Social Media Monitoring

Technische Universität München

Aysenur Kocak, Youssef Lotfy, Leonhard Stengel

Supervised by Ahmed Mosharafa, Prof. Dr. Georg Groh

Social Computing Group, Department of Informatics

Background

Automated Journalist

Problem Statement: Automatically generate news content from structured data with minimal human interference.

Our Contribution: Social Media Monitoring

Goal: Implement a feature for detecting relevant events on social media.

Method: Developed using EDTBERT by Pradhan et al. (2024) [3]

Methodology

1) Preprocessing: We cleaned tweets by removing stopwords, punctuations, web links, non-ASCII characters, retweets, and converting text to lowercase. We divided tweets into fixed-sized time windows.

2) Graph Construction: We created tweet representations as nodes in a graph with edges denoting similarities based on the combination of structural and contextual embeddings.

2.a) Contextual Embeddings: To get contextual embeddings, we used Sentence Transformers library from HuggingFace. Tweets were embedded in a vector space such that similar tweets are close and can be found using cosine similarity. In our work, we used the model "all-mpnet-base-v2".

2.b) Structural Embeddings: We derived structural information from overlapping hashtags and named entities (NEs) within tweets, which indicates the formation of cliques where tweets are interconnected due to their association with a common event. To determine the structural association between nodes (tweets) in the graph, we calculated the degree of overlap in hashtags and named entities between tweets using the Jaccard Similarity measure.

5) Graph Clustering: Constructed graphs are then passed to the event detection module. To detect clusters of related tweets, we used the Markov Clustering (MCL) method. MCL is a graph clustering algorithm that groups nodes based on their connectivity. It simulates random walks within a graph to identify clusters by iteratively expanding and inflating the graph's adjacency matrix. The process emphasizes densely connected regions, effectively separating clusters without needing a predefined number of clusters. In this project, we used "markov-clustering" library from Python.

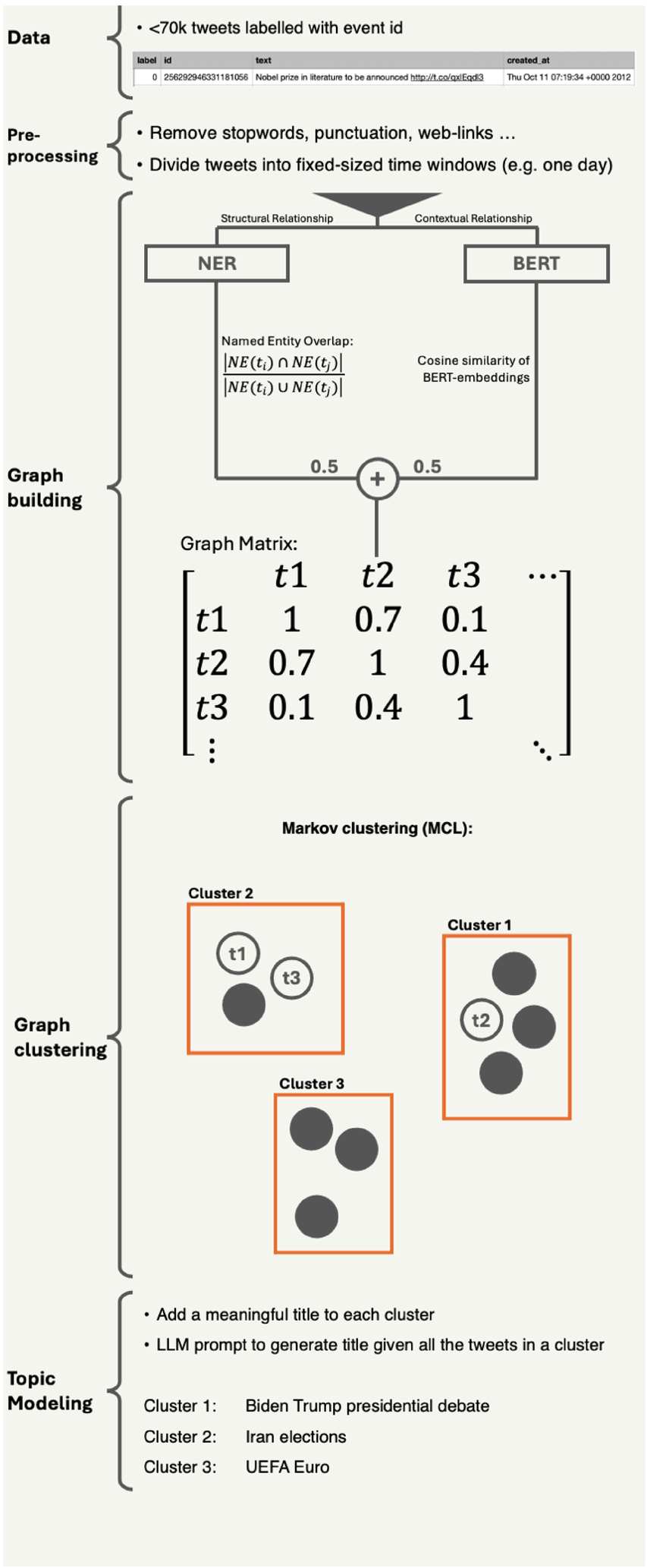

Fig. 1. Social Media Monitoring Workflow Diagram

6) Event Summarization: To generate concise summaries for each cluster, we employed a large language model (LLM). We created prompts based on the extracted top keywords from the clusters and generated cluster summaries & titles using Google's FLAN-T5 base LLM. Summarization effectively provided insights into detected events, and it improved the ability of our system to generate news with little intervention from human side.

Literature Review

Clustering-Based Event Detection

In clustering-based event detection, tweets are grouped into clusters using an online clustering algorithm. The online clustering algorithm evaluates the similarity of incoming tweets with existing clusters. Once the clusters are formed, these clusters are analyzed using various features (temporal, social, topical) to distinguish between event-related and non-event related clusters. Then, a classifier is applied to predict which clusters correspond to real-world events [1].

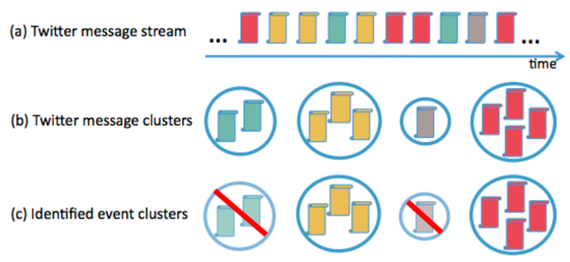

Fig. 2. Conceptual diagram: Twitter event identification [1]

Graph-Based Event Detection

In graph-based event detection, tweets are represented as nodes in a graph, where edges between nodes indicate relationships or similarities, such as shared hashtags or keywords. By analyzing the structure and connectivity of this graph, clusters or communities of related tweets can be identified, which are likely to correspond to events. This method leverages the network properties of the data to detect and track events based on the patterns of interactions and connections between tweets [2].

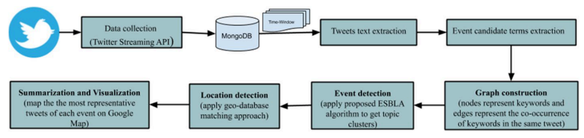

Fig. 3. A workflow diagram in a graph-based event detection module [2]

Event Detection Using Contextual Embeddings

Tweets are represented as high-dimensional vectors that capture the semantic context of their content using pre-trained language models (e.g., BERT). These embeddings provide nuanced representations of tweets that allows for the detection of events based on the contextual similarity between tweets. This method enhances the accuracy of event identification by taking the deeper meaning and relationships within the tweet data into account [3].

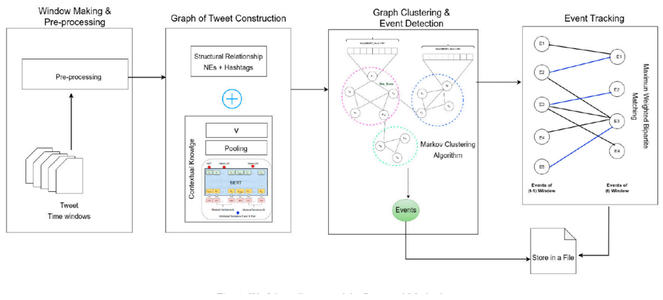

Fig. 4. Workflow diagram of EDTBERT [3]

Results

Our goal is to detect events on social media and cluster them based on contextual and structural similarity. The results are then shown in the form of a hot topics section showing the cluster titles assigned by the FLAN-T5 model. Our results have a **precision of 85%**, a **recall of 70%** and an **F1-score of 77%** on the Event2012 Dataset. Furthermore, the dashboard contains cluster visualizations which has the query used to detect the cluster as a central entity. Each cluster is visualized as a circle which has a radius based on the cluster size and a position relative to the central query based on the cosine similarity between the embeddings of the cluster's title and the query.

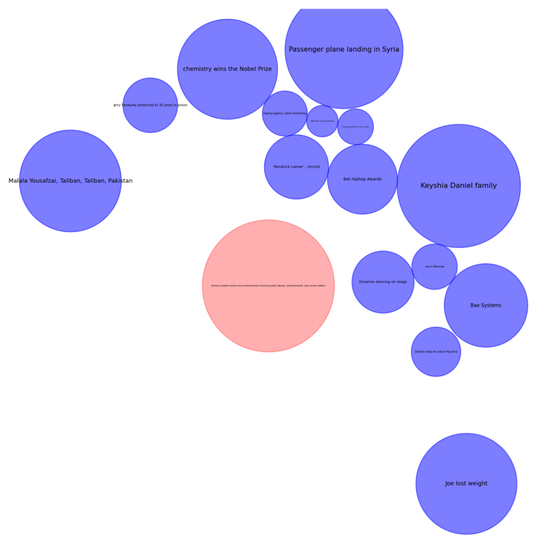

Fig. 5. Cluster Visualization for Event2012 Dataset

Conclusions

Event detection on social media platforms is a dynamic and rapidly evolving research area with significant implications across various fields. Identifying events from tweets provides valuable insights and supports critical decision-making processes. EDTBERT effectively identifies unspecified events in the Twitter stream by leveraging the structural relationships and contextual representations of tweets. By generating a graphical representation of tweets and utilizing a pre-trained BERT sentence transformer, we captured both the interconnectedness and contextual associations among tweets. Our method employed graph clustering to find semantically similar tweet clusters. Results show that our approach effectively recognizes and monitors events.

Key Achievements:

- Accurate Event Detection
- Robust Performance

References

[1] Becker, Hila & Naaman, Mor & Gravano, Luis. (2011). Beyond Trending Topics: Real-World Event Identification on Twitter. ICWSM. 11.

[2] Singh, Jagrati & Pandey, Digvijay & Singh, Anil. (2023). Event detection from real-time twitter streaming data using community detection algorithm. Multimedia Tools and Applications. 83. 1-28. 10.1007/s11042-023-16263-3.

[3] Pradhan, Mohanty & Lal. (2024). EDTBERT: Event Detection and Tracking in Twitter using Graph Clustering and Pre-trained Language Model. Procedia Computer Science. 233. 481-491. 10.1016/j.procs.2024.03.238.