Dockerized Knowledge-Oriented Multi-modal Social Event Detection System

Yuting Lyu
stevenlyu24@mosesbrown.org

Moses Brown School, RI

March 8, 2022

Abstract

Faced with an increasing amount of unstructured multimodal data appearing on various social platforms (e.g., Twitter/Instagram), we seek to effectively understand the complex social events portrayed on these platforms. However, conventional information extraction systems cannot understand these data because they cannot handle real-world analysis or require extensive tuning and many manually annotated examples to successfully comprehend these events. To solve this problem, this paper develops a knowledge-oriented artificial intelligence system that can identify and analyze these data and complex events and bring them to the user's attention. Our research aims to understand complex events described in multimedia inputs by developing a semi-automated system that identifies, links, and temporally sequences their subsidiary elements, the participants involved, as well as the complex event type. This project proposes a systematic analysis of world events, such as the Boston Marathon bombing, Capital Riots, Covid-19, etc. We have successfully evaluated our system on various datasets and have shown significant improvement compared to other previous methods.

1 Introduction

As unstructured multimedia data rises exponentially, fast understanding of world events is essentially a more challenging task. Humans understand events by organizing them into frequently occurring narrative structures. These structures are abstracted as knowledge, the organized units of graphs that represent memory patterns used in human cognition. In general, the previous research work can be divided into two waves. Among them, the first wave of rule-based symbolic reasoning methods (Nguyen & Grishman, 2015; Chen, Xu, Liu, Zeng, & Zhao, 2015), such as conventional information extraction systems that use artificial feature extraction, cannot handle real-world event analysis. At the

same time, the second wave of machine learning or artificial intelligence-based systems (Zhao, Jin, Wang, & Cheng, 2018) requires too many manually generated annotated examples as training data to supervise machine learning so that it cannot meet the actual needs of event understanding.

To solve these challenges, our research project aims to develop a knowledge-oriented Artificial Intelligence (AI) system that can recognize complex events and bring them to users' attention to solve these challenges. Our project seeks to understand the complex events described in the multimedia input by developing a semi-automated system that identifies, links, and sorts its subsidiary elements, involved participants, and complex event types in chronological order. Inquisitive events can produce changes that significantly impact national security or participate in the causal chain that makes such impacts.

Our experiments demonstrate views and analysis of different world events (e.g., Boston Marathon bombing/capital riot/Covid-19, etc.) based on graph-based neural networks' multimedia knowledge graphs. Additionally, results on other public datasets reveal that our proposed system outperforms the other state-of-the-art approaches.

- Our project overwhelms the shortcomings of previous systems. It assembles a knowledge-oriented artificial intelligence system that can identify and analyze these data and complex events and bring them to the user's awareness.
- Our proposed dockerized system can thoroughly comprehend and connect the elements that make up a complex event, which incorporates various data types such as images, text, video, audio, and captions, thus effectively overcoming misinformation.
- Our project successfully analyzes world events like the Boston Marathon bombing, capital riots, and Covid-19 to a level of sophistication. Experimental results on other public datasets illustrate that our proposed system outperforms other state-of-the-art methods.

2 Related Work

In this section we review the previous works done in the Event Detection field and recognizes the problems that comes with most of the solutions.

2.1 Event Detection Approach

Traditional methods to solving this problem consist of extracting solely text-based information. The first wave (Nguyen & Grishman, 2015; Chen et al., 2015) are the rule-based symbolic reasoning methods. They cannot handle real-world problems. The second wave (Zhao et al., 2018) are AI-based systems that require a lot of manually annotated samples which are very time-consuming to annotate. This wave of systems is mostly unusable since they require a large

amount of data to be fed into the system in order to achieve any effect. However, more recent approaches (Ferguson, Lockard, Weld, & Hajishirzi, 2018; L. Huang et al., 2016) to this problem directed more attention towards the multimedia side of information such as images, captions, videos, and audios, thus resulting in an algorithm with a better capability of understanding different types of knowledge. Ones that are only focused on fake news (Fung et al., 2021; Zellers et al., 2019) and do not focus on merging the information to create a knowledge graph of various events. Because most of these algorithms don't consider every source of information possible, it is limited for that reason. In comparison, we propose a more comprehensive approach to this problem by combining semantic features, knowledge graphs, cross-media consistency checking, each offers complementary information, while previous attempts focus on only one or a few of these aspects.

2.2 Event Detection Datasets

In this subsection, we explore previous datasets to reveal their shortcomings. The ACE 2005 dataset (Doddington et al., 2004) is a good starting point for most textual or linguistic-related data work. Yet one shortcoming is that it contains 33 event types, which is quite a small amount, considering that the majority of the event types used have seen almost no annotated instances of use compared to other ones. Moreover, Linguistic Data Consortium's 2015E78 (Consortium, 2005) improves ACE 2005 in terms of the quantity of the entities, yet it is still a subtle improvement. Similarly, TAC-KBP (Ellis, Getman, & Strassel, 2014) (Ellis et al., 2015) (Ellis et al., 2016) improves upon everything ACE 2005 and LDC 2015E78 does, yet it and other Event Detection (ED) such as the ones mentioned above datasets (Yang et al., 2018) shares a similar fate as it only has nine event types and 38 subtypes.

The more recent datasets such as Maven (Wang et al., 2020), however, gives a much more thorough and large amount of event types as it has 168 event types with 4480 Wikipedia documents and 118,732 event mention instances. Yet these datasets all have two shortcomings: 1) they are not multimedia and do not include hot topics into their datasets. 2) Hot topics such as Covid-19, 2021 United States Capitol attack, and Boston Marathon Bombing, which are our main focus points, are not involved in these datasets either. To solve these problems, we created a dataset containing both text and multimedia data with enough event types to accurately assess and train our model.

3 Knowledge-Oriented Multimedia Event Detection System

In this section, we will introduce our dockerized knowledge-oriented multimedia event detection system.

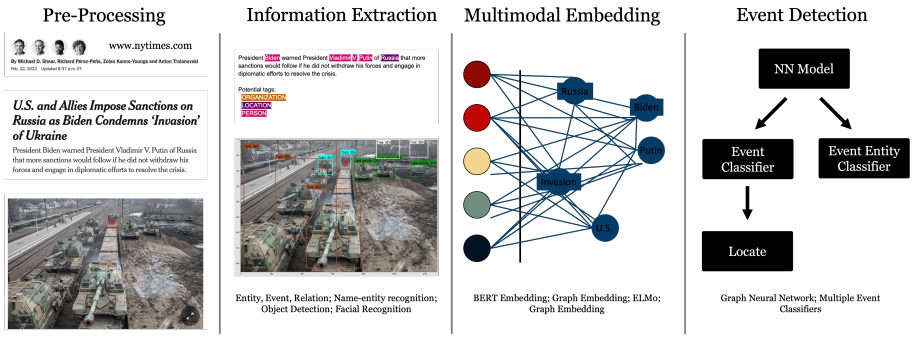

Figure 1: Our system overview divides into four steps consisting of pre-processing, information extraction, multimodal embedding, and knowledge-oriented event detection. These four steps build our foundation of the system thereby granting it the ability to comprehend complex world events such as the Russo-Ukrainian Crisis presented above.

3.1 Overview of System Framework

There are four steps in building the system: 1) Pre-processing, 2) Information Extraction, 3) Multimodal Embedding, 4) Knowledge-Oriented Event Detection, which will be explained in the following paragraphs. In pre-processing, we prepare the unstructured data to optimize the outcome by removing excess information that is in the way, such as punctuations, stopwords, frequent words, and rare words. In information extraction, we analyze the data by turning it into structured knowledge, classifying entities, and identifying objects. In multimodal embedding, we use word embedding, graph embedding, and background embedding to use vectors to link related information together. In Knowledge-Oriented Event Detection, we build a Knowledge Graph (KG) to understand complex world events.

3.2 Pre-processing

With the large amounts of unstructured multimodal data, pre-processing is crucial for effectively optimizing the outcome. There are two general steps that are required for Text, image, and video formats of the input: Text pre-processing and Multimedia pre-processing. We needed to perform extensive text cleaning like tokenizing, removing stopwords, removing punctuations, stemming or lemmatization, and more.

- Text pre-processing involves removing useless information from untouched Text. The process eliminates information such as punctuation, stopwords, and frequent words, and rare words. It also uses the skill of lemminizing and stemming from converting words into their base form, such as changing the word "geese" to "gees" using lemminization and changing "gees" to "goose" using stemming.

Listing 1: Add a new column to text that only has lower case letters

```
df["text_lower"] = df["text"].str.lower()
df.head()
```

Listing 2: Change punctuations in place of spaces

```
PUNCT_TO_REMOVE = string.punctuation
def remove_punctuation(text):
    """custom function to remove the punctuation"""
    return text.translate(str.maketrans('', '',
        PUNCT_TO_REMOVE))
df["text_wo_punct"] = df["text"].apply(lambda text:
    remove_punctuation(text))
```

Listing 3: Remove stopwords in place of spaces

```
STOPWORDS = set(stopwords.words('english'))
def remove_stopwords(text):
    """custom function to remove the stopwords"""
    return " ".join([word for word in str(text).split
        () if word not in STOPWORDS])
```

- For Multimedia pre-processing, keyframe extraction is used on the inputs with an MP4 format to extract the most important frames from the video and convert them into a JPG format. Because most videos have around 60 frames each second, we must eliminate useless information from the video by only detecting large changes within each frame and extracting the significant ones out. Here we use FFmpeg[1] to extract and save the K-Frames in JPG format. Cropping or trimming the video down may also be thumbnails. The said audio from the videos may also be used for voice recognition, but we will work only with the selected frames in this project.

3.3 Information Extraction

After the pre-processing step, we move onto the Information Extraction (IE) step, which uses logical reasoning to analyze unstructured/semi-structured multimedia data to understand and summarize events. There are three steps in this process: 1) Extract Entity, Events and Relations model, 2) Named-Entity Recognition (NER), 3) Object Detection/Facial Recognition.

- We will try to extract Entity, Events, and Relations in this procedure as this is a subsection of Information Extraction work. It turns unstructured text in different domains into structured knowledge. A few challenges to this are overlapping and nested entities and long-ranged dependencies. Inspired by the previous work (Fung et al., 2021), this is usually done with

[1] `https://github.com/FFmpeg/FFmpeg`

pre-trained BERT (Devlin, Chang, Lee, & Toutanova, 2019) with a layer of LSTM (Graves & Schmidhuber, 2005) or fine-tuned BERT, which will be explained later.

- Name-entity recognition is similar to Entity, Events, and Relations as it classifies the named entities in the unstructured text and turns them into predefined categories. Most approaches do it with conditional random fields. Here, we will identify, link, and sort its subsidiary elements, involved participants, and complex event types in chronological order. This process could be broken down into two distinctly different actions: detection of names and classifying these names. These systems could be created by using computer-linguists, which would be more precise at the cost of time, and statistical approaches without supervision at the cost of inputting large amounts of annotated data. Specifically our implementation is based on Stanford's Named Entity Tagger[2].

- Object detection (image localization and image classification) and facial recognition are used for identifying objects and familiar faces. Object detection uses pre-trained CNN and either Tensorflow, PyTorch, or Keras. Facial recognition uses similar things as object detection, with the main difference being that there is holistic processing to identify a face in a specific image. An example of Object Detection is the use of Region with CNN features that involves the steps of: Input the image, Extract the region proposals, Compute the CNN features, and classify the regions. Yet this is relatively slow, so to improve upon this, Faster R-CNN and You Only Look Once (YOLO) are introduced into object detection. Object detection comes with other various forms of tasks such as text detection and text recognition, which would help the system yield better results as more information is fed into it. In our settings, we use Faster R-CNN because the amount of annotated data is limited and because it enables near cost-free region proposals. (Ren, He, Girshick, & Sun, 2016).

3.4 Multimodal Embedding

In word embedding we seek to learn semantically the words' meaning and graph it directly in relation to other words in a document. Multimodal embedding combines both visual and textual information to improve performance. We then use vectors to build this multimodal representation. BERT is primarily used for background embedding, we create word vectors using BERT and ELMo (B-LSTM) and graph embedding though a IE system that will construct a Knowledge Graph. Out of these, the content node used here is BERT.

- BERT (Bidirectional Encoder Representations from Transformers) is a language model that is based off of transformers for NPL usage. With BERT, we the feed words into the BERT architecture which is jointly

[2] `http://nlp.stanford.edu:8080/ner/`

conditioning on both left and right context in all layers. In our research, we use BERT for content and to produce the semantic meaning for each word though a summarizing based BERT encoder which uses a weighted embedding system that would allow the averages of the encoded token embeddings across sentences.

- NPL tasks have reached a higher level due to Google's BERT and ELMo (Birdirectional LSTM). It uses encoders and masked language models to become state-of-the-art-level models. Before moving onto inputting results into the neural network, we first have to put it through a multimodal embedding system, which typically relates a word in text/image/audio to a specific vector that has the meaning of the word so that similar words should be closer together. This involves three steps: Token Embedding, Segment Embedding, and Position Embeddings. When behaving differently token embeddings are just vector representations of words. Segment embeddings are vector representations to show whether the vectors are similar or not. Position embedding will help BERT understand the difference between visually similar but semantically different words. ELMo functions similarly to Word2Vector as it helps us to represent words in embeddings.

- Graph Embedding is created from each multimedia news article. We construct a document leveled Knowledge Graph (KG) through a multimedia Information Extraction (IE) system. The IE system constructs a map of entities that are extracted from text and images. Here u represents a global context node we uses $1|nbr(u)|$ to calculate the credibility with other global context nodes.

$$h_{e_{ev}} = relu(W_t \cdot [h_{n_u}, h_{e_{uv}}, h_{n_v}]) \tag{1}$$

$$h_{n_u} = relu(\frac{1}{|nbr(u)|} \sum_{v \in nbr(u)} h_{e_{uv}}) \tag{2}$$

  Where the first equation uses ELMo's h_{n_u} and h_{n_v} that functions like the edges of the nodes, it connects these two with $h_{e_{uv}}$ and feeds it into a weighted activation network. Then it uses the second equation to draw the graph representing the global context nodes with its neighbors.

3.5 Knowledge-Oriented Event Detection

Knowledge Oriented Event Detection consists of a knowledge-oriented system that incorporates human knowledge to capture the linguistic clues of the relationship between words and phrases. In the previous steps, Building a Knowledge Graph from text data uses sentence segmentation, dependency parsing, parts of speech tagging, and entity recognition to help both machine and us to understand the relationships between entities. After feeding inputs into the

neural network, we build knowledge graphs. Then we have a data visualization framework that allows us to rapidly develop queries and visualizations of the data in the knowledge graphs. A barebone structure consists of two nodes and a relation (edge) in between. In the end, the output result should be a dockerized knowledge graph consisting of world events that both the machine and humans could easily understand. Multiple Relationships could be an issue. Based on multimedia graph feature expression, Semantic features that are useful for identifying causal relations are also created. We then merge the extracted graph features using AVG and MAX pooling. And though a Softmax function, we treat it like a multiple edge classification problem.

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{3}$$

Where, $exp(x_i)$ represents the standard exponential function for input vector with $exp(x_j$ as the standard exponential function for output vector. It helps us to turns a vector of K real values into a vector of K real values that sum to 1.

4 Experiments

4.1 Experimental Settings

Analyzing global events is a large part of our experiment. To reasonably evaluate the performance of our proposed model, followed by the previous work, we collected Wikipedia articles describing complex events on the Boston Marathon bombing, Capital Riots, and Covid-19 to evaluate our system in event extraction and tracking tasks. Testing events such as Covid-19 will give us a deep analysis of what this specific event is about and benefit us greatly as it would be straightforward to understand. As another part of our experiment, we run our system through the MAVEN (Wang et al., 2020) dataset and compare the results to previous state-of-the-art approaches.

To finish this experiment, we chose MAVEN and our dataset. MAVEN (Wang et al., 2020) is the most recent and has a large number of event types, totaling 168 event types, 4480 Wikipedia documents, and 118,732 event mention instances. For the second choice of the dataset, we selected 120 videos from the internet and processed them through our system. We use accuracy to test our system by randomly selecting 100 documents to process, and out of that 100, it will determine the percentage.

4.2 Compared with the state-of-the-art methods

We compare our methods with the following baselines. 1) DMCNN (Zhang, Yang, Hu, & Liu, 2018) uses a sliding window with CNN embedding. 2) BiLSTM (Zhou et al., 2016) uses Bidirectional Long-short term memory. 3) BiLSTM+CRF (Z. Huang, Xu, & Yu, 2015) uses BiLSTM and conditional random field. 4) MOGANED (Yan, Jin, Meng, Guo, & Cheng, 2019) uses Multi-Order

Table 1: Compared with the state-of-the-art methods

Method	P	R	F-1
DMCNN	66.3±0.89	55.9±0.50	60.6±0.20
BiLSTM	59.8±0.81	67.0±0.76	62.8±0.82
BiLSTM+CRF	63.4±0.70	64.8±0.69	64.1±0.13
MOGANED	63.4±0.88	64.1±0.90	63.8±0.18
DMBERT	62.7±1.01	72.3±1.03	67.1±0.41
BERT+CRF	65.0±0.84	70.9±0.94	67.8±015
Ours	**72.0±0.85**	**74.6±0.90**	**73.3±0.5**

Graph Convolution and Aggregated Attention. 5) DMBERT (Wang, Han, Liu, Sun, & Li, 2019) uses adversarial training for weakly supervised event detection and BERT+CRF (Devlin et al., 2019; Lafferty, McCallum, & Pereira, 2001) use BERT and conditional random field.

Experimental results are listed in Table 1. Comparing BiLSTM to BERT+CRF, we can see a vast increase of scores as it shows that BERT+CRF is more effective than BiLSTM, but our model has the best results overall. In general, we can see an increase in F-1 scores. Our method combines both BERT and graph neural networks, resulting in better performance. Because our model is very sophisticated, the plus-minus is varied. Our dataset shows that our method accuracy is 75.4, the recall is 80.2, and the F-1 is 77.7, respectively.

4.3 Discussion and future works

We've provided a clear and novel system for complex event analysis, yet there is much more to be done in this field. We will be looking to improve this system as well. In the long term, we hope to extend our approaches to cover a broader range of sources as well as comprehend more languages than just English alone, as that can allow us to access Future works may involve improving the F-score and making the system analyze newer and more sophisticated results and new types of data.

5 Conclusion and acknowledgement

In conclusion, we have provided a clear and systematic way of analyzing real-world events like the Boston Marathon bombing, capital riots, and Covid-19 through multiple data sources and dockerizing the system to connect elements that make up a complex event. Our framework can also be used in various circumstances with the F1 scores better than other state-of-the-art systems. We are very thankful for CMU's Linux servers and various publicly available classes that are up on Youtube, as they've helped us a lot through the basics of NLP.

References

Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015). Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of acl-ijcnlp* (pp. 167–176). Retrieved from `http://aclweb.org/anthology/P15-1017` doi: 10.3115/v1/P15-1017

Consortium, L. D. (2005). ACE (Automatic Content Extraction) English annotation guidelines for events. *Version*, *5*(4).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacl-hlt* (pp. 4171–4186). Retrieved from `https://www.aclweb.org/anthology/N19-1423` doi: 10.18653/v1/N19-1423

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., & Weischedel, R. (2004). The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of LREC*. Retrieved from `http://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf`

Ellis, J., Getman, J., Fore, D., Kuster, N., Song, Z., Bies, A., & Strassel, S. M. (2015). Overview of linguistic resources for the TAC KBP 2015 evaluations: Methodologies and results. In *Tac.*

Ellis, J., Getman, J., Fore, D., Kuster, N., Song, Z., Bies, A., & Strassel, S. M. (2016). Overview of linguistic resources for the TAC KBP 2016 evaluations: Methodologies and results. In *Tac.*

Ellis, J., Getman, J., & Strassel, S. M. (2014). Overview of linguistic resources for the TAC KBP 2014 evaluations: Planning, execution, and results. In *Tac.*

Ferguson, J., Lockard, C., Weld, D., & Hajishirzi, H. (2018). Semi-supervised event extraction with paraphrase clusters. In *Proceedings of naacl* (pp. 359–364). Retrieved from `https://www.aclweb.org/anthology/N18-2058` doi: 10.18653/v1/N18-2058

Fung, Y., Thomas, C., Reddy, R. G., Polisetty, S., Ji, H., Chang, S.-F., ... Sil, A. (2021). Infosurgeon: Cross-media fine-grained information consistency checking for fake news detection. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 1683–1698).

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, *18*(5-6), 602–610.

Huang, L., Cassidy, T., Feng, X., Ji, H., Voss, C. R., Han, J., & Sil, A. (2016). Liberal event extraction and event schema induction. In *Proceedings of acl* (pp. 258–268). doi: 10.18653/v1/P16-1025

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, *abs/1508.01991*. Retrieved from `http://arxiv.org/abs/1508.01991`

Lafferty, J., McCallum, A., & Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Nguyen, T. H., & Grishman, R. (2015, July). Event detection and domain adaptation with convolutional neural networks. In *Proceedings of acl* (pp. 365–371). Retrieved from `https://www.aclweb.org/anthology/P15-2060` doi: 10.3115/v1/P15-2060

Ren, S., He, K., Girshick, R., & Sun, J. (2016). *Faster r-cnn: Towards real-time object detection with region proposal networks.*

Wang, X., Han, X., Liu, Z., Sun, M., & Li, P. (2019). Adversarial training for weakly supervised event detection. In *Proceedings of naacl* (pp. 998–1008). Retrieved from `https://www.aclweb.org/anthology/N19-1105` doi: 10.18653/v1/N19-1105

Wang, X., Wang, Z., Han, X., Jiang, W., Han, R., Liu, Z., ... Zhou, J. (2020). Maven: A massive general domain event detection dataset. *arXiv preprint arXiv:2004.13590*.

Yan, H., Jin, X., Meng, X., Guo, J., & Cheng, X. (2019). Event detection with multi-order graph convolution and aggregated attention. In *Proceedings of emnlp-ijcnlp* (pp. 5766–5770). Retrieved from `https://www.aclweb.org/anthology/D19-1582` doi: 10.18653/v1/D19-1582

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., & Manning, C. D. (2018). HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of emnlp* (pp. 2369–2380). Retrieved from `https://www.aclweb.org/anthology/D18-1259` doi: 10.18653/v1/D18-1259

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2019). Defending against neural fake news. *Advances in neural information processing systems*, *32*.

Zhang, X., Yang, W., Hu, Y., & Liu, J. (2018). Dmcnn: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *2018 25th ieee international conference on image processing (icip)* (pp. 390–394).

Zhao, Y., Jin, X., Wang, Y., & Cheng, X. (2018). Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of acl* (pp. 414–419). Retrieved from `https://www.aclweb.org/anthology/P18-2066` doi: 10.18653/v1/P18-2066

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., & Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 207–212).