Early Event Detection and Report Generation from Heterogeneous Data Sources

Aravind M S[1], Shiyas K[2], Suresh Babu K[3], Prof. Linda Sara Mathew[4]

[1]Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam
[2]Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam
[3]Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam
[4]Professor, Department of Computer Science and Engineering, Mar Athanasius College of Engineering, Kothamangalam

---

Abstract - *People use online social networks to express their opinions and feelings about many topics, and these online social networks witness the first occurrence of various events like disease outbreaks, political events, natural calamities, etc. directly from the users around the globe. To get a better view of situations all over the world, a highly optimized event detection system is necessary. Based on data collected from heterogeneous data sources, the event detection and report generation system is designed by making use of the high-end machine learning models over Django capable to detect the various events happening at a particular location and generating a brief description about the same. The overall system runs with the support of three machine learning models. Clustering is done using the K-means clustering algorithm, LSTM recurrent neural network is applied for event detection, and report generation is done using the gpt2 model. Barely any good work has been done so far to properly identify various events occurring in various places.*

Key Words**: Machine learning, Clustering, event detection, Heterogeneous data, lstm**

1. INTRODUCTION

The traditional media predominantly covers general events and thereby targets a vast audience. Events that target a minority of people are reported very rarely. Social media platforms such as Instagram, Facebook, and Twitter are popular sources of information as well. Since these social media contain semi-structured and unstructured data, extracting valuable and structured information from these can be challenging. So it is not a good method to use social media platforms alone for event detection.

In the past, while performing event identification tasks in social media posts, researchers primarily relied on textual attributes as their primary source of data. Other distinguishing characteristics, such as the post's timestamp, user behavioral habits, and geolocation, have been effectively taken into account in addition to the content itself. However, we propose that the event identification algorithm be driven by semantic information at the tweet level. After all, semantically distinct events, as well as the tweets associated with them, are more likely to be distinguished than semantically connected occurrences. For example, it is pretty straightforward to discern between tweets about a sporting event and tweets about a concurrent political argument.

The use case we discuss in this paper is the segmentation of heterogeneous data into discrete events. The tweets in this collection are linked to a certain event, and it's our responsibility to group them together so that the concealing event structure is replicated in these clusters. For this, we employ a one-pass clustering approach.

1.1 Event detection system

Event detection is the process of detecting real-time events from a particular location based on matching patterns of an event type. Event kinds are defined by event patterns and circumstances. Subscribers to an event type should be notified if a set of events matching the pattern of the event type is detected during the analysis. Filtering and aggregation of events are usually part of the analysis. As the name indicates, the event detection system detects real-time events like disease outbreaks, sports events, political events, etc. happening at a particular location. After collecting the real-time tweets from Twitter using Twitter API and real-time news from a particular location using news API, the dataset is passed through two machine learning models for clustering and event detection based on the related semantic topics of the heterogeneous data. With the help of event detection systems, early cures or solutions can be found for many unforeseen events like the covid-19 pandemic and other events that are potentially harmful to the human race. Words are frequently treated as signals in existing event detection algorithms. Words in the time domain are the subject of some studies.

2. RELATED WORK

2.1 Real-time event detection from the Twitter data stream using the TwitterNews + Framework

This paper [1] focuses on the detection of events from Twitter data. This paper proposes an event detection system that uses a combination of specialized inverted indices and an incremental clustering approach to locate both major and minor noteworthy events in real-time from the Twitter data stream at a cheap computational cost. In addition, execution of a detailed parameter sensitivity analysis to fine-tune the settings in TwitterNews+ for optimal performance. The evaluation is based on the system's effectiveness, with a publicly available corpus serving as a benchmark dataset. The evaluation's findings demonstrate a considerable improvement in recall and precision when compared to five state-of-the-art baselines. This paper used only Twitter as a source of data for event detection which may not give 100% genuine results. To get a more realistic view of the various events happening all over the world, we need to fetch data from different sources. The TwitterNews+, Showed good performance compared to the other baseline models. When it comes to Twitter alone, this produces good results. But 100% genuine results cannot be obtained from analyzing Twitter alone.

2.2 Event detection using wavelet hashtag signal analysis

In this paper [2] Twitter event recognition was supposed to be based on signals derived from Hashtag count mentions. They were also said to illustrate the progression of Twitter trends. It uses a technique called map to reduce which generates one signal concerning each hashtag mentioned in the time interval defined. Wavelets analysis is a popular method of signal processing that detects changes and peaks in signals. The time-frequency representation is carried through a method called continuous wavelets transformation (CWT) and this can offer good localization for time and frequency. The events in the Twitter stream are detected using two wavelet tools. The hashtag signal peaks are detected by using peak analysis and the changes in the hashtag signal are detected by local maxima detection. Due to the noisy characteristics of the tweet stream, there can be hashtag signals with high variance between successive time intervals. The wavelet hashtag signal analysis also gives importance to the Twitter data. And the event detection algorithm was developed in the R programming language and the database used was MongoDB using the resulting JSON document.

3. PROPOSED SYSTEM

This is an event detection and report generation model based on three machine learning models and the objective is to detect real-time events from heterogeneous data sources. Twitter [3] and news API [4] are the two heterogeneous data sources that are used to collecting data, in real-time. The overall system is divided into four modules. The first module is the data collection part in which the data collection from heterogeneous data sources is performed. After collecting the data, they are clustered using a k-means clustering algorithm. The data collected will be in a semi-structured or unstructured format, so it must be converted to the structured format before clustering.

After clustering, the event detection is performed. In our system, event detection is done using the lstm neural network and this is the third module. Finally, a brief description of the detected events is generated.

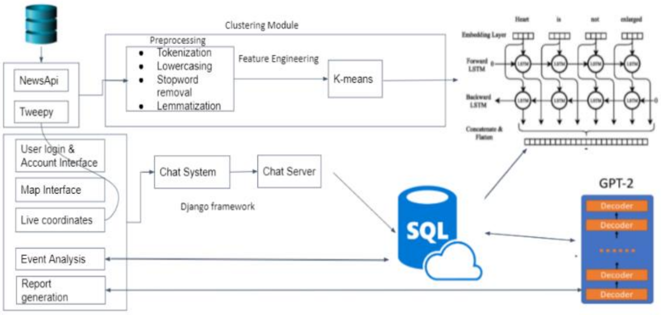

Fig 1: Architecture diagram

3.1 Data Collection and Clustering

The proposed system consists of collecting, cleaning, preprocessing the data, clustering, and implementing the models for event detection and report generations. Figure 3 represents the overview of this system.

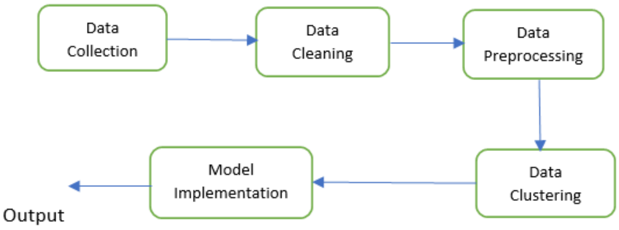

Fig 2: Data Clustering Steps

A. Data Collection

For training, the machine learning model datasets were collected from Kaggle. For event detection, the dataset collected contains the topics disease, sports, and political events respectively. The datasets contain an id, text, date, location columns. For report generation, the model is fine-tuned using a dataset containing news report samples. This dataset contains id, location, date, text columns. In the application, side lives data collection is done using two APIs known as Tweepy and News API. Tweepy is used to collect data from Twitter and News API used to collect news data from various news resources.

B. Data Cleaning

In the data cleaning part basically, the unwanted data is removed from datasets. It includes removing or replacing unwanted columns and empty rows or values and it can be done using the pandas library. Each of the data should be numerically labeled for the model prediction part. For data processing, short-length data are better, because it provides good speed. NLTK library provides many features for data cleaning like stop words removal. After data cleaning, it can be passed to preprocessing part.

C. Data Processing

Machine learning models basically process numerical data. So the text data collected should be converted to a numerical format. Since the LSTM event detection model is a multiclass classification problem the labels should be encoded using the Sklearn library LabelEncoder. For the training and testing part of the model, the dataset should be divided into training and testing sets. Then the text data is tokenized to convert it into the numerical format and padded data so that all data can have the same length at the time of processing. For training the Gpt2 report generation model the data is given in the format of a text file containing news headlines and corresponding text reports.

D. Data Clustering

The data clustering part is mainly included to remove unwanted data collected using the APIs. This will reduce the processing time on the application side. Kmeans clustering works by assigning data points to the cluster centroids. In k means initially cluster number can be specified. K means using Tf-Idf is the method implemented here in the clustering module. The Tf-Idf (term frequency-inverse document frequency) is a weight that ranks important words in a document which helps in assigning words to each category based on the frequency of words. Using Tf-Idf, top words per cluster can be easily detected.

3.2 Event Detection Model

LSTM is a recurrent neural network that has a memory module and it can remember long-term dependencies or patterns. LSTM has better performance for text classification and prediction compared to other classification models. Here we are using a variation of LSTM called Bidirectional LSTM which has forward and backward processing LSTM units. An LSTM network consists of different cell blocks having different functions. Two different states are passing between cells, they are called hidden state and cell state. The memory blocks are the modules that help the model to remember the patterns and this memory is modified with the help of a module known as gates.

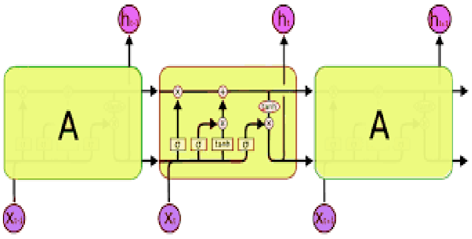

Fig 3: LSTM model

For removing data from cell state a forget cell is implemented. The information which has no use in LSTM is removed using multiplication of filter and it optimizes the performance of LSTM. The data is added to the cell state with the help of the input gate. Output gate is used as a tool for showing current cell data. A Bidirectional LSTM consists of two LSTM which have forward and backward input processing modules. Bi-LSTMs effectively increase the amount of information available to the network, improving the content available to the algorithm. Here the Lstm model is imported using the TensorFlow Keras library.

i) The softmax classifier is used for multi-class classification purposes.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad (1)$$

where the Z_i is an input vector that can take any real value. The denominator ensures that all the output values of the function will sum to unity.

ii) The optimizer used with the LSTM model is Adam, which is a replacement optimization algorithm. AdaGrad and RMSProp are the algorithms that provide the best features to Adam which can handle noisy problems with sparse gradients.

iii) Here the loss function implemented is categorical cross-entropy. Categorical cross-entropy loss function use in multi-class classification. The loss function determines the error in multi-classification tasks.

$$\text{CCE}(p, t) = -\sum_{c=1}^{C} t_{o,c} \log(p_{o,c}) \quad (2)$$

Where, C is the number of classes and t, o, c, p are the observations used in the computation

3.3 Report Generation Model

The decoder part of the Transformer architecture is used in the Gpt2 which is used for generating text. The full version of the Gpt2 model is not released due to the chance of using the model features illegally for many purposes. In the proposed system the pre-trained model of Gpt2 is used. For training purposes mainly we are fine-tuning the GPT2 model using a news report dataset. The heading of news will be given as input and the corresponding report has to be generated by the model. Using the PyTorch and transformers libraries we can import the gpt2 model.

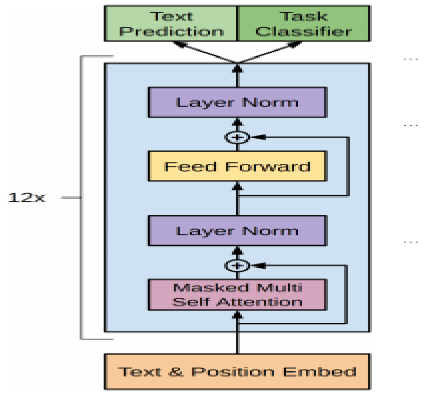

Fig 3: GPT2 Architecture

4. EVALUATION OF PROPOSED FRAMEWORK AND ITS ANALYSIS

This chapter discusses the results of the experiments that are conducted in implementing early event detection and report generation system, Also the metrics used to measure the performance of machine learning models, and we will be discussing existing methods and our methods used in this project.

4.1 Comparison of Cluster models

In our analysis part, we have analyzed mainly two cluster algorithms Kmeans and Dbscan. Theoretically, Kmeans have higher performance over large datasets compared to Dbscan. But we have to prove it practically.

K-means is a centroid-based or segment-based bunching calculation. This calculation partitions every one of the focuses in the example space into K gatherings of likeness. The similitude is for the most part. The similarity is usually measured using Euclidian Distance. The similitude is generally estimated utilizing Euclidian Distance. DB-Scan is a thickness-based bunching calculation. The key truth of this calculation is that the neighborhood of each point in a group that is inside a given sweep (R) should have a base number of focuses (M). This calculation has demonstrated incredible proficiency in identifying exceptions and taking care of commotion. In our projects, if we use K-means, we enjoy the benefit of proclaiming the group number remotely.

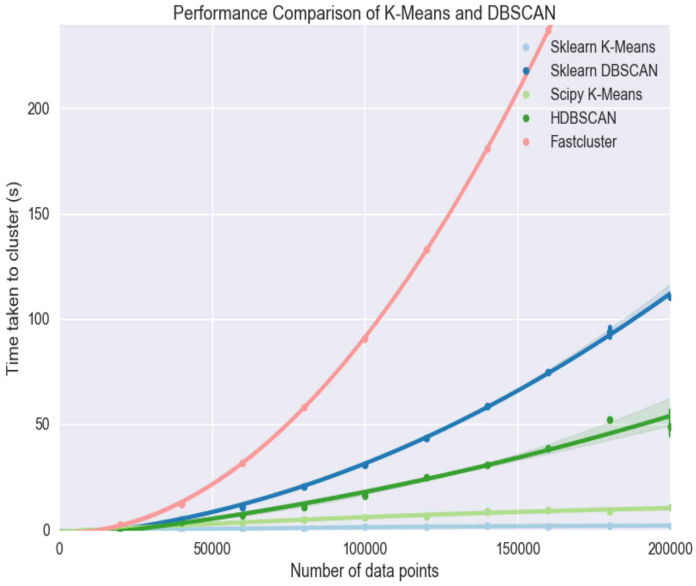

Fig 4: Performance evaluation of Clustering (K-means vs Dbscan)

Here we have taken the three datasets used for LSTM model training. We are trying to Cluster the data into three categories using Kmeans and Dbscan. In the above figure, we can see that Kmeans has clustered into 3 clusters of data whereas Dbscan has a single cluster. Our dataset contains a High amount of football data football, then comes the election. From the Kmeans Graph, it can be seen that it has almost accurately classified the dataset.

In K-means we have used Tf-idfVectorizer for the feature extraction part. Which is extract features based on the frequency of words belongings to each category.

After feature extraction, Kmeans Finds the Clusters along with top words per cluster. In Dbscan we use the elbow method for finding the right Eps hyperparameter. Based on that Dbscan model is trained. From the evaluation, we found that Dbscan has performed poorly on a large dataset by clustering data into one cluster. Whereas the K-means Clustering method created 3 clusters corresponding to each category accurately.

4.1 Comparison of Text classification models

In our method, we have chosen the LSTM model because it has the advantage of remembering long-term dependencies is better than RNN because it avoids the vanishing gradient problem. It is problem in which the weight update value remains constant as the size of the network increases. From the literature survey we conducted, we found below information

GRU (Gated recurring units) is another model which is also a solution to the vanishing gradient problem. But it has lower performance over a large dataset. In our project, we are collecting data from more than one site, so there will be a large amount of data. When it comes to GAN(generative adversarial network) it is also the latest model but it is mainly developed for image and video classification when it uses for text classification it has a more computational cost and poor performance compared to LSTM.

There are other text classification models including SFPM (soft frequent pattern mining), BNgram, SVM (support vector machines). From the analysis of these models, it is found that SVM has the highest accuracy around 88-90% compared to other models. SFPM and BNgram models are based on the frequency of the words in a class but as the number of words increases or the words to which the pattern should compare increases, these models perform less efficiently. An SVM model is a depiction of the models as centers in space, arranged so the examples of the extraordinary classes are secluded by a sensible opening that is pretty much as wide as could be anticipated. New models are then arranged into that comparable space and expected to have a spot with a class reliant upon the side of the opening on which they fall. Its weights are it requires full checking of data and it is directly suitable to 2 class tasks. LSTM performs better contrasted with SVM in all of the circumstances. This is an immediate aftereffect of its ability to review or neglect to recall the data in a useful manner than SVM.

4.2 Comparison of Text Generation models

LSTM, GAN, and GPT-2 are some of the models utilized for language generation. We discovered the following information through our literature review:

The Long-Term Dependency Model (LSTM) is a recurrent neural network that can detect and train long-term dependencies. GPT-2, on the other hand, is a huge language model with 1.5 billion parameters that was trained on a large dataset.8 million web pages in a database. According to research, GPT-2 is simpler, easier, and faster to utilize than the LSTM network. It generates text that may easily be mistaken for the real thing. In GPT2, the quality of the output text is also closely proportional to the quality of the data used to train the network. As a result, if we want better outcomes, we need to spend more time cleaning and correcting our data. Unsupervised learning is made possible by the employment of Generative Adversarial Networks, a powerful family of neural networks. When GAN and GPT2 experiments are compared, it is clear that GAN's text is significantly worse than GPT2. To improve the efficiency of event detection, data should be collected to the greatest extent possible. To do so, we must also have access to unstructured and semi-structured data. For that purpose, we are utilizing heterogeneous data sources.

When we rely on models like LSTM, we must collect a large number of datasets, which takes a long time to construct an efficient model like GPT2, which has been trained on a large dataset and has better performance.

5. CONCLUSION

Early event detection can help in dealing with the events that can affect humans and it also helps authorities to identify what are the current trends in a location. Hence, to detect these events and to generate a brief description of the same, early event detection and report generation system is developed successfully with the help of three machine learning models. Real-time data are collected from heterogeneous sources and clustered successfully with the help of the K-means clustering algorithm. And then with the help of the LSTM neural network, event detection is done. And finally, a brief description is generated using GPT-2 model. The models used have shown good results for both event detection and report generation. Giving a very good efficiency than the existing methods.

REFERENCES

[1] Mahmud Hasan, Mehmet Ali Orgun, "TwitterNews: Real time event detection from the Twitter data stream, doi:10.7287/peerj.preprints.2297V1

[2] Mario Corderio, “Twitter event detection : combining wavelet analysis and topic inference summarization”, 2012

[3] Twitter API, “https://developer.twitter.com/en/docs”

[4] News API Documentation, “https://newsapi.org/docs”

[5] Van Quan Nguyen, Tigen Nguyen Anb, Hyung-Ieong Yang, “Real time event detection using recurrent neural network in social sensors”

[6] Laxmi Lydia, P. Govindaswamy, S.K. Lakshmanaprabhu,” Document Clustering Based On Text Mining K-Means Algorithm Using Euclidean Distance Similarity, 2018

BIOGRAPHIES

Aravind M S
Btech CSE student, Mar Athanasius College of Engineering, Kothamangalam

Shiyas K
Btech CSE student, Mar Athanasius College of Engineering, Kothamangalam

Suresh Babu K
Btech CSE student, Mar Athanasius College of Engineering, Kothamangalam

Prof. Linda Sara Mathew
Assistant Professor
Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam