We hypothesize that **knowledge-oriented multimedia data** can help create an accurate analysis of events. Our research program analyzes complex events to understand social hotspots. The outcome of our research is to propose a Dockerized knowledge-oriented multimodal event detection system.

The main goal of our research is to understand complex events described in multimedia input by developing a system that has the ability to **identify**, **link** and **sort /** **elements**, **participants** and **event** types in chronological order.

We propose a **systematic** analysis of world events such as the Boston Marathon bombing, Capital Riots, Covid-19, etc.

We build this on a **dockerized system** as it can both **emulate** the target platform and **cross-compile** through multiple scenarios. With the nec

There are 4 key steps to building the system. 1. Pre-processing, 2. Information Extraction, 3. Multimodal Embedding, 4. Knowledge-Oriented Event Detection.

With the large amounts of unstructured multimodal data today, pre-processing is crucial for effectively optimizing the outcome. There are two general steps that are required: 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 for texts. For Multimedia pre-processing, we need to use FFmpeg to extract the keyframes from mp4 files. And with the extracted images, we can move onto the next step. We understand “we reason by constructing a mental model of the situation”.

After preprocessing is Information Extraction (IE), which uses logical reasoning to analyze unstructured or semi-structured multimedia data to understand and summarize events. There are three steps in this process: 1) Extract Entity, Events and Relations model using pre-trained BERT, 2) Named-Entity Recognition using Stanfords’ name entity tagger (NER), 3) and Object Detection or Facial Recognition, using Faster-CNN for it’s cost free region proposals.

In multimodal 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). We build graph embedding through an IE system that will construct a Knowledge Graph.

Event Detection consists of a knowledge-oriented system that incorporates human knowledge to capture the linguistic clues of the relationship between words and phrases. Building a Knowledge Graph from text data uses sentence segmentation, dependency parsing, parts of speech tagging, and entity recognition 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. 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.

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 dataset and compare the results to previous state-of-the-art approaches.

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. Our dataset shows that our method accuracy is 75.4, recall is 80.2, and the F-1 is 77.7, outperforming most other state-of-the-art systems.

For future work we hope to share a demo of this system publically on a website as well as comprehend more languages than just English alone.

We are very thankful for CMU’s linux servers as well as various different publically available classes that are up on Youtube as they’ve helped us a lot through the basics as well as their website.

Transformers and BERT:

Generally speaking, BERT for applications like NER is that the authors (of whichever implementation you use) performed a large-scale pretraining effort to create the embeddings. You can then “fine-tune” those for your specific task using far less computation, but the rub is that you need to use the same tokenization scheme (I.e. the BERT Tokenizer) in order to have your input “fit” the existing embeddings. Intuitively, tokenization is mapping a word in your text to an index number. If the embedding was trained thinking that word number 42 is “cat” then things won’t work well if you tokenize differently and provide a 43 instead when “cat” pops up in your text.

With the effective description of “All is silent on the Eastern front”, it propels the white and black movie into the life and death of the soldiers who all had some sort of internal conflict. The graphic images were incredibly crucial to the experience as it gives the images inherent meaning.

“To me, war was the greatest sin

### What is a transformer?

The original transformer is an encoder-decoder-based neural network that is mainly characterized by the use of the so-called [attention](https://arxiv.org/pdf/1902.02181.pdf) (i.e. a mechanism that determines the importance of words to other words in a sentence or which words are more likely to come together) and the non-use of recurrent connections (or recurrent neural networks) to solve tasks that involve sequences (or sentences), even though RNN-based systems were becoming the standard practice to solve natural language processing (NLP) or understanding (NLU) tasks. Hence the name of the paper "Attention is all you need", i.e. you only need attention and you don't need recurrent connections to solve NLP tasks.

Both the encoder and decoder are composed of

* attention modules
* feed-forward (or fully connected) layers
* residual (or skip) connections
* normalization layers
* dropout
* label smoothing
* embedding layers
* positional encoding

The decoder part is also composed of a linear layer followed by a softmax to solve the specific NLP task (for example, predict the next word in a sentence).

### 

### 

### 

### BERT:

BERT stands for Bidirectional Encoder Representations from Transformers, so, as the name suggests, it is a way of learning representations of a language that uses a transformer, specifically, the encoder part of the transformer.

* BERT is a language model, i.e. it represents the statistical relationships of the words in a language, i.e. which words are more likely to come after another word and stuff like that. Hence the part Representations in its name, Bidirectional Encoder Representations from Transformers.  
    
   BERT can be trained in an unsupervised way for representation learning, and then we can fine-tune BERT on the so-called downstream tasks in a supervised fashion (i.e. transfer learning). There are pre-trained versions of BERT that can be already fine-tuned and used to solve your specific supervised learning task.  
    
   On the other hand, the original transformer was not originally conceived to be a language model, but to solve sequence transduction tasks (i.e. converting one sequence to another, such as machine translation) without recurrent connections (or convolutions) but only attention.
* BERT is only an encoder, while the original transformer is composed of an encoder and decoder. Given that BERT uses an encoder that is very similar to the original encoder of the transformer, we can say that BERT is a transformer-based model. So, BERT does not use recurrent connections, but only attention and feed-forward layers. There are other transformed-based neural networks that use only the decoder part of the transformer, for example.
* BERT uses different hyper-parameters than the ones used intransformers to achieve the best performance.
* BERT also uses segment embeddings, while the original transformer only uses word embeddings and positional encodings.

Why am I interested in this project?

1. I was intrigued by the global events such as Covid and how hard it hit the world. And within less than 5 months into developing this system as of right now, Russia wanted to start a war with Ukraine. This project will be incredibly helpful in terms of both information processing and fake news detection.
2. NLP is very interesting for me as I view it as a challenge on a daily basis since there is so much excess information in the world that we need something to process them all through to get rid of the useless / fake ones.
3. I enjoy coding and have a background in competitive coding. I wanted to try something new such as artificial intelligence. I wanted computers to understand multimedia information as it would be really cool to create an all knowing computer that spits out the answer “42” as the answer to the universe.

What is Docker?

1. Docker is a set of platforms that use OS-level virtualization to deliver software in packages called containers. You do not need to implement any sort of pre-requisites and can use the projects right away.

Why use graph embedding?

1. Graph embedding is used for drawing out a map of different information as it is very useful for multimodal information processing. The different nodes contain things such as text and image, and with the equation listed below we are able to draw the graph representing the global context nodes and its neighbors.

This project can…

1. Create an easily accessible info chart, helpful to understanding for those who don't have the time to do hours upon hours of research themselves and want the gist of the event.
2. Detect fake news or propaganda by comparing and contrasting.
3. Predict results of certain events that would help strategically.

What did I get out of this project…

1. I got way more information about Artificial Intelligence, RNN, CNN, BERT, LSTM and various different algorithms, Python etc, and how it applies to the real world such as search engine results, recommendations etc.
2. Because this project took several months to complete, I had to manage my time properly between school work and extracurricular activities and sports that I am doing.
3. I have a clear view of my goal and what I want to do in the future as well as colleges that I think I am going to apply to.

For hardware, we use 4 2080 GPU from Nvidia and 64gb ram.