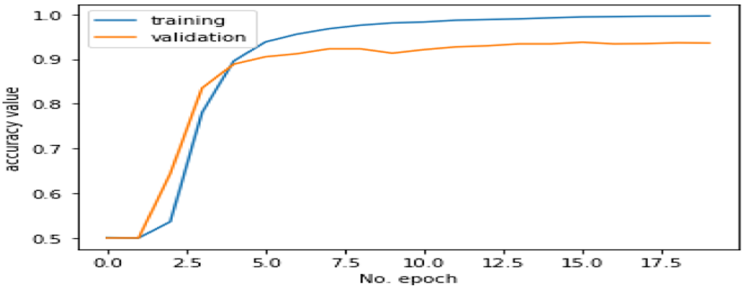

Fig. 3. Training and validation Accuracy graph

The above figure (Fig. 3) depicts that, the training and validation accuracy increase from one epoch to the next epoch. This shows that the model learns more Amharic event features from one epoch to the others. The following graph also shows that how to minimize the loss value from one epoch to the next epoch.

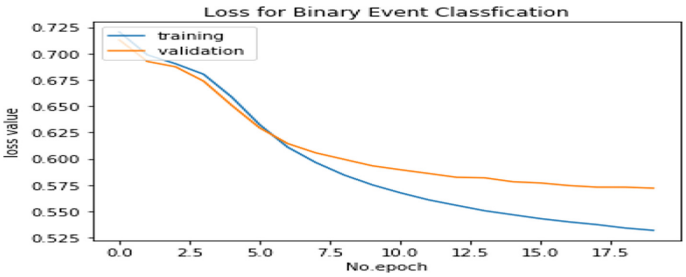

Fig. 4. Training and validation Loss from event detection BiLSTM model

The above figure (Fig. 4), shows the loss value of both validation and training is decreasing from one epoch to the next epoch. This graph shows that the event detection model learns more features from each epoch during the training phase and detect the input sentence without overfitting and underfitting problem.

3.8 Event Classification Model Experimental Results

The output of the BiLSTM event detection model is fed into another BiLSTM model, which identifies event types like traffic accidents, fire accidents, conflicts, and natural disaster accidents. On the training set, we begin training the BiLSTM classification model with four event classes label datasets. The dataset distribution for Amharic event classification experiment is shown in the diagram below (Fig. 5).

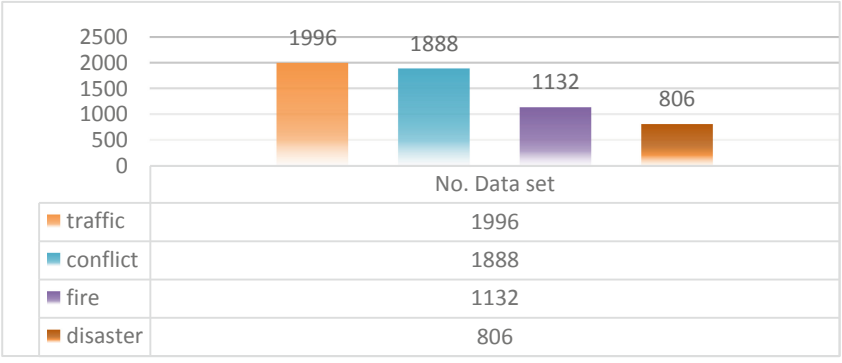

Fig. 5. Dataset distribution for event classification

This event classification experiment used a total dataset of 5815 event data as input to the BiLSTM event classification model. From the total data set 4,159 used for training, 462 for validation, and 1194 were used for testing purposes. The distribution of the dataset over four classes as traffic (ትራፊክ), fire (እሳት), conflict (ግጭት), Natural Disasters (የተፈጥሮ አደጋዎች). The data was split 80% for training, 10% for validation, and 10% for testing the splitting ratio.

Finally, we combined BiLSTM deep learning Amharic event classification with CBOW word embedding and found that the predicted model performed well, as shown in the table below.

Table 2. BiLSTM event classification experimental result

Class	Label	Precision	Recall	F-score
Traffic accident	0	0.94	0.88	0.92
Conflict	1	0.89	0.90	0.92
Fire	2	0.92	0.91	0.91
Natural disaster	3	0.79	0.84	0.78
Testing accuracy	**89%**			

The following figure (Fig. 6) shows that, the training and validation accuracy improves from one epoch to the next epochs.

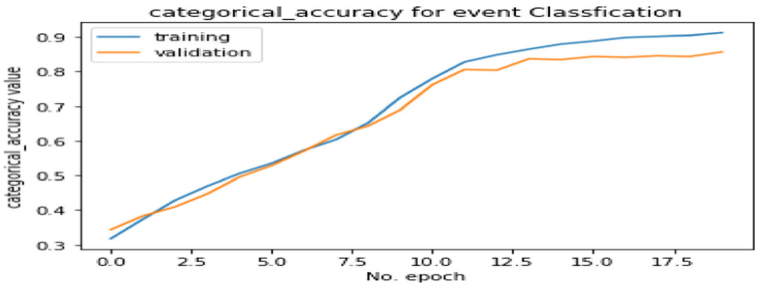

Fig. 6. Training and validation accuracy for BiLSTM event classification model

The last sub-component of event extraction, as mentioned in the previous section, is event argument extraction. In this paper, we develop a rule for extracting event arguments such as location and time. The rule can take an Amharic sentence or document as input, analyze each token, and then extract event arguments from the input document automatically.

After the three extensive experiments of the proposed model for Amharic event extraction, we observe that the BiLSTM with Word2vec outperforms the best result when comparing the other selected deep learning techniques LSTM and CNN.

4 Conclusion

The problem of event extraction for Amharic texts has been addressed in this paper. We proposed a BiLSTM deep learning approach for Amharic event detection and classification that can capture text contextual information. For the BiLSTM embedding layer, we used a proper text preprocessing technique and proposed the word2vec word

embedding model. The event detector model also detects events, and the classification model categorizes them into pre-defined categories like traffic accidents, conflict, fire, and natural disasters. To evaluate the performance of the proposed Amharic event extraction model, experiments are conducted on both event detection and classification of Amharic text data sets. The experiments show that BiLSTM combined with Word2vec is effective for both event detection and classification. In this study, the BiLSTM-word2vec model was compared to some current state-of-the-art deep learning methods.

The proposed model can detect events in the Amharic testing data set 82%, 83%, 92%, 94.2%, LSTM, CNN, BiLSTM and Word2vec BiLSTM respectively. Another proposed event classification model can classify events in the Amharic testing dataset 81%, 78%, 87.6%, and 89.4% for, CNN, LSTM, BiLSTM, and Word2vec-BiLSTM respectively. This may have an impact on the event extraction model. We also plan to propose an event extraction model that includes an Amharic grammar checker as well as a subcomponent called Name Entities Recognition (NER).

References

1. Polakof, A.C.: Why are events, facts, and states of affairs different? Disputatio **9**(44), 99–122 (2017). https://doi.org/10.2478/disp-2017-0029
2. Zhou, D., Chen, L., He, Y.: A simple Bayesian modelling approach to event extraction from Twitter. 52nd Annu. Meet. Assoc. Comput. Linguist. ACL 2014 - Proc. Conf., vol. 2, pp. 700–705 (2014). https://doi.org/10.3115/v1/p14-2114
3. Sahoo, S.K., Saha, S., Ekbal, A., Bhattacharyya, P.: A platform for event extraction in hindi. Proc. 12th Conf. Lang. Resour. Eval. (LREC 2020), no. May, pp. 11–16 (2020)
4. Petroni, F., et al.: An extensible event extraction system with cross-media event resolution. Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., pp. 626–635, (2018). https://doi.org/10.1145/3219819.3219827
5. Hordofa, B.A.: Event extraction and representation model from news articles. **16** (3, 1–8 (2020)
6. Tadesse, E., Aga, R.T., Qaqqabaa, K.: Event extraction from unstructured amharic text. no. May, pp. 2103–2109 (2020)
7. Nguyen, V.Q., Anh, T.N., Yang, H.J.: Real-time event detection using recurrent neural network in social sensors. Int. J. Distrib. Sens. Networks **15**, 6 (2019). https://doi.org/10.1177/1550147719856492
8. Björne, J., Salakoski, T.: Biomedical event extraction using convolutional neural networks and dependency parsing. 98–108 (2019). https://doi.org/10.18653/v1/w18-2311
9. Huang, L., et al.: Liberal event extraction and event schema induction. 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap., vol. 1, pp. 258–268 (2016). https://doi.org/10.18653/v1/p16-1025
10. Zhang, Y., Liu, Z., Zhou, W.: Event recognition based on deep learning in Chinese texts. PLoS ONE **11**(8), 1–18 (2016). https://doi.org/10.1371/journal.pone.0160147
11. Wang, W., Ning, Y., Rangwala, H., Ramakrishnan, N.: A multiple instance learning framework for identifying key sentences and detecting events. Int. Conf. Inf. Knowl. Manag. Proc., vol. 24–28-Octo, pp. 509–518 (2016). https://doi.org/10.1145/2983323.2983821

12. Ji, H., Grishman, R.: Refining event extraction through cross-document inference. ACL-08 HLT – 46th Annu. Meet. Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf., no. June, pp. 254–262 (2008)
13. Sahnoun, S., Elloumi, S., Yahia, S.B.: Event Detection Based on Open Information Extraction and Ontology. In: Nguyen, N.T., Chbeir, R., Exposito, E., Aniorté, P., Trawiński, B. (eds.) ICCCI 2019. LNCS (LNAI), vol. 11683, pp. 244–255. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28377-3_20
14. Ribeiro, S., Ferret, O., Tannier, X.:. Unsupervised event clustering and aggregation from newswire and web articles. pp. 62–67 (2018). https://doi.org/10.18653/v1/w17-4211
15. Zhou, D., Chen, L., He, Y.: An unsupervised framework of exploring events on Twitter: filtering, extraction and categorization. Proc. Natl. Conf. Artif. Intell. **3**, 2468–2474 (2015)
16. Valenzuela-Escárcega, M.A., Hahn-Powell, G., Hicks, T., Surdeanu, M.: A domain-independent rule-based framework for event extraction. ACL-IJCNLP 2015 – 53rd Annu. Meet. Assoc. Comput. Linguist. 7th Int. Jt. Conf. Nat. Lang. Process. Proc. Syst. Demonstr. pp. 127–132 (2015). https://doi.org/10.3115/v1/p15-4022
17. Miwa, M., Thompson, P., Korkontzelos, I., Ananiadou, S.: Comparable study of event extraction in newswire and biomedical domains. COLING 2014 – 25th Int. Conf. Comput. Linguist. Proc. COLING 2014 Tech. Pap. pp. 2270–2279 (2014)
18. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning based text classification: a comprehensive review. arXiv, 1(1), pp. 1–43 (2020)
19. Nguyen T.H., Grishman, R. Modeling skip-grams for event detection with convolutional neural networks. EMNLP 2016 – Conf. Empir. Methods Nat. Lang. Process. Proc. no. January, pp. 886–891 (2016). https://doi.org/10.18653/v1/d16-1085
20. Nguyen, T.H., Fu, L., Cho, K., Grishman, R.: A two-stage approach for extending event detection to new types via neural networks. pp. 158–165 (2016) https://doi.org/10.18653/v1/w16-1618
21. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. NAACL HLT 2016 – Proc. Conf. pp. 300–309 (2016). https://doi.org/10.18653/v1/n16-1034
22. Ma, J., Wang, S.: Resource–Enhanced Neural Model for Event Argument Extraction (2018)