الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي والبحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Sétif 1 University – Ferhat ABBAS
Faculty of Sciences
Department of Computer science

جامعة سطيف 1 فرحات عباس
كلية العلوم
قسم الإعلام الآلي

THESIS

Submitted in accordance with the requirements for the degree **DOCTOR**

Domain : Mathematics and Computer science

Field of Study: Computer science

Specialty : Data, Text and Web Mining

submitted at the Department of Informatics by

NOUI Malika

THEME

Useful event detection on online social medias

Defended on: [Month, Year]

In front of the jury composed of

TOUMI Lyazid	MCA.	University of Ferhat Abbas Sétif 1	President
LAKHFIF Abdelaziz	MCA.	University of Ferhat Abbas Sétif 1	Reporter
LAOUADI Mohamed Amine	MCB.	University of Ferhat Abbas Sétif 1	Co-reporter
MEDIANI Chahrazed	MCA.	University of Ferhat Abbas Sétif 1	Examiner
NOUIOUA Farid	Prof.	University of Bordj Bou Arréridj	Examiner
AKHROUF Samir	Prof.	University of M'sila	Examiner

الملخص

حولت وسائط التواصل الاجتماعي الإنترنت إلى منصة ديناميكية، مما مكّن المستخدمين من إنشاء محتوى مختلف ومشاركته والتعليق على المحتوى النصي المتعلق بفئات مختلفة من الأحداث. ومع ذلك، فإن معالجة البيانات المستمدة من وسائط التواصل الاجتماعي تمثل تحديات كبيرة لاكتشاف الأحداث وتصنيفها بسبب الحجم الهائل للمعلومات والبنية غير الرسمية للنصوص في كثير من الأحيان. لمعالجة هذه المشكلة، سلطت الأبحاث الضوء على أهمية قاعدة البيانات، والتي لا غنى عنها لتحليل ومعالجة المحتوى النصي الوارد من وسائط التواصل الاجتماعي، لا سيما لكشف الأحداث وتصنيفها. في هذه الأطروحة، نقترح حلولًا لاكتشاف الأحداث وتصنيفها، مع التركيز بشكل خاص على التغريدات المنشورة على منصة التواصل الاجتماعي Twitter. يعد اكتشاف الأحداث أمرًا بالغ الأهمية في العديد من المجالات، مثل السياسة والرعاية الصحية وإدارة الكوارث والعلوم والرياضة والاقتصاد. تشمل مساهماتنا تطوير مجموعات البيانات ونماذج الكشف والتصنيف القائمة على الاستفادة من التطورات الأخيرة في التعلم العميق، ولا سيما بنى المحولات مثل BERT، على وجه التحديد المتغيرات التالية المصممة حسب السياقات والاحتياجات المختلفة مثل،
BERT Base، BERT Large، DistilBERT، CAMeLBERT، ARAELECTRA وغيرها. أسفرت تجاربنا مع هذه النماذج عن نتائج واعدة مقارنة بأحدث التقنيات، وحققت معدلات دقة رائعة التي وصلت إلى أكثر من 94٪ في بعض الحالات. تؤكد هذه النتائج على أهمية مناهجنا وتطبيقاتها المحتملة عبر مختلف المجالات.

الكلمات المفتاحية: وسائط التواصل الاجتماعي، NLP، الكشف عن الأحداث المفيدة،نماذج قائمة على المحولات، BERT.

Abstract

Social media has transformed the Internet into a dynamic platform, enabling users to create, share, and comment on textual content related to various categories of events. However, processing data derived from social media presents a significant challenges for event detection and classification due to the massive volume of information and the often informal structure of texts. To address this issue, research has highlighted the importance of corpus and datasets, which are indispensable for analyzing and processing textual content from social media, particularly for event detection(ED) and classification. In this thesis, we propose solutions for event detection and classification, with a specific focus on tweets published on the social media platform Twitter. Event detection and classification are especially crucial in numerous fields, such as politics, healthcare, disaster management, science, sports, economics and others. Our contributions include the construction of datasets and the development of advanced models for detection and classification, leveraging recent advancements in deep learning, notably transformer architectures such as BERT model. Specifically, a variety of model variants have been experimented to address different contexts and requirements, these include BERT Base, BERT Large, DistilBERT, CAMeLBERT, ARAELECTRA, among others. Our experiments with these models have shown promising results compared to the state of the art, achieving impressive accuracy rates that have reached more than 94% in some cases. These out comes underscore the relevance of our approaches and their potential applications across various domains, specifically in event detection and classification of text in OSM.

Keywords: social media, NLP, useful event detection, Transformers based models, BERT.

Resumé

Les médias sociaux ont transformé Internet en une plateforme dynamique, permettant aux utilisateurs de créer, partager et commenter le contenu textuel lié aux diverses catégories d'événements. Cependant, le traitement des données provenant des médias sociaux présente des défis importants pour la détection et la classification des événements en raison du volume massif d'informations et de la structure souvent informelle des textes. Pour relever cette problématique, des recherches ont souligné l'importance des corpus et des ensembles de données (datasets), indispensables à l'analyse et au traitement des contenus textuels des médias sociaux, notamment pour la détection et la classification des événements. Dans le cadre de cette thèse, nous proposons des solutions pour la détection et la classification des événements, en mettant l'accent sur les tweets publiés sur le réseau social Twitter. Cette détection est particulièrement importante pour de nombreux secteurs d'activité, tels que la politique, la santé, la gestion des catastrophes, les sciences, le sport, l'économie et autres secteurs. Nos contributions comprennent la construction d'ensembles de données et le développement de modèles avancés pour la détection et la classification, en tirant parti des progrès récents dans l'apprentissage profond, notamment les architectures de transformateur telles que le modèle BERT. Plus précisément, une variété de variantes de modèles ont été expérimentées pour répondre à différents contextes et exigences, notamment BERT Base, BERT Large, DistilBERT, CAMeLBERT, ARAELECTRA, etc. Nos expérimentations avec ces modèles ont produit des résultats prometteurs par rapport à l'état de l'art qui ont dépassé 94% dans certains cas. Ces performances témoignent de la pertinence de nos approches et de leur potentiel dans divers contextes applicatifs, spécifiquement dans la détection d'événements et la classification de texte dans OSM.

Mots-clés: médias sociaux; NLP, détection des événements utiles, modèles basés sur les transformeurs, BERT. .

Acknowledgments

Thanking is easier because we are aware that we have not arrived there all alone. Behind the success of any student is a crowd of people who contributed to his success.

I would like to thank, first of all, Allah, the Almighty, for giving me health, courage, and good will to do this modest work.

This research would not have been possible without assistance and advices of supervisors. I would like to thank my supervisors Abdelaziz Lakhfif and Mohamed Amin Laouadi for their guidance and encouragement throughout this PhD. Particularly thanks to ***Abdelaziz Lakhfif*** for to lead me in the proper direction for being there anytime I needed, thank you so much for your patience.

I also want to express my gratitude to the members of the jury for having accepted to be responsible for evaluating this project.

A very special gratitude goes out to all my friends for lending an ear and taking my mind of things when needed.

I would like to express my gratitude to my brother ***zohir*** for the many years of support for my PhD study and In recognition of the thinks towards all my family.

Dedications

To

The memory of my dear father.

My mother, may she find here the modest expression of my love, affection and my respect.

My sisters Nacira, Hassina.

My brother Zohir and his wife Wafa

Ma sister Nabila and her husband Marwan

My nephews Anwar, Safwat Yasser, SAFI Eddine

My nieces Serine, Hanaa, Safaa.

My aunt Farida and her children.

My friends and all those I love.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Contents

In a relatively short period, social media has become an integral part of the daily activities of internet users and professionals. Based on Global WebIndex statistics from 2021, 57.6% of the world's population actively used social media, with daily usage averaging between 6 and 8 hours—particularly following the impact of COVID-19 [1]. These platforms facilitate information sharing, commenting, and personal expression, establishing a new landscape for public relations. Social media is grounded in collaboration and the exchange of information, allowing widespread participation due to minimal barriers to entry. This structure encourages users to contribute actively and share their perspectives, effectively closing the gap between the public and traditional media. Unlike conventional media, which primarily transmits information in a one-sided manner, social media fosters dialogue and highlights the importance of attentive listening and engagement.

Web services known as social network sites enable users to make public or semi-public profiles, share connections, and view their list. Users can build profiles on these networked communication services, establish connections, and exchange content. Social media generates vast volumes of information, including private data and significant data on notable events like political developments, natural disasters, and economic trends, providing valuable opportunities for predictions [2, 3, 4].

In natural language processing (NLP), one of the most crucial tasks is event detection and classification, focusing on the identification of event instances within textual data. Its significance has received a lot of attention lately because of its effects on many different areas., such as finance, elections, social events, sports, and so on. The complexity of event detection and classification involves multiple sub-tasks of varying difficulty, prompting researchers to develop a range of techniques and methodologies. A significant focus has been placed on the concept of similarity, which is essential for effective event detection and classification, leading to the establishment of various formulas to improve clustering efficiency. Additionally, the evaluation of proposed models necessitates the creation of diverse linguistic corpora and the establishment of relevant evaluation measures. Since the last decade , event detection and text classification have been studied using methods such as topics, incremental TF-IDF models [5], and BERT, a large-scale transformer-based language representation model Recent benchmark datasets have been introduced for fine-grained event classification, allowing comparison of various models' performance.

To enhance the detection and classification of useful events in online social media, this thesis explores deep learning and transformer based models.This enhancement is achieved through the proposal of several frameworks. The first framework introduces a model that combines long short term memory network(LSTM) with Bidirectional Encoder

Representations from Transformers(BERT) representations to effectively detect and classify events in tweets[6]. To support this research, a dataset of approximately event-related tweets was compiled and categorized into distinct event types. Additionally, a framework for the multi-classification of disaster tweets has been proposed to improve the analysis of social media data during emergencies . A deep learning-based similarity detection model, utilizing BERT, has been proposed for effective event detection and classification. This study introduces a model for event detection and classification in short texts like tweets, chat messages, and online comments. While clustering algorithms have been widely used for classifying tweets, we compare their performance on Arabic and English tweets using TF-IDF (Term Frequency-Inverse Document Frequency) and pre-trained BERT models. The analyzed algorithms include K-means, K-medoids, and Agglomerative clustering.

1.1 Motivation

Online discussions generate vast amounts of contextual data and information, which may include personal details related to updates and privacy. However, a significant portion of the conversations revolves around events, such as political events (e.g., elections), natural occurrences (e.g., floods, cyclones), and economic events (e.g., sales). The detection of events and the classification of texts on social media offer the possibilities to make previsions in several areas for example concerning marketing, how to increase the production of one product, decrease for another?. For natural phenomena like earthquakes, cyclones, and innondations, what precautions will be taken by governments to minimize their consequences?. For epidemics, what health advice should be provided to patients?. So event detection and classification will, thus, provide beneficial information for making good decisions and to adapt adequate strategies in all areas of life.

In the last several years, researchers on NLP (Natural Language Processing) have increasingly focused on the detection and classification of events from social media platforms, particularly Twitter. This trend arises from the platform's vast and varied content made up of users, which presents an abundant source of accurate data. By employing advanced algorithms and analytical techniques, these researchers aim to detect significant events, classify them into relevant categories, and analyze the importance and context surrounding these occurrences. In addition to improving our comprehension of public conversation, this approach offers insights into issues that may warrant further investigation of useful events related to important domains, including marketing, public health, disaster, and so on.

Natural language processing (NLP) is now much more effective and efficient because of recent developments in artificial intelligence, especially in the fields of neural networks and deep learning. By using sophisticated algorithms that replicate the connections between neurons in the human brain, these technologies enable machines to comprehend, interpret, and produce human language more accurately.

1.2 Research Challenges

Certainly, Online social media data, often referred to as social media analytics, encompasses the vast amount of information generated by users on Facebook, Twitter, Instagram, LinkedIn and so on platforms. Absolutely, while social media data offers a treasure trove of insights, there are also numerous challenges associated with using it effectively:

- Finding relevant information that can be classified as valuable content presents an

interesting challenge. While many tweets may not relate to significant events, this situation provides an opportunity to develop better strategies for identifying and extracting meaningful insights by focusing on filtering techniques and honing our understanding of what constitutes useful information.

- Event detection and classification algorithms should be designed to account for the Twitter stream's nature, which constantly evolves as new information emerges. These algorithms need to effectively manage the influx of redundant messages that may refer to the same issue or event, as this can lead to misinformation or confusion about the significance of the updates. Moreover, it is essential to recognize that the open platform allows anyone to report an event; therefore, not all reported incidents are reliable or newsworthy. Careful assessment of the credibility and relevance of these events is crucial in order to remove intrusions and concentrate on useful events.

- The distinct syntactic structures and semantic nuances of each language, along with various dialects, make it challenging to detect, classify, and analyze user messages for useful even detection. These linguistic features can lead to misunderstandings and complicate the accurate interpretation of intent and meaning, particularly when idiomatic expressions and culturally specific references are involved.

1.3 Contributions

Online social media platforms serve as crucial channels for real-time information exchange, with millions of users sharing their experiences and insights. This study presents effective methodologies for detecting and classifying events within these environments, specifically Twitter platform, by combining deep-learning, supervised and unsupervised learning techniques. Consequently, the following are this thesis's main contributions:

- **Building Datasets for Event Detection and Classification:** One of the contributions of this thesis is to build new datasets for effective event detection, classification, and clustering. A dataset is collected from the Twitter platform , while additional datasets are generated from various corpus and sources using different algorithms or selection methods.

- ***Models and algorithms for event detection and classification:*** To detect and classify useful event-related tweets, we proposed four frameworks based on supervised learning:

    1. We introduced a model that integrates BERT representations with an LSTM network to effectively identify and categorize significant events mentioned in

tweets. For this purpose, we collected a dataset of event-related tweets, which were categorized into 50 different types of events [6].

2. A framework that involved a multi-classification approach to categorize disaster-related tweets by employing recurrent neural networks (RNN) and long short-term memory (LSTM) networks [7].
3. This study presents a similarity model designed for the detection of relevant events. We systematically compare the performance of models utilizing various pre-trained word embeddings as features, specifically BERT, with those models that employ a Term Frequency-Inverse Document Frequency (TF-IDF) presentation.
4. A semantically model was developed to address the challenges of event detection and classification in short text on three datasets.

- ***Also, we adopted a comprehensive approach based on unsupervised learning***, focusing on the comparison between the TF-IDF method and a pre-trained BERT model, specifically DistilBERT, for analyzing English tweets. For Arabic tweets, we utilize two advanced models: Araelectra-base-discriminator and bert-base-arabic-camelbert-da-sentiment. In addition, we implement several clustering algorithms to group similar tweets based on their semantic content. Finding the tweets that best represent a certain event and grouping them appropriately is one of the biggest problems the NLP community is currently facing. To address this, we propose an algorithm based on the mathematical concept of permutations without repetition to label the clusters.

1.4 Publications

- **Journal Papers**
  Malika Noui, Ablaziz Lakhfif, and Mohamed Amin Laouadi, ***"Event detection and classification in tweets using deep learning"*** Engineering, Technology & Applied Science Research, vol. 15, no. 1, pp. 19977–19982, 2025.[6].https://doi.org/10.48084/etasr.9238

- **Conference Papers**
  1. Malika Noui and Ablaziz Lakhfif. ***"Useful Event Detection on Short Text Using Deep Learning"***. In 2022, The Fifth International Symposium on Informatics and its Applications (ISIA 2022) Mohamed Boudiaf University of M'Sila, Algeria, November 29th-30th.
  2. Malika Noui, Ablaziz Lakhfif, and Mohamed Amin Laouadi, ***"Towards useful event detection and sentiment analysis of osn for disaster man-***

agement". In 2024 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM) (2024), IEEE, pp.1–7. https://doi.org/10.1109/ICT-DM62768.2024.10798933[7].

3. Malika Noui, Ablaziz Lakhfif, and Mohamed Amin Laouadi, ***"A Comparative study of Clustering methods for Event Detection and Classification in Arabic and English Tweets"*** . In 2025 Mediterranean Conference on Computer Science and Artificial Intelligence MCCSAI'2025.

1.5 Thesis Outline

The dissertation is organized into six chapters:

- **Chapter 1** introduces the project, outlines the contributions and motivations and highlights the challenges that arise. Lists the papers that have been published, a brief summary of each chapter, and the thesis outline.
- **Chapter 2** describes the background of research, provides an overview of key concepts related to event analysis: defining an event, event types, and certain events that are particularly useful, explores the event process, and explains event Detection and classification on online social media and on Twitter.
- **Chapter 3:** This chapter outlines the process of collecting data from Twitter and documents. It also provides a detailed discussion of various datasets that are utilized in this study. The chapter defines both specified and unspecified event detection and classification. Section 3 is dedicated to the state of the art in event detection and classification.
- **Chapter 4:** This chapter goes over the basic ideas of methods for event detection and classification. Gives a full rundown of different kinds of neural networks, such as Artificial Neural Networks (ANNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTMs), and Gated Recurrent Units (GRUs). Each type of network will be examined in terms of its architecture, functionality, and equations. Additionally, the chapter explores various techniques for data representation, focusing particularly on word embeddings and transformers , which are crucial for natural language processing tasks such as Word2Vec, GloVe, FastText, and BERT's models **ARAELECTRA, CAMeLBERT, and DistilBERT**. The discussion will also cover methods for measuring text similarities like cosine similarity, Euclidean distance, etc. Furthermore, the chapter elaborates on the metrics used for evaluating the performance of models such as accuracy, precision, recall, and F1-score. It will

also provide insight into activation functions and their role in determining the output of neural networks (Sigmoid, Tanh, Softmax, etc.).Lastly, the chapter presents three popular clustering techniques: **K-means, K-medoids, and agglomerative clustering**.

- **Chapter 5:** This chapter offers an in-depth exploration of the practical methodologies involved in collecting datasets and developing models for event detection and classification. It begins by detailing the techniques for collecting tweets, delivering datasets for event detection, and illustrating dataset statistics. It outlines effective frameworks for event detection and classification. The primary framework combines Long Short-Term Memory Networks (LSTM) with Bidirectional Encoder Representations from Transformers (BERT) to efficiently detect and classify events in tweets. Additionally, we propose a multi-classification framework specifically designed for disaster tweets to enhance social media data analysis during emergencies. Furthermore, we introduce a deep learning-based model using BERT for similarity detection, aimed at identifying events in short texts such as tweets and chat messages. We also compare the performance of various clustering algorithms, including K-means, K-medoids, and Agglomerative clustering, on Arabic and English datasets . This comparison utilizes TF-IDF and pre-trained BERT models, measuring performance using metrics such as the Silhouette score, accuracy, precision, recall, and F1-score.

- **Chapter 6** gives an in-depth review of the results of our research and offers suggestions for possible future directions of exploration.

Chapter 2

Background

Contents

2.1 Introduction

Chapter 2 delves into the concept of events on OSM, focusing on their definition, types, and detection methods. It extends the discussion to the realm of Online Social Media (OSM), emphasizing the importance of social media platforms as sources of data for event detection and categorization. Twitter, as a case study, is examined in depth, highlighting its functionalities, usage statistics, and challenges as a source of information. This chapter aims to bridge the theoretical understanding of events with practical applications in social media research.

2.2 Defining an event

2.2.1 What is an event?

The definition of an event varies depending on the discipline. Despite the lack of a formal, accepted definition, we can examine a number of definitions that offer insightful perspectives on this idea.

Definition 1: An event is a notable and memorable occurrence that occurs at a certain time and place..[8, 9, 10].

Definition 2: Something is considered significant when it captures the attention of the media. For instance, you might come across a news article or view a news report that highlights its importance. This communication helps to raise awareness and fosters discussion about various topics in society. [8, 10].

Definition 3: An important event that occurs in the real world is called a trending event, characterized by three key factors: [11] :

1. T_e present time associated to event;
2. A number of documents published during this time period T_e that successfully handle the event D_e;
3. Discover the key features that define the occurrences in which T_e emerges as a trending time period within the document stream D_e.

A "document stream" is a group of social media documents, such as YouTube videos and Twitter messages, that are distinguished by contextual elements like titles and tags that help convey their importance and content.

Definition 4: An event, denoted as **E**, is constituted by a combination of three fundamental components [12].The first component is a semantic property, represented as **S**, which captures the inherent meaning or context of the event. The second component is a time interval, denoted as **I**, which specifies the duration or time frame during which the event occurs. Finally, the spatial entity, represented as **SP**, identifies the location or physical setting associated with the event. Together, these components comprehensively represent the event as ***E(I, SP, S)***, illustrating its multifaceted nature.

Definition 5: An event is a structured sequence of clusters of trending entities organized chronologically, known as a "cluster chain" This framework enhances the understanding of the relationships among these entities as they gain prominence over time. Each cluster reflects current interest, providing important insights into the processes of information flow and trends in both business and academic contexts.[8].

2.2.2 Types of events

Events can be categorized into several distinct types [9]:,

- **Planned:** These events are systematically organized with a predetermined timeline and designated venue. This category encompasses a wide range of activities, including conferences, trade shows, and community festivals, such as book fairs.
- **Unplanned:** In stark contrast, unplanned events arise unexpectedly and without prior notification. These occurrences can significantly disrupt daily life and may include natural disasters, such as earthquakes or floods, as well as social upheavals like strikes. The unpredictable nature of these events often necessitates rapid response measures from authorities and communities to mitigate their impact and ensure public safety.
- **Breaking News:** This category comprises events that garner immediate and extensive coverage from mainstream news media. Such events typically include significant political developments, like election results, natural disasters, or major international incidents. The international press is essential to the public's dissemination of information, facilitating informed discourse on these pivotal occurrences and their wider implications.
- **Local :** Local events are confined to a specific geographical area and tend to affect only the residents or stakeholders within that region. Examples include community gatherings, sports events, or minor incidents such as car accidents. While their impact may be limited geographically, these events often foster a sense of community engagement and local identity.

- **Gobal :** Unlike local events, global events have far-reaching implications and are not restricted by geographic boundaries. This category includes major happenings such as the World Cup, international summits, and global awareness campaigns. These events attract participants and audiences from around the world, fostering intercultural exchange and collaboration on a large scale. Their significance often extends beyond mere entertainment, influencing economic, social, and political landscapes globally.

2.2.3 Useful event

A useful event is one that pertains to significant axes of life, addressing essential areas such as health, sports, disaster management, business, and so on. Usually, these events provide spaces for community involvement and communication with the goal of raising living standards and promoting a healthy society.

2.2.3.1 Useful event in Politics

Political conversation has been more prevalent on social media sites like Facebook and Twitter in recent years.. This trend has prompted researchers to enhance their methodologies for detecting political events effectively.[13, 14, 15, 16, 17, 18]

2.2.3.2 Useful event in Health

In terms of health, useful events can include health fairs, workshops on nutrition and wellness, and seminars focusing on mental health awareness. Such gatherings not only provide valuable information but also encourage individuals to take proactive steps towards maintaining their health, fostering a culture of prevention and early intervention. Health has received significant interest in several research studies [19, 20, 21, 22, 23, 24].

2.2.3.3 Useful event in Disaster

Disaster management events are crucial for preparing communities for unforeseen emergencies. Workshops and training sessions focusing on emergency preparedness, response strategies, and recovery plans ensure that individuals and families are equipped to handle crises. These events help raise awareness about the importance

of preparedness, community coordination, and resource management in times of disaster [25, 26, 27, 28, 29].

2.2.3.4 Useful event in Social sciences

Security in society presents an important topic often discussed within social networks, to discover how citizens react to security problems such as crime, violence, cyber threats, hate speech, disruptive behavor, accidents and so on. Several studies have been carried out[30, 31, 32, 33, 34, 35, 36, 37].

2.2.3.5 Useful event in Sport

An emerging area of scholarly interest is the study of event detection in sports on social media platforms. Researchers analyze trends and discussions to uncover how fans engage, share experiences, and build online communities. This analysis identifies crucial moments, such as game highlights and trending fitness challenges, enabling event organizers and sponsors to enhance outreach strategies.[38]

2.2.4 Definition of Event detection

The concept of event detection can vary significantly depending on the context in which it is applied[8]. For instance, within the domain of news wire documents, event detection is often described as the process of identifying specific trigger words that signal the occurrence of particular events. This process also involves categorizing these identified events into distinct and refined types to enhance clarity and understanding[39]. In contrast, when we consider event detection in the context of microblogs and OSM such as Twitter, the underlying principles are conceptually similar to those found in classification, clustering tasks and so on(see Figure 4.1).

Event detection in online social media refers to identifying real-world events based on user interactions such as posts, likes, shares, and comments. This process involves [40]

$$E = \{e_1, e_2, \ldots, e_M\} \tag{2.1}$$

where M is the number of detected events, and each event e_i consists of several components:

- $R(e_i)$: The textual description of the event, summarizing what happened.

- $A_{e_i} \subset A_n$: A subset of actions from the OSN that are related to the event.
- $T_{e_i}^A$: The time period in which these actions occurred, ranging from the event's start time $t(A_{e_i,\text{start}})$ to its end time $t(A_{e_i,\text{end}})$.
- loc_{e_i} : The location associated with the event, if applicable.
- I_{e_i} : The set of involved users who participated in discussions or interactions related to the event.

Current approaches to event detection often focus on specific aspects rather than capturing the full scope of an event.

2.2.5 Event detection methods

Two primary categories of event detection were identified [41, 42, 43, 44] : New Event Detection (NED) and Retrospective Event Detection (RED).

- NED methods analyze real-time Twitter streams to identify newly emerging events.These approaches leverage the dynamic nature of social media to detect unexpected occurrences as they unfold.
- RED methods, in contrast, rely on historical datasets to retrospectively identify significant events. By examining past trends and anomalies, RED enables the detection of events after they have taken place.

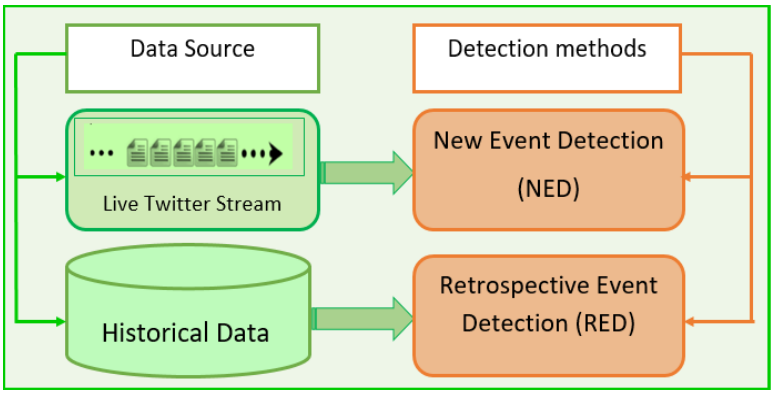


Figure 2.1: Detection methods (NED,RED)

2.3 Event Detection on OSM

2.3.1 Definition of social media

According to [45, 3, 46], social network sites, also known as social media, are online resources that enable visitors to:

- Create a public or semi-public profile inside a system that has boundaries;
- enumerate the other users with whom they are associated;
- View and navigate both their own and other users' connections within the system.

According to one definition, social network sites, often known as social media, are networked communication platforms where users can:[47]

- make content and profiles;
- make connections and create visual and audio interactions with those links;
- share user-generated content.

2.3.2 Social media data

Social media is a valuable source of information. The online conversations generate a significant amount of data. While some of this data may be personal, such as updates about statuses and privacy information, a large portion consists of discussions about various events. These events include political happenings (e.g., elections), natural disasters (flood, cyclone and so on), and economic activities (sales), among others. [48]

2.4 Event Detection on Twitter

2.4.1 Presentation

Twitter was established in 2006 and is classified as a microblogging platform where users communicate with their followers in "real time" by sending 280 character tweets [49, 50, 51]. Using hashtags, responses, and mentions, users can have conversations.
Tweets, one of Twitter's most widely used microblogging platforms, can have a maximum character count of 280. Over 500 million tweets are sent daily. 316 million monthly active users send them every day. One Consequently, there have been a number of proposals to use Twitter as a source of current news and information, such as tracking the coronavirus (COVID-19) pandemic [52], responding to natural disasters [53], tracking epidemics [54],

or keeping tabs on Brazilian political elections [55].

Every day, almost 500 million tweets are written by the more than 100 million individuals. Twitter is one of the most widely used social networking and online news sites.

2.4.2 Functionalities

- **Send a tweet:** Tweets are like Facebook posts. However, there are only 280 characters available. So you can send short messages, for example, on which project you are currently working, on what you are currently interested in, etc. A tweet can also contain photos and videos. If your tweets please, users will, subscribe to your Twitter profile and will therefore, be your followers. You can also, as an author, follow other profiles by clicking on the "follow" button. A tweet can only be issued by a Twitter account and has no duration of lapse and only the author of a tweet can decide on the delete.

- **The hashtag (#):** Using the hashtag, you highlight terms in keywords. It is characterized by the hash symbol (#) and will be flagged by Twitter as a link. You make it clear to your followers what your tweet is about and thus show the importance of the term used.
  Common hashtags Used on a daily basis, there are in each language, each country, etc... It should be known at minimum 3:

  - **#TT: Trending Topics**, used to talk about a topic currently popular on Twitter.
  - **#FF : Follow Friday**, used the Friday to suggest to his subscribers to follow certain accounts.
  - **#LT : Last Tweet**, used to make reference to the last tweet issued.

- **The mentions (@):**It allows you to mention a Twitter user in order to respond directly to them. They act as a link to the user's profile. Remember that all followers can see this message because it is not private.

- **The ReTweet (RT):** The "ReTweet" function can be defined as the action of republishing a message on Twitter, like the “share” feature on Facebook. ReTweets are a good tool because they show that the information you just read on Twitter is important to you and may be of interest to followers of your tweets.

2.4.3 Twitter usage statistics

On average, there are about 6,000 tweets posted every second on Twitter, which equates to more than 350,000 tweets sent every minute, 500 million tweets each day, and approximately 200 billion tweets annually. The number of tweets per day across Twitter's history is displayed in the figure below:

Figure 2.2: Growth of posted tweets worldwide per day
http://www.internetlivestats.com/twitter-statistics/

2.4.4 Challenges in using Twitter as a source of information

Using twitter as a source of information is a great challenge considering the particularities of this information, which can be noted[9].

- **Large stream and Velocity:**The very high number of tweets posted per minute on Twitter, we are talking about more than 500 million tweets per day.
- **Brevity:** Because users typically post brief messages, standard text mining and natural language processing techniques are inappropriate. Another barrier is the tweets' brief (280 characters) length, which suggests a problem with semantic understanding.
- **Noise:** Tweets are noises through the use of colloquial language, abbreviation etc.
- **Dynamic nature:** Event detection cannot be static; we must manage conversational continuity and discontinuity as well as the evolution of events across time.

- **Weighting used in vector presentation:** The choice of the weighting of the vector representation is very delicate. Textual representations such as Term frequency,Inverse document frequency (IDF), TF-IDF weighting, are not sufficient.
- **Similarity Measures:**determining the similarity between tweets is a primary task in the detection of events; several syntactic and semantic similarities approaches are available, but it is necessary to select which one corresponds to the objectives targeted, as hybrid approaches can be used

2.5 Conclusion

Chapter 2 provides a comprehensive exploration of event detection, particularly within the context of Online Social Networks and Twitter. The insights gained underscore the evolving role of social media platforms in event detection and information dissemination. While these platforms offer valuable opportunities for research and analysis, they also present unique challenges that must be addressed. This chapter lays a strong foundation for leveraging social media as a tool for understanding and responding to events in an increasingly digital world.

Chapter 3

Techniques and Methods

Contents

3.1 Introduction

In this chapter, we explore a diverse array of methodologies employed to analyze and process data effectively for event detection and classification tasks . Beginning with similarities-based methods, we delve into how texts are represented and measured for similarity. We then present machine learning approaches, distinguishing between supervised and unsupervised learning paradigms. Following this, deep learning methodologies, including LSTM, GRU, and the groundbreaking Transformer models, are examined in detail, along with their integration of word embeddings. Finally, we address activation functions and evaluation metrics, emphasizing their pivotal roles in optimizing and assessing the performance of models.

3.2 Similarities based methods

3.2.1 Texts similarities

Various methods have been proposed to measure the similarity between texts, ranging from basic string comparisons to advanced machine learning models (see Figure 4.1). Assessing the similarity between texts is a critical challenge across numerous fields. It plays a pivotal role in disciplines like:

- **Textual Analysis:** Facilitates understanding and comparing literary styles, sentiments, or thematic elements within texts.
- **Data Mining:** Supports the extraction of patterns, trends, and insights from large textual datasets, including social media content or customer feedback.
- **Natural Language Processing (NLP):** Assists in machine learning tasks like translation, summarization, and question answering by determining semantic or contextual similarity.
- **Event Detection and Classification:** This plays a vital role in identifying and categorizing events within textual data, which is especially useful in fields like disaster management, public safety, and real-time information analysis.

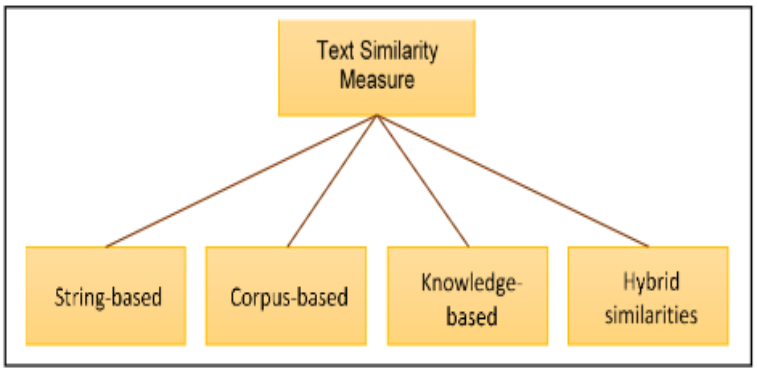


Figure 3.1: Text Similarity Approaches
[56]

Term frequency

Note how many times a term appears in a document. Indication of the term's significance inside the text.The amortization of variations and/or consideration of document length are made possible by standardizing frequencies:

$$\boxed{tf(t,d) = \frac{f_{(t,d)}}{\sum\limits_{t'} f_{(t',d)}}} \tag{3.1}$$

Inverse document frequency (IDF)

When used in a document, a term that is found in practically every corpus **(D)** has minimal impact. On the other hand, an uncommon term that appears in a document has to The **IDF** calculates a term's significance within a corpus.

$$\boxed{idf(t,D) = log_{10}\frac{N}{n_t}} \tag{3.2}$$

N: How many documents are in the corpus?
n_t: How many documents in which the word is used?
Note: Another formula is used sometimes to avoid getting a null **IDF**

$$\boxed{idf(t,D) = log_{10}(1 + \frac{N}{n_t})} \tag{3.3}$$

TF-IDF weighting

Relativizing the importance of a term in a document **(TF)** by its importance in the

corpus **(IDF)**.

$$\boxed{tfidf(t, D) = f_{(t,d)} log_{10} \frac{N}{n_t}} \tag{3.4}$$

3.2.2 Measures of Similarity

3.2.2.1 Cosine Similarity

It lets us determine how two vectors are related by computing the cosine of the angle [57]. The definition of the cosine similarity between A and B is:

$$\boxed{cos_{sim}(d_1, d_2) = \frac{d_1.d_2}{|d_1|.|d_2|}} \tag{3.5}$$

Here:
$d_1.d_2 = [tf(t_1, d_1) * tf(t_1, d_2)] + [tf(t_{21}, d_1) * tf(t_2, d_2)], ... + [tf(t_n, d_1) * tf(t_n, d_2)]$

$|d_1| = \sqrt{tf(t_1, d_1)^2 + tf(t_2, d_1)^2 + ... + tf(t_n, d_1)^2}$

$|d_2| = \sqrt{tf(t_1, d_2)^2 + tf(t_2, d_2)^2 + ... + tf(t_n, d_2)^2}$

3.2.2.2 Jaccard Coefficient

Based on whether a term is present in a document or not, the Jaccard coefficient is used to compare how similar or unsimilar two documents are. The ideal way to calculate it is to divide the total number of terms that are present in at least one of the two texts by the total number of terms that are common to both documents.[58, 59, 60] It is defined as,

$$\boxed{Jaccard(A, B) = \frac{|A \bigcup B|}{|A \bigcap B|}} \tag{3.6}$$

Or

$$\boxed{Jaccard(D_1, D_2) = \frac{D_1 \times D_2}{|D_1| + |D_2| - D_1 \times D_2}} \tag{3.7}$$

Here: $|D_N| = (x_{N1}^2 + x_{N2}^2 + x_{N3}^2 + ... + x_{N_m}^2)$
$D_1 \times D_2 = (x_{11} \times x_{21} + x_{12} \times x_{22} + x_{13} \times x_{23} + ... + x_{1m} \times x_{2m})$

3.2.2.3 Dice Coefficient

By dividing the total number of terms in both papers by the number of common terms in the compared texts, the dice coefficient is computed. The formula is defined

as[58, 59, 60]

$$\boxed{Dice\ coefficient = 2\frac{|A \bigcap B|}{|A| + |B|}} \tag{3.8}$$

[0, 1] is the range of the dice coefficient value map, where 0 denotes not intersecting and 1 denotes an exact match. Also, based on the presence or absence of terms in documents, the Dice coefficient is comparable to the Jaccard coefficient.

3.2.2.4 Overlap Coefficient

The size of the union of sets A and B over the size of the smaller set between A and B is the Overlap Coefficient, sometimes referred to as the Szymkiewicz–Simpson coefficient.The definition of the formula is[58, 60]:

$$\boxed{overlap(D1, D2) = \frac{D_1 \times D_2}{min(|D_1|, |D_2|)}} \tag{3.9}$$

3.2.2.5 Euclidean distance(L2)

As described by [61], Euclidean distance computes the similarity between two documents d_1 and d_2 by reducing the distance between their vector representations ($\vec{x}$ and $\vec{y}$) to a single point.The distance is defined as follows:

$$\boxed{Euclidean\ distance(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}} \tag{3.10}$$

3.2.2.6 Manhattan distance(L1)

The Manhattan distance is a measure of the separation between two locations in a vector space with N dimensions. To put it simply, it is the total of the absolute differences between two points' measurements in every dimension. It is described as[61]:

$$\boxed{Manhattan\ distance(\vec{x}, \vec{y}) = \sum_{i=1}^{n}|x_i - y_i|} \tag{3.11}$$

3.3 Machine Learning based methods

Machine learning has emerged as a powerful tool for solving complex problems across various domains, driven by its ability to learn patterns from data and make informed predictions or decisions. Among its many approaches, supervised and unsupervised learning stand out as foundational techniques, each with distinct characteristics and applications. This section delves into these core methods, exploring their principles, differences, and practical implications to provide a comprehensive understanding of machine learning strategies.

3.3.1 Supervised learning

Supervised learning is a fundamental approach in machine learning where a model is trained using labelled data. This means the training dataset includes input-output pairs, where the input is the data used for prediction, and the output (or label) is the desired result.

3.3.1.1 Neural Networks (ANN)

Neurons are the cells that compose the brain. The inspiration for Artificial Neural Networks (ANN) came from the human brain, which has roughly 1011 neurons connected to one another via 10,000 connections. [62].

a/ Model of an artificial neuron: An ANN is composed of artificial neurons that are connected to each other, ANN's model developed from biological neural network [63].

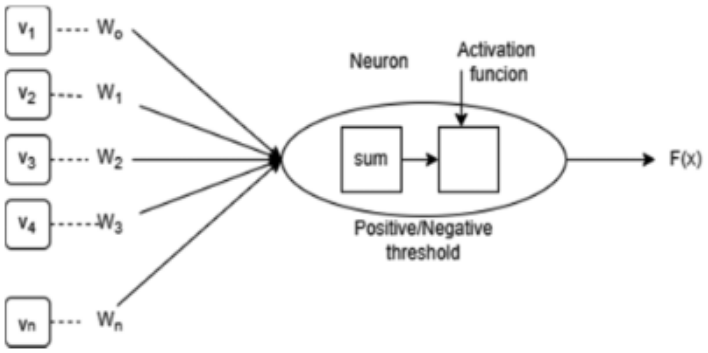


Figure 3.2: Model of an artificial neuron

b/ Neural network architecture: The ANN's architecture consists of the input layer and includes neurons in proportion to the amount of hidden layer inputs. The complexity output layer determines how many hidden layers, there are and how many neurons are there in each layer: Although it can have more than one output, it typically has a

single neuron with an output that is between 0 and 1, or larger than 0 and less than 1. [62].

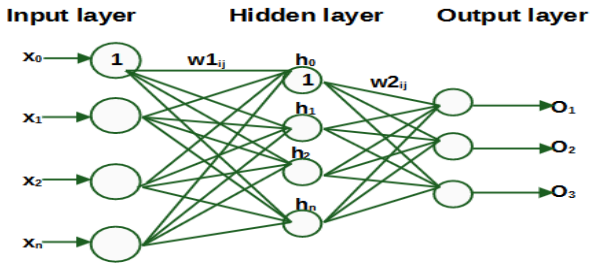


Figure 3.3: Neural network architecture

3.3.1.2 Recurrent Neural Network (RNN)

One benefit of RNNs is their ability to handle sequential data, including texts that show correlations between closely spaced sequences. Recursion of a transition function to its hidden states for every symbol in the input sequence is the foundation of RNNs. The current input x_t and the previously concealed states h_{t-1} are used to compute the value of the hide layer activation at time t as a function f:

$$h_t = f(x_t, h_{t-1})$$

$$h_t = \emptyset(Wx_t + Uh_{t-1}) \tag{3.12}$$

where W is the input-to-hidden weight matrix, U is the state-to-state recurrent weight matrix, and $\emptyset$ is usually a logistic sigmoid function or a hyperbolic tangent function.

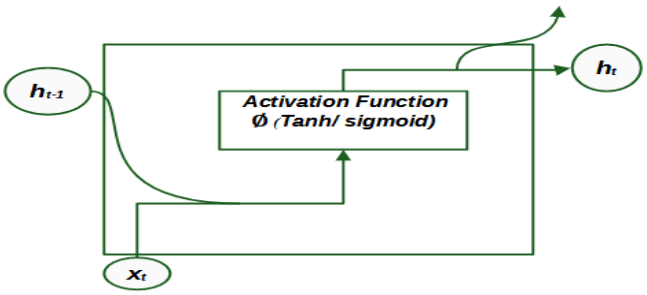


Figure 3.4: Recurrent Neural Network (RNN)

3.3.2 Unsupervised learning

Unsupervised learning is a powerful machine learning approach in which the model is trained on data without labeled outputs. This enables the model to identify patterns and structures in the data independently, making it a valuable tool for extracting insights from unstructured information.

3.3.2.1 K-means

The K-means algorithm is a technique for unsupervised machine learning. that partitions a dataset into distinct clusters. Its primary objective is to guarantee that the points in a given cluster are identical. The process starts with establishing k centroids representing the central positions of the clusters, and then assigns samples to the nearest group using Euclidean distance. The process continues until cluster assignments remain unchanged or a predetermined number of cycles are reached.[64]

3.3.2.2 Agglomerative clustering

Agglomerative clustering is a hierarchical method used for clustering textual data. It uses a gradual aggregation process, starting with individual data points and merging them in iterations until a stopping criterion is met. This method is versatile for large data with complex relationships and is easy to implement. The distance measure and linkage function used are critical decisions in agglomerative clustering. However, improper pairing can lead to poor cluster performance and inaccurate results.[64]

3.3.2.3 K-Medoids

Medoids are the central points of a cluster that are found using the K-Medoids technique. Whereas K-Means employed the sum of squared Euclidean distances for data objects, K-Medoids use k as a representative object to minimize the sum of dissimilarities of data objects. Additionally, this distance metric lowers outliers and noise. [65].

Supervised versus unsupervised learning

The two primary forms of machine learning, supervised and unsupervised, are appropriate for various tasks and kinds of data. Training a model on labelled data, where the inputs (features) and their matching outputs (labels) are supplied, is known as supervised learning. Training a model on data without labelled outputs in order to find patterns, structures, or correlations within the data is known as unsupervised learning..

Table 3.1: Supervised versus unsupervised learning

	Supervised	Unsupervised
Features	Labeled data	Unlabeled data
	Map inputs to outputs,	Identify hidden structures
Types of problems	Classification:Predicting categories events detection, image classification.	Clustering: Group similar data customer segmentation, Dimensionality Reduction
	Regression,	Density Estimation
Algorithms	Linear Regression[**66**] Logistic Regression [68] Decision Trees [69] Support Vector Machines[71] Neural Networks [72]	K-Means Clustering[67] Agglomerative[38] DBSCAN, K-Medoids [70] Principal Component Analysis (PCA)[71]

3.4 Deep Learning-Based Methods

3.4.1 Long short-term memory (LSTM)

To explicitly address this problem of learning long-term dependencies, Hochreiter and Schmidhuber (1997) [73] suggested long short-term memory (LSTM). A distinct memory cell within the **LSTM** is kept up to date and only makes its contents visible when it is judged necessary. Several variations of the **LSTM** have emerged since the original model was proposed in 1997. The model and different equations are depicted in the following figure[74].

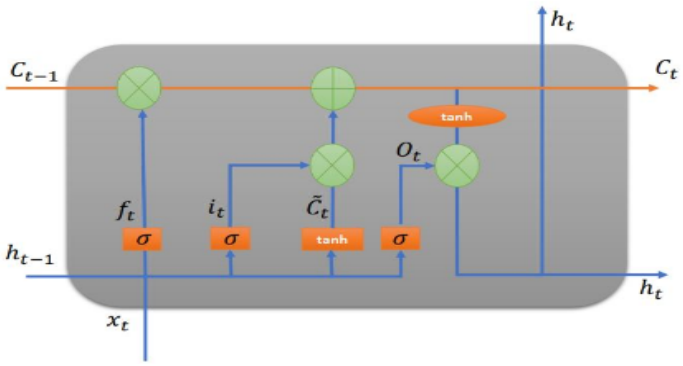


Figure 3.5: The cell structure of a long short-term memory unit.

$$f_t = \sigma(x_t W_f + h_{t-1} U_f + b_f) \tag{3.13}$$

$$i_t = \sigma(x_t W_i + h_{t-1} U_i + b_i) \tag{3.14}$$

$$o_t = \sigma(x_t W_o + h_{t-1} U_o + b_o) \tag{3.15}$$

$$\tilde{C}_t = \tan(x_t W_C + h_{t-1} U_C + b_C) \tag{3.16}$$

$$C_t = \sigma(f_t \times C_{t-1} + i_t \times \tilde{C}_t) \tag{3.17}$$

$$h_t = \tanh(C_t) \times o_t \tag{3.18}$$

3.4.2 Gated Recurrent Neural Network(GRU)

In general, GRU and LSTM cells function nearly identically; however, because GRU cells only have two gates compared to LSTM cells' three, they are less likely to overfit on small datasets. Like the LSTM, the GRU is made to remember and forget information dynamically. The typical GRU model operates as follows:[74].

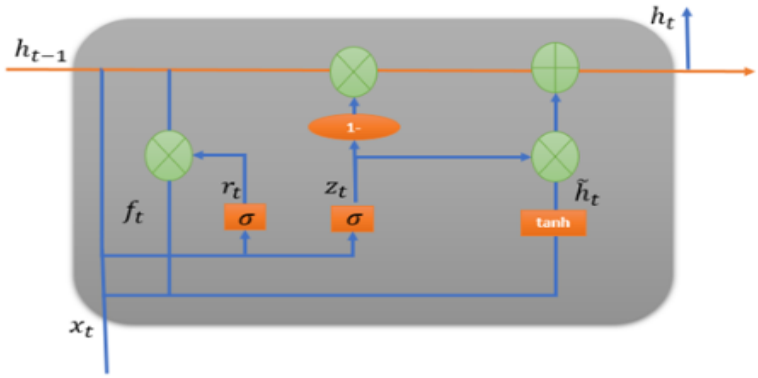


Figure 3.6: The cell structure of a gated recurrent unit.

$$z_t = \sigma(x_t W_z + h_{t-1} U_z + b_z) \tag{3.19}$$

$$r_t = \sigma(x_t W^r + h_{t-1} U^r + b_r) \tag{3.20}$$

$$\tilde{h}_t = \tan(r_t \times h_{t-1} U + x_t W + b) \tag{3.21}$$

$$h_t = (1 - z_t) \times \tilde{h}_t + z_t \times h_{t-1} \tag{3.22}$$

3.5 Word embedding

When using deep neural networks for text classification, word embedding is an essential step. A general name for a vectorized representation of words is "word embedding," in which words are mapped to vectors rather than a one-dimensional space. Various techniques have been put forth to create embeddings, citing:

- **One Hot Embedding (OHE):**One-hot encoding is an NLP technique that represents words as sparse numerical vectors. Each word is mapped to a unique vector,

where most values are zeros and a single indicates the word's position in the vocabulary. Although simple, this method doesn't capture contextual or semantic relationships between words, making it less suitable for tasks requiring deeper understanding [75].

- **Term Frequency-Inverse Document Frequency (TF-IDF):** A term's frequency (t) is calculated by dividing the number of times it appears in a document by the total number of terms in that document. IDF (inverse document frequency) is used to assess a term's importance within a document. IDF is computed as IDF (t) = log(N/DF), where DF is the number of documents that include the phrase t and N is the number of documents. A term's importance in a particular text in relation to the entire corpus is shown by the resulting TF-IDF score. TF-IDF is frequently employed in clustering methods. [76, 77, 78, 79].

- **Word2vec:** Word2vec, a method for word expression that incorporates word meaning and context, was proposed by Mikolov et al. [80, 81] in 2013. It incorporates two learning algorithms: the skip-gram algorithm and the continuous bag-of-word **(CBOW)** algorithm. Numerous research, including those that use event detection, emotion analysis, and emotion classification [82, 83].

- **GloVe:** Another technique for creating vector representations for words is termed global vectors **GloVe**, which is an unsupervised learning approach. The model developed by Pennington et al. (2014) [84] in 2014 uses aggregated global word-word co-occurrence information from a corpus for training, and the representations that are produced demonstrate intriguing linear substructures of the word vector space. A number of research employ **GloVe** [85, 86] for event detection on tweets.

- **Fasttext:** developed by Bojanowski and colleagues (2017). Researchers [87] have created a new model based on the skip-gram model, in which a bag of n-gram characters represents each word. Each character in an n-gram has a vector representation, and words are the total of these representations. Numerous research, such [88, 89], use a fast text model.

3.6 Transformers

Transformers have revolutionized natural language processing (NLP) by enabling models to process and understand textual data with exceptional accuracy and efficiency. Leveraging mechanisms like self-attention and contextual embeddings, transformers such

as BERT, GPT, and RoBERTa excel at capturing the relationships between words and sentences, regardless of their positions in the text.

3.6.1 Transformer model

The transformer model consists of encoder and decoder blocks powered by a softmax activation function to normalize output probabilities. Input data is processed by embedding words and assigning positional vectors to capture their contextual meaning. The encoder block uses multi-head attention and feed-forward networks to compute attention vectors, which highlight relationships between words in a sentence. These vectors are passed to the decoder block, which integrates them with its masked multi-head attention layers to analyze relationships across the entire document. Finally, a linear layer and the softmax activation function convert the processed vectors into a probability distribution for the output. The multi-head attention layers enable parallelization, significantly improving efficiency in processing [90].

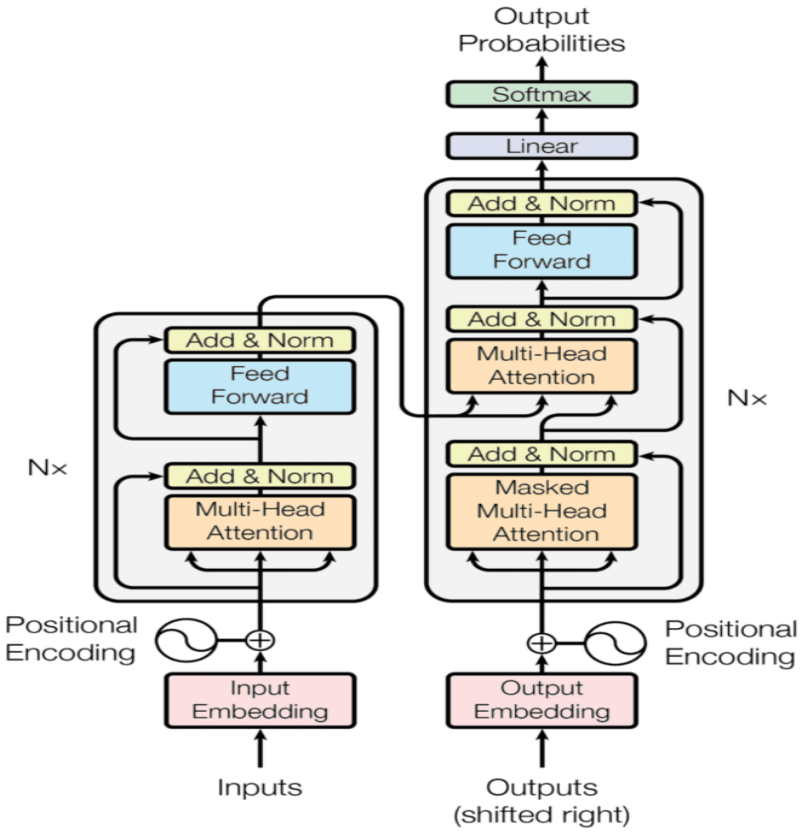


Figure 3.7: The Transformer - model architecture [91].

3.6.2 Pretrained Models: BERT

Pretrained Foundation Models (PFMs) play a crucial role in Artificial Intelligence (AI), particularly in the big data era. Introduced in [92], PFMs encompass a variety of models and functionalities explored in key AI domains like natural language processing (NLP), computer vision (CV), and graph learning. Known for their versatility, PFMs excel in tasks such as text classification, text generation, image classification, object detection, and graph classification. Their ability to learn from large datasets and adapt to smaller-scale tasks makes them invaluable for enhancing data processing workflows[93].

The BERT model, short for Bidirectional Encoder Representations from Transformers, is a widely adopted natural language processing model introduced by Google researchers in 2018[94]. Available in two configurations, Base and Large, it features varying levels of complexity: the Base version consists of 12 encoder layers with 110 million parameters, while the Large version includes 24 encoder layers and 340 million parameters . BERT has demonstrated remarkable effectiveness across diverse NLP tasks, making it an indispensable tool in the research community. Numerous studies have utilized BERT, including works such as [95, 96, 97]. Given its success and adaptability, BERT has solidified itself as a foundational model within the domain of natural language processing.

Following this development, specific versions of BERT have been introduced. The ALBERT model, presented by Lan et al. in 2019[98], is 'A Lite' version of BERT and represents an unsupervised language representation learning approach that is widely used. To develop the ALBERT model, researchers employed a parameter reduction technique to address memory limitations while ensuring the model's effectiveness without significant degradation. In contrast to English, Arabic presents challenges due to its rich morphology, limited resources, and relatively unexplored syntax. AraBERT is a model specifically designed for the Arabic language [99], with an architecture comparable to that of BERT. Antoun et al. (2020) introduced pre-trained AraBERT models, which are publicly accessible at `http://github.com/aub-mind/araBERT/`. Additionally, the Robustly Optimized BERT Pretraining Approach (RoBERTa) was introduced by Yinhan et al. in 2019 [100]. RoBERTa includes modifications such as training with larger batch sizes and longer sequences, as well as the removal of the Next Sentence Prediction (NSP) task to enhance pretraining performance.

DistilBERT

DistilBERT, a condensed and efficient variant of the BERT model, is trained using a distillation approach applied to the BERT base. It achieves a significant reduction in complexity by employing 40% fewer parameters while maintaining a vocabulary size

of 3,052,245. To optimize its performance, its hyperparameters are carefully calibrated throughout its design process. DistilBERT has been successfully utilized in numerous studies [101, 102].

ARAELECTRA

ARAELECTRA, an Arabic language representation model introduced in [103], employs a reduced model size without compromising performance. It is pre-trained using large Arabic text corpora with the replacement token detection objective. ARAELECTRA has been evaluated across various Arabic natural language processing tasks, such as named-entity recognition, sentiment analysis, and reading comprehension, and has been applied in multiple studies [104, 105].

CAMeLBERT

CAMeLBERT is a collection of BERT-based models pre-trained on Arabic texts of diverse sizes and formats. One notable variation is "Bert-base-arabic-camelbert-da-sentiment," specifically tailored for sentiment analysis tasks [106].

3.7 Functions of Activation

The activation unit is essential for regulating the outputs of neural cells in artificial neural networks.. It serves as a predecessor for the backpropagation algorithm. Differentiable activation functions are essential in backpropagation because they enable smooth transitions during weight updates, reducing the likelihood of issues that could hinder convergence. These functions are particularly important when dealing with complex, non-linear interactions, as they help map response variables to inputs effectively. However, training neural networks with multiple hidden layers presents several challenges. These challenges can include problems such as vanishing gradients, erratic weight updates, and overly complicated formulas, which can impede the learning process. To solve these problems, several activation functions such as Sigmoid, Tanh, ReLU, and SoftMax are frequently employed. [107].

3.7.1 The function of Linear Activation

A straight line defined by y=x is similar to a linear activation function [107]. The output of a neural network will always be a linear combination of the input if all of its layers use linear activation functions. The output's range will be between negative

and positive infinity. Usually, only the output layer receives a linear activation function. The network's capacity to discover intricate patterns is restricted when linear activation functions are used across the board.

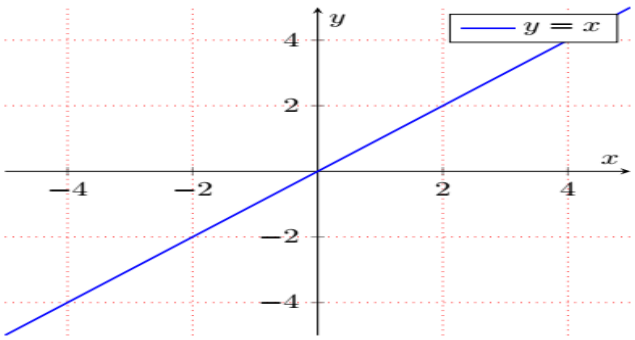


Figure 3.8: Function of Linear Activation

3.7.2 Non-Linear Function of Activation

In order for neural networks to extract complex patterns from data, non-linear activation functions are essential. Incorporating non-linearity, these functions empower the model to capture relationships that are not simply straight lines, allowing it to adapt and generalize across a wide variety of datasets. This flexibility helps the network differentiate between outputs more effectively. In essence, "non-linear" refers to the characteristic that the relationship between inputs and outputs cannot be accurately described as a straightforward linear combination, opening up a richer landscape for modeling complex phenomena [107] .

3.7.2.1 Sigmoid Function

The sigmoid is used as a neural network activation function in [108, 109]. This non-linear transformation is very helpful for binary classification problems because it compresses the output to an interval from 0 to 1. The sigmoid function's smooth and continuous traits, which guarantee that small variations in the input result in small changes in the output, are one of its main advantages. This characteristic makes it possible to efficiently update the neural network's weights during training, which is essential for gradient-based optimization techniques like backpropagation. All things considered, the Sigmoid Activation Function helps neural networks understand intricate patterns in data, which improves their performance. The following equation provides the sigmoid function formula [107]:

$$\boxed{\sigma(z) = \frac{1}{1 + e^{-z}}} \tag{3.23}$$

- Complex patterns cannot be accurately modelled by linear equations but rather by neural networks.
- They are helpful for binary classification since the output values fall between 0 and 1.
- When x values range between -2 and 2, the function exhibits a high gradient. This sensitivity shows that even minor adjustments to input x can result in significant changes to output y, which is important to know during training.

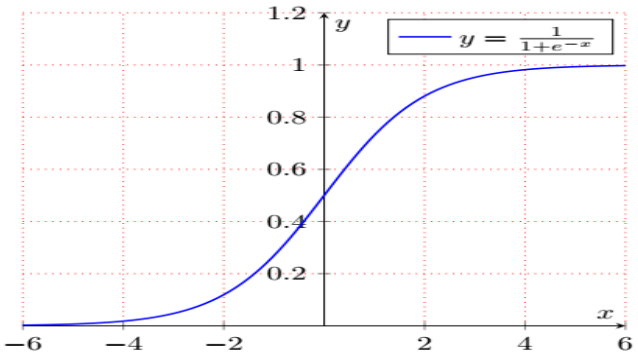


Figure 3.9: Sigmoid Function

3.7.2.2 Tanh

A shifted form of the sigmoid that can extend across the y-axis is the hyperbolic tangent function, often known as the tanh function. It is described as [107]:

$$\boxed{tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}}} \tag{3.24}$$

- **Interval of value:** Consistently, outputs vary between -1 and +1.
- **Non-linear:** Facilitates the modelling of intricate data patterns.
- **Utilization in Hidden Layers:** Frequently employed in hidden layers because of its zero-centred outputs, which promote easier learning for the following layers.

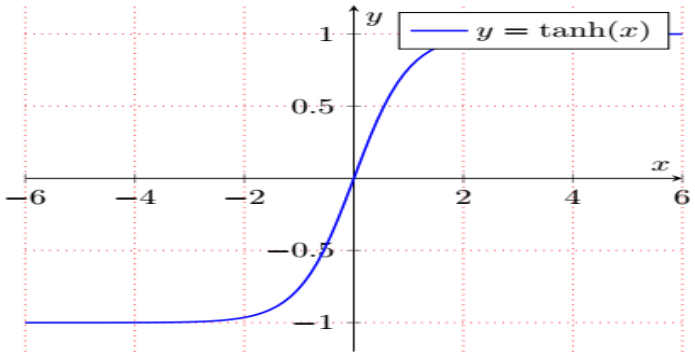


Figure 3.10: Tanh Function

3.7.2.3 Sigmoid and Tanh comparison

A shifted sigmoid that can extend over the y-axis is called a tanh.

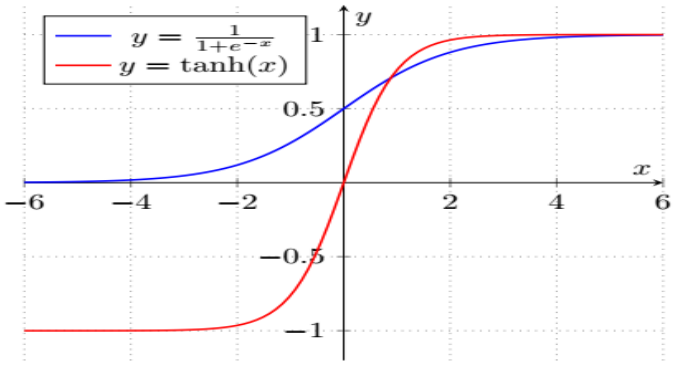


Figure 3.11: Sigmoid and Tanh comparison

3.7.2.4 ReLU (Rectified Linear Unit) Function

The formula for the ReLU (Rectified Linear Unit) activation function is A(x) = max(0, x. According to this definition, the output will be bigger than zero or x itself for any given input value x. If x is a positive number, the ReLU function will return the value of x, effectively allowing positive inputs to pass through unchanged. Conversely, if x is a negative number or zero, the function will output 0. This characteristic of the ReLU function helps introduce non-linearity into models while maintaining computational efficiency,resulting in its broad adoption in different neural network topologies. It's described as [107]:

$$\boxed{Relu(z) = max(0, z)} \tag{3.25}$$

- **Value Range:** [0, $+\infty$), which means the function only produces non-negative values.

- **Category:** Category: It is a non-linear activation function that increases back-propagation efficiency and helps neural networks learn intricate patterns more successfully.
- **The advantage over other Activation:** Because of its simpler mathematical procedures, this function is computationally less expensive than tanh and sigmoid. Just a small number of neurons are activated at once; the network is sparse, effective, and simpler to compute.

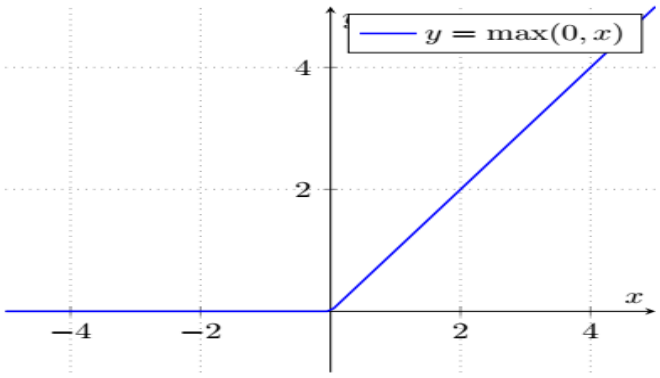


Figure 3.12: Rectified Linear Unit

3.7.2.5 Exponential Linear Units(ELU)

When compared to other activation functions, the Exponential Linear Unit (ELU) has a tendency to converge more quickly and yield more precise results. The addition of an alpha constant, which has to be a positive value, is a crucial component of ELU. For non-negative inputs, the Rectified Linear Unit (ReLU) and the ELU respond similarly to the identity function, making them quite similar. On the other hand, ELU operates differently than ReLU for negative inputs. In particular, ReLU transitions abruptly, but ELU approaches an output of -∞ gradually. The exponential linear unit (ELU) with $0 < \alpha$ is [110]

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ \alpha \times (\exp(x) - 1) & \text{if } x \leq 0 \end{cases} \tag{3.26}$$

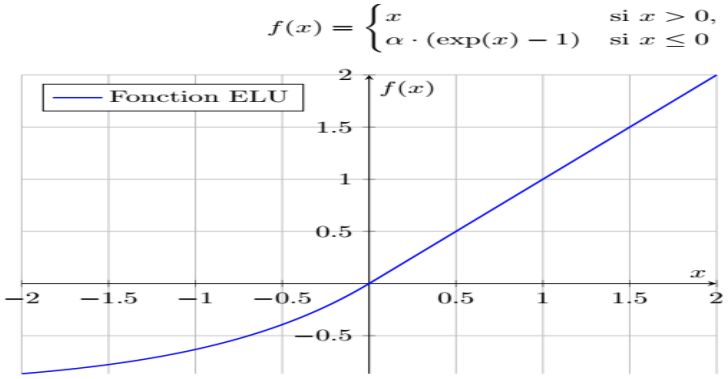


Figure 3.13: Exponential Linear Unit (ELU)

3.7.2.6 Softmax Function

Multi-class classification issues are the focus of the softmax function's design. It turns a neural network's raw output scores into probabilities. In order to ensure that the sum of all probabilities equals 1, this function compresses the output values for each class into a range between 0 and 1. It is described as [107]:

$$\boxed{\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad for\ i = 1, 2, \ldots, K} \tag{3.27}$$

- **Nature:** Softmax is a type of activation function that is non-linear..
- **Functioning:** The Softmax function assigns probabilities to each class, aiding in class identification.

3.7.2.7 SoftPlus Function

The Softplus function is a useful Neural network activation function because of its characteristics of always producing positive outputs and being differentiable at all points. This distinguishes it from the traditional Rectified Linear Unit (ReLU) function, which outputs zero for any negative input and can lead to non-differentiable points. that incorporates an extra parameter a > 0 and can be defined using the equation $\text{softplus}_a(x) = \frac{\log(1+\exp(ax))}{a}$ [111]. It is a simple form for a $= 1$ is defined as :

$$\boxed{A(x) = \log(1 + e^x)} \tag{3.28}$$

- **Category:** Non-linearity characterizes the Softplus function.
- **Interval values:** With the exception of ReLU's strict zero threshold, the function generates values in the range $(0, \infty)$.

- **Smoothness:** Unlike ReLU, which has rapid discontinuities that can occasionally lead to issues during optimization, Softplus is a fluid, continuous function.

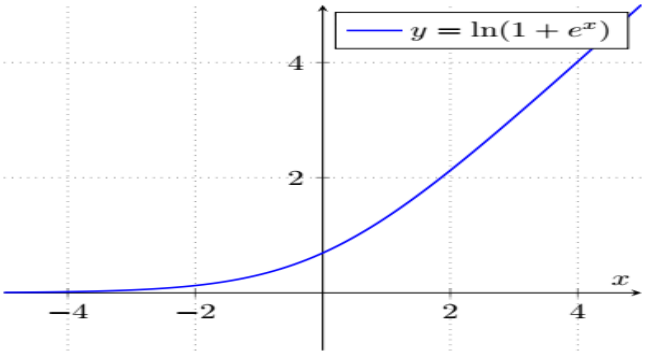


Figure 3.14: Softplus Function

3.8 Evaluation metrics

Metrics used to evaluate a model's efficiency or are known as evaluation metrics. These metrics, which offer quantitative and objective ways to assess something's performance, are frequently employed in a variety of industries and disciplines, including machine learning, data analysis, and research. The particular challenge and objectives determine the evaluation measures to be used. Different metrics are appropriate for different situations. In machine learning for tasks like binary or multiclass classification, the common evaluation metrics used are: accuracy, precision, recall, F1 score. By displaying the number of accurate and inaccurate predictions for each class in a classification task, it aids in evaluating how well a model is functioning. Four values are commonly found in a confusion matrix: The instances where the model accurately predicted the positive class are known as True Positives (TP); True Negatives (TN): These are instances in which the negative class was accurately predicted by the model; When the model predicted a positive class when the actual class was negative, this is known as a false positive (FP). When the model predicted a negative class when the actual class was positive, this is known as a false negative (FN). A useful method for evaluating a classification model's performance is the confusion matrix, which can be used to compute a number of performance metrics, including F1-score, recall, accuracy, and precision.

3.8.1 Confusion Matrix

A confusion matrix is a tool used in machine learning and statistical classification to assess how well a classification model performs. It provides a clear summary of the

model's predictions compared to the actual labels in the dataset, offering insights into both accurate and inaccurate classifications. In the context of binary classification:

- **True Positive(TP):** The model successfully predicts the positive class.
- **True Negative(TN):** The model accurately identifies the negative class.
- **False Positive(FP):**The model mistakenly predicts the positive class as the negative class.
- **False Positive(FN):**The model mistakenly predicts the negative class as the positive class.

There are different types of confusion matrices [112]:

- **Binary Confusion Matrix:** Specifically designed for two-class (binary) classification scenarios.
- **Multiclass Confusion Matrix:** Includes predictions across multiple classes.

3.8.2 Measuring Performance

Accuracy

Accuracy is one of the most crucial performance indicators for a machine learning classification model. It is the percentage of correctly predicted cases (including true positives and true negatives) in the dataset relative to all instances. Accuracy can be calculated using the following formula:

$$\boxed{\text{Accuracy} = \frac{T_p + T_n}{T_p + F_p + T_n + F_n}} \tag{3.29}$$

Here, true positives are represented by ***Tp***, true negatives by ***Tn***, false positives by ***Fp***, and false negatives by ***Fn***.

Precision

Precision is an important metric in machine learning, particularly for classification tasks. It is defined as the ratio of true positive predictions to the total number of the model's positive predictions. This metric indicates how many of the positive predictions made by the model are actually correct [113]. The formula for Precision is as follows:

$$\boxed{Precision = \frac{T_p}{T_p + F_p}} \tag{3.30}$$

In this context, ***Tp*** represents true positives, and ***Fp*** denotes false positives.

Recall

In simpler terms, recall measures how effectively the model identifies all positive instances. A high recall value indicates that the model successfully captures most positive cases while minimizing false negatives[113]. The formula for Recall is as follows:

$$\boxed{Precision = \frac{T_p}{T_p + F_n}} \tag{3.31}$$

In this context, ***Tp*** represents true positives, and ***Fn*** denotes false negatives.

F1-Score

$$\boxed{F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}} \tag{3.32}$$

F-measure

F-measure is a general term for a metric that combines precision and recall into a single score. It can be adjusted to give more importance to either precision or recall using a parameter called β [114]. For example, the F2-score emphasizes recall more than precision, while the F1-score is a special case of the F-measure with equal weighting for precision and recall.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{3.33}$$

Keyword Recall

This measures how well a system identifies the correct words or keywords for events. Specifically, it calculates the fraction of correctly detected keywords compared to the total number of ground truth (GT) keywords. Ground truth represents the actual, true keywords for events as determined by a reference or dataset [38].

$$\boxed{\text{Keyword Recall} = \frac{\sum_{t \in T} |W_d^t \cap GT^t|}{\sum_{t \in T} |GT^t|}} \tag{3.34}$$

where:

- T: Set of time windows during which events occur.
- W_d^t: Set of keywords detected by the system at time t.
- GT^t: Set of ground truth (relevant) keywords at time t.

- $|W_d^t \cap GT^t|$: Number of correctly identified keywords, which are present in both W_d^t (detected keywords) and GT^t (ground truth keywords) at each time window t.
- $|GT^t|$: Total number of ground truth keywords at each time window t, representing the total relevant keywords.

Silhouette Score

The Silhouette Score is a metric used to evaluate the quality of clustering by measuring how well data points fit within their own cluster compared to other clusters. It ranges from -1 to 1, where a score close to 1 indicates well-separated, cohesive clusters, and a score close to -1 indicates that points may be assigned to the wrong cluster. A score around 0 suggests overlapping clusters.

$$\boxed{S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}} \tag{3.35}$$

Where: a: The average distance of the data point to all other points in the same cluster (intra-cluster distance).
b: The average distance of the data point to points in the nearest cluster that it is not a part of (inter-cluster distance).

Mutual Information (MI)

quantifies the dependency between two variables, measuring how much knowing one variable reduces uncertainty about the other. The formula involves probabilities of joint and individual occurrences [115].

$$\boxed{MI(X,Y) = \sum_{y \in Y} \sum_{x \in X} \log \left(\frac{p(x,y)}{p(x)p(y)} \right)} \tag{3.36}$$

Dunn Index

The Dunn Index is a metric used to evaluate the quality of clustering in data analysis. It is defined as the ratio of the minimum inter-cluster distance (the smallest distance between points in different clusters) to the maximum intra-cluster distance (the largest distance within a single cluster) [116]. Mathematically, it can be expressed as:

$$\boxed{D = \frac{\min d(i,j)}{\max d'(k)}} \tag{3.37}$$

Purity

This metric evaluates the accuracy of clustering by assigning each cluster to the class that is most frequent within it and then measuring the proportion of correctly assigned data points. A higher purity indicates better clustering performance. Mathematically, purity can be expressed as [117]:

$$\boxed{\text{Purity} = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r)} \tag{3.38}$$

Where: - n_r: Number of data points in cluster r. - n: Total number of data points. - $P(S_r)$: Purity of cluster r. - k: Total number of clusters.

In clustering, **average purity** typically refers to the mean purity across all clusters in a solution. It evaluates how well the clusters align with the ground truth classes, providing an overall measure of clustering quality. To calculate average purity:
- Compute the purity for each cluster individually. - Take the average of these purity values across all clusters.

Keyword-Precision@K (K-Precision)

K-Precision is a metric used to evaluate the accuracy of retrieved keywords in tasks like event detection or natural language processing. Specifically, it measures the proportion of relevant keywords correctly identified among the top-K retrieved keywords. It is the fraction of relevant keywords within the top-K results retrieved by a system, defined as [118]:

$$\boxed{K\text{-Precision} = \frac{\text{Number of relevant keywords in the top-}K}{K}}$$

Perplexity

Perplexity is a concept frequently used in language modelling and natural language processing (NLP). It measures how well a probabilistic model predicts a sample of data. Think of it as a way to determine how "confused" or "uncertain" a model is when making predictions. Lower perplexity means the model is better at predicting text, while higher perplexity means it struggles more [119].

$$\boxed{\text{Perplexity}(D_{\text{test}}) = \exp\left(-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\right)} \tag{3.39}$$

Where:

- M: The total number of documents in the dataset.
- w_d: The set of words in document d.
- $p(w_d)$: The probability assigned to the words in document d by the model.
- N_d: The total number of words in document d.

3.9 Conclusion

This chapter has provided a comprehensive exploration of computational methods, spanning similarities-based techniques, machine learning approaches, and deep learning innovations. Additionally, the focus on activation functions and evaluation metrics highlights the intricate mechanics that drive model functionality and effectiveness. By synthesizing these elements, the chapter offers a well-rounded understanding of how advanced systems are designed, optimized, and evaluated.

Chapter 4

Literature Review

Contents

4.1 Introduction

The chapter examines event detection and classification, focusing on the datasets and techniques used to identify events in data streams. It categorizes datasets into those sourced from Twitter and general domains, highlighting their relevance to detection strategies. The chapter also discusses specified and unspecified event detection methods while presenting a review of state-of-the-art advancements in the field, providing a comprehensive overview of progress in event detection research.

4.2 Event Detection Datasets

In this section, we examine the currently dataset for ED:

4.2.1 Datasets from Twitter

Twitter datasets are essential for research in machine learning, natural language processing, event detection, and classification. Researchers can access a diverse collection of publicly available Twitter datasets covering various topics and events, facilitating automated event identification and categorization.

4.2.1.1 Event collections

Event2012 corpus , provided by McMinn et al. [120], comprises 120 million tweets gathered from October 9 to November 7, 2012. This extensive corpus serves as a valuable resource for examining social media dynamics and user engagement during that period. It has been employed in various Twitter-based event detection applications [121, 122]. From this dataset, distinct event types were selected for further analysis, allowing for insights into social media responses to specific events. It has been used for examples in [123, 124, 125, 126].

4.2.1.2 EveTAR collections

A collection of multi-task Arabic tweet tests is called EveTAR. EveTAR can be used for four distinct tasks over Arabic tweets: event detection, ad-hoc search, timeline building, and real-time summary. There are 50 subjects signalling important events, and 355 million Arabic tweets crawled in January 2015 [127]. EveTAR has been used in this study [128].

4.2.2 Datasets from the general domain

In 2020, the large event detection MAVEN dataset (Wang et al., 2020) [129] was created by combining human annotation and machine generation using 4,480 Wikipedia articles. It seeks to overcome the shortcomings of current ED databases, including their lack of data and inadequate coverage of event categories. The frames specified in the linguistic resource Frame net (Baker et al., 1998[130]), which covers a wide range of events in the general domain, are the source of the event categories in **MAVEN**. A bigger data size and event coverage are shown by the fact that MAVEN covers **168** event categories and 118,732 event mentions when compared to previous datasets. MAVEN dataset used in these studies [131, 132]

4.3 Event Detection techniques

4.3.1 Specified Event Detection *(SED)*

The detection of planned or so-called specified events detection [44, 133]deals with events whose several properties are predefined such as time, location, subject ect. Predefined data and features that are anticipated to show up in the data to represent an event are processed by SED. The methods outlined later use a variety of machine learning (ML), data mining, and text analysis approaches to try to leverage Twitter textual content, metadata information, or both. In their research on Controversial Events on Twitter, Ana-Maria Popescu and Marco Pennacchiott[134] used Gradient Boosted Decision Trees (GBDT) as the ML framework. By adding the event detection score as an extra feature to the controversial model, the authors have also suggested combining the two stages (detection and scoring) into a single-stage system, which has increased performance.

4.3.2 Unspecified Event Detection *(UED)*

The best data source for identifying unknown or so-called unplanned events is user-posted tweets. Usually, unknown events are found by making use of the temporal patterns or signals of Twitter streams [44, 133]. Analyzing the temporal features of the Twitter stream by tracking bursts to find significant keywords and concepts to highlight events is a basic method for UED. In their work on Characterizing Emerging patterns on Twitter, Hila Becker and Luis Gravano[135] concentrated on describing the patterns that may be identified on Twitter using current baseline methodologies. They gathered identified patterns for this from two distinct sources. They start by gathering local trends that Twitter finds and post every hour. The trends are accessible through an application programming interface (API)from the Twitter service. Secondly, they use a basic burst-

detection algorithm on a sizable Twitter dataset in order to find more trends to supplement and broaden the trends that Twitter provides.

4.4 Approaches for event detection

In online social media like Twitter, event detection is conceptually similar to clustering, where systems process time-ordered documents and group them into relevant event-based clusters. However, the unique nature of OSM introduces several challenges, including massive document volume, the prevalence of everyday posts that must be filtered out, and the noisy, short-text format that often contains spelling and grammar errors. Several approaches have been explored for event detection in OSM, these approaches including temporal approaches, topic modelling, incremental clustering, graph theory, rule mining and bursty event detection [136]. Other authors have proposed different approaches for event detection, such as [137]:

1. **Clustering Based Approaches:** These are widely used for event detection in social media data streams. These methods aim to group similar data points (e.g., tweets, posts) into clusters based on shared characteristics, helping identify patterns and emerging events. Some common types of clustering approaches include:

- **Incremental Clustering:** Continuously updates clusters as new data arrives. Example: Becker et al.[138] used this method for detecting Twitter events.
- **K-Means Clustering:** Divides data into pre-defined clusters by minimizing variance within each cluster. Example: McCreadie et al. successfully applied it for event detection [139].
- **Hierarchical Clustering:** Forms a tree-like structure of clusters, merging similar ones iteratively. Example: Corney et al. [140] clustered word n-grams; Li et al. [167, 38] grouped semantic terms.

2. **Neural Network-Based Approaches :** Recent advancements in deep learning and supervised learning have significantly improved event detection and classification in social media, particularly on Twitter. These approaches leverage powerful models to enhance accuracy and efficiency [136, 141, 142].

A taxonomy in event detection, presented by Panagiotou et al. [40], organizes related work based on fundamental data mining techniques. These include clustering, outlier detection, and classification, which form the backbone of various approaches. The taxonomy helps categorize algorithms by their methodology, whether they focus on stream clustering, anomaly detection, or pivot-based techniques. It also distinguishes between

supervised and unsupervised methods, where supervised models use historical data for training, and unsupervised approaches rely on scoring functions. Essentially, the taxonomy provides a structured way to understand the different strategies used in event detection research.

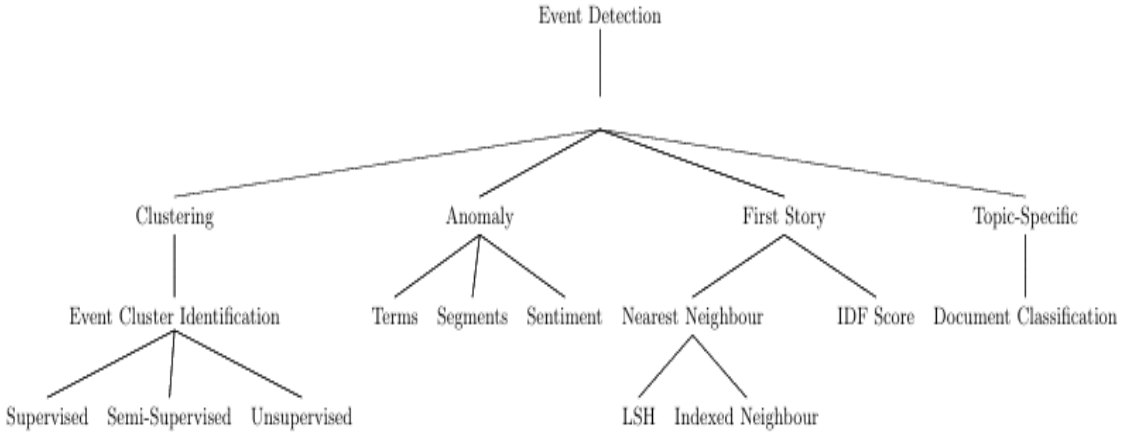


Figure 4.1: An abstract taxonomy of event detection approaches. [40]

4.5 Different techniques and methods proposed for event detection and classification on OSM

This section presents various approaches proposed for Twitter event detection and classification. Table 4.1 summarizes the techniques employed by researchers for event detection, along with the datasets used and the best performance achieved in each study.

Adel et al. introduced a structured approach to event detection and classification on Twitter [143], encompassing five key phases: data collection, preprocessing, detection, classification, and evaluation. Initially, tweets are gathered from various sources to construct a comprehensive dataset. The preprocessing stage then refines the raw data by eliminating noise, irrelevant elements, and stopwords, thereby enhancing model performance. The event detection process determines whether a tweet contains event-related information, followed by classification, in which machine learning and deep learning techniques categorize detected events into predefined domains such as sports, politics, natural disasters, and entertainment. Finally, the evaluation phase assesses the model's effectiveness using F1-score.

Pradhan et al. [144] introduce EDT_BERT, an advanced tweet-processing method for event detection and evolution tracking. This approach constructs a graph wherein tweets are nodes, and their relationships, based on overlapping hashtags, named entities, and semantic analysis from a pre-trained BERT model, constitute the edges. Event clusters are identified through graph clustering, and their temporal evolution is analyzed using the Maximum-Weight Bipartite Graph Matching (MWBGM) algorithm.

Prasad et al. [145] utilize Twitter API data to detect and classify transportation disasters in Nigeria by analyzing user, location, and timestamp metadata. Their approach, integrating the BERT model with the AdamW optimizer, achieved an 82% accuracy rate, underscoring its effectiveness in real-time disaster classification.

Kersten et al. [141] investigate how Deep Learning techniques enhance disaster management by analyzing tweets for disaster-related information. The study employs models such as RNN, CNN, BLSTM, and Transformer architectures (BERT, RoBERTa, DeBERTa) for two key tasks: identifying disaster-relevant tweets and classifying them into eight disaster types (e.g., earthquakes, floods). The research combines accessible datasets for training while exploring effective preprocessing strategies and bias mitigation methods to improve performance.

Hansi Hettiarachchi et al. [38] proposed Embed2Detect, a novel method leveraging hierarchical clustering and word embeddings. The approach employs Skip-gram text representation to maximize the average log probability of context terms, using a three-layer neural network as a log-linear classifier. Hierarchical Agglomerative Clustering (HAC) is applied for data clustering. Embed2Detect consists of four primary components: an event window identifier, word embedding learner, stream chunker, and event word extractor.

Kersten et al. [146] proposed a Twitter stream clustering workflow that extracts tweets for each interval (t_n) across successive time periods (t_{n-1}, t_n, t_{n+1}). After preprocessing, Chinese Restaurant Process (CRP)-based clustering is performed. A distinguishing feature of this workflow is merging clusters from the same and different time periods based on cosine distance. Temporal criteria are also applied for cluster pruning. Two experiments were conducted: the first assessed clustering performance using a subset of Events2012, while the second evaluated clustering based on cluster purities.

Jeffery Ansah et al. [147] proposed SensorTree, a framework for detecting protest events by leveraging structural connections within user networks to identify bursts in online communities. The method tracks the growth of propagation trees, constructed from tweets and Twitter follower networks, to model community expansion dynamics over time.

By storing tree size and capturing burst periods, SensorTree seeks to infer the events driving these bursts. Using follower relationships, time metrics, and tweet content, it creates semantic propagation trees through four steps: Instantiation, TreeGrowth, NewCom, and TreeTerminate. Additionally, two algorithms, TreeConstruct and EventExtract, are implemented for efficient event detection.

Dhiman and Toshniwal [115] proposed a model for Twitter data analysis, representing it as a sentence graph to balance contextual relationships with computational efficiency. Their approach involves tweet preprocessing, feature representation generation using Joint Spherical Embedding (JoSE) to capture textual semantics via vector directions, and the creation of a weighted graph to illustrate relationships. Semantic and temporal similarities are integrated with MCP Clustering [148] to form robust clusters. Additionally, the authors provided a pseudo-code algorithm for an event detection model, offering deeper insights into Twitter data dynamics.

Pandya et al. [125] introduced MaTED (Metadata-assisted Twitter Event Detection), an extension of the Twevent framework [149]. Their system comprises four modules: (1) Identifying Important Phrases from Tweets, which involves extracting text using JSON and generating keywords with DBPedia Spotlight [150]and WordNet; (2) Extracting Bursty Phrases by applying a burstiness probability formula; (3) Clustering Bursty Phrases, following the method proposed by Chenliang Li, Aixin Sun, and Anwitaman Datta [149]; and (4) Event Characterization, where events are defined as collections of related terms, visualized using the MABED approach [151].

DSTTM (Dynamic Spatio-Temporal Tweet Mining), proposed by Mahdi Farnagh et al. [152], is a method for extracting events from real-time geotagged Twitter data. It addresses spatial heterogeneity by calculating spatial-temporal distances and text similarity for spatiotemporal clustering. The method utilizes advanced NLP techniques and embedding models, including Word2Vec [153], GloVe [154], and FastText [87], to vectorize short Twitter messages. It employs the OPTICS density-based clustering algorithm to compute cluster order, alongside the clustering extraction algorithm by Schubert and Gertz [155]. DSTTM effectively adapts to tweet density variations across locations to handle the heterogeneity of large-scale geotagged Twitter data.

Sehaa, a big data analytics system for healthcare proposed by Shoayee Alotaibi et al. [21], comprises four modules to analyze health-related Twitter data from Saudi Arabia. The Data Collection Module employs a Twitter streaming API and the Sehaa JSON Parser Algorithm to capture and store tweets. The Pre-Processing Module cleans and labels the data using the Sehaa Pre-Processing Algorithm. The Classification Module

integrates six classifiers, including Naive Bayes and Logistic Regression, alongside feature extraction methods such as BiGram, TriGram, HashingTF, and CountVectorizer, in a two-stage process. Lastly, the Validation Module assesses classifier performance to ensure accuracy. This system effectively identifies healthcare trends and patterns.

The daily generation of geo-localized tweets has driven the search for advanced techniques to analyze latent knowledge and spatial models in various contexts. Zeinab Ghaemi and Mahdi Farnaghi [116] introduced VDCT (Varied Density-based spatial Clustering for Twitter data), an extension of VDBSCAN, to detect and extract geolocated events from Twitter data in the presence of spatial heterogeneity. Tweets are represented as tuples [x, y, c, l], where x and y are geographical coordinates, c is the tweet's textual content, and l is the cluster label. Cosine similarity is used to measure text similarity, while VDCT integrates both textual content similarity and spatial proximity to form clusters.

The Weighted Dynamic Heartbeat Graph (WDHG) approach, introduced by Zafar Saeed et al. [118], identifies events in text streams by transforming Twitter text streams into temporal graphs. Aggregated micro-documents are combined to form super-documents, which generate graphs based on word co-occurrence. These graphs are converted into WDHGs from adjacent pairs. Features such as aggregated centrality and growth factor are extracted, with a rule-based classifier using aggregated centrality to label WDHGs as event candidates. Event-related topics are then extracted by merging these candidates, while a heartbeat score calculated by multiplying aggregated centrality and growth factor provides further insights.

Ahmad Hany Hossny and Lewis Mitchell [156] propose a method utilizing a spike detection temporal filter to transform daily count vectors of word pairs into binary form. These binary vectors are compared with event occurrence binary vectors using the Jaccard metric, selecting the top n-word pairs with the highest similarity as predictive features for event-associated days. The method evaluates these features' effectiveness in event prediction using classifiers such as Decision Trees, Naive Bayes, KNN, SVM, and Logistic Regression.

Sit et al. [28] present a methodology for identifying, classifying, and analyzing disaster-related tweets across different storm phases. Geo-located tweets were collected during Hurricane Irma using eleven hurricane-related keywords and a spatial bounding box. Classification methods included logistic regression, SVM, CNNs, LSTM networks, and ANNs, while LDA was employed to extract insights such as donations and affected individuals. The DBSCAN algorithm was utilized to identify areas of prominence and impact.

Ahmad Hany Hossny et al. [157] propose a model to predict event occurrence using tweet features, including words, n-grams, skip-grams, and bags-of-words. These features are transformed into daily time-series vectors and correlated with event time-series data through a five-step process: (1) Singular Value Decomposition (SVD) ensures feature independence; (2) k-means clustering groups orthogonalized features based on Euclidean distances; (3) a lookup table maps features within clusters to their centroids; (4) the mapping is applied to the original data; and (5) correlation scores are recalculated to strengthen the association between textual features and events.

Yuqian Yuqian Huang, Yue Li, and Jie Shan (2018) [158] focused on detecting small-scale spatial-temporal events and their textual content. They proposed a three-stage workflow for event detection: (1) grouping tweets by day and generating charts to visualize the number of tweets and users per day; (2) clustering daily tweets using ST-DBSCAN, an algorithm for spatial-temporal data clustering developed by Derya Birant and Alp Kut (2007) [159], and generating spatial, temporal, and textual patterns for each cluster; (3) employing Latent Dirichlet Allocation (LDA) to identify potential topics within clusters and analyze the structure of each tweet.

Oystein Repp and Heri Ramampiaro [142]] utilized Artificial Neural Networks (ANNs) for text classification, employing a convolutional neural network architecture and Long Short-Term Memory (LSTM) networks. Latent Dirichlet Allocation (LDA) was applied to extract information such as donations and affected individuals, while DBSCAN identified prominence and impact areas for each category. Words were lower-cased using AvgGloVe, with user tags and hashtags separated. The study utilized Keras with Theano, incorporating dropout in a fully connected neural network [160] and Rectified Linear Units (ReLU) for detecting news-relevant tweets. Additionally, two alternative extensions of the Petrović et al. event clustering method [161] were proposed.

Muskan Garg and Mukesh Kumar [162] proposed TWCM (Twitter Word Co-occurrence Model), an event detection technique for analyzing uncertain social media data. The method involves pre-processing Twitter feeds by removing special characters, stop words, emoticons, hashtags, and @-mentions. A directed graph is constructed, where each word is a node and adjacent words within tweets form connections in a word co-occurrence network. Edge weights represent the frequency of co-occurrences between word pairs. Edges with the highest weights are identified to extract frequent biterms, retaining only edges with weights exceeding the threshold ($k = \sqrt{\text{number of edges}}$). The lowest-weight edges are removed in each cycle, and topological sorting is used to derive event phrases.

Shih-Feng Yang and Julia Taylor Rayz [117] proposed an event detection method leveraging hashtags in tweets. They utilized the K-means clustering technique [163] and adopted feature extraction from STREAMCUBE [164]. Tweets were collected using Tweepy APIs, capturing attributes such as creation time, retweet count, text content, hyperlinks, hashtags, and location. During preprocessing, tweets were stemmed, lowercased, and stripped of special characters, stop words, and hyperlinks, while hashtags were extracted as unigram features and removed from original messages. The HASHTAG-CLUSTER-STATIC and NEAREST-NEIGHBOR algorithms were employed for hashtag clustering, with distances between hashtags calculated during the clustering phase.

Edouard et al.[165] proposed a method to identify event-related tweets and classify them into categories through a three-step framework: preprocessing for data cleaning, Named Entity replacement for enhanced recognition and linking, and a classification step for categorizing tweets into topics such as Sports and Politics. They introduced two configurations—one involving a sequential classification where tweets are first filtered for relevance before categorization, and another where tweets are directly classified into predefined categories without initial filtering.

Tweet-SCAN, an event discovery technique developed by Joan Capdevila et al. [166], builds upon the DBSCAN algorithm (Density-Based Spatial Clustering of Applications with Noise). The method analyzes three parameters: space, time, and text—separately, assigning specific metrics to each dimension based on their data types. Spatial and temporal dimensions utilize Euclidean distance, while text analysis employs Jensen-Shannon (JS) distance, a suitable metric for probability distributions (Endres and Schindelin, 2003).

Chen et al. [167] present a novel real-time method for detecting events on social media, particularly Twitter. Events are characterized by 'who,' 'what,' 'where,' and 'when,' with tweets often containing terms corresponding to these aspects. The method introduces semantic categories, grouping terms into classes such as named entities, mentions, locations, hashtags, verbs, nouns, and links. These semantic classes enhance the identification and clustering of tweets related to the same event through a similarity calculation process.

Nasser Alsaedi, Pete Burnap, and Omer Rana (2017) [168] developed an event detection framework with five primary components: data collection, preprocessing, classification, online clustering, and summarization. The Naive Bayes classifier was employed for classification, treating each word as a feature due to its simplicity and efficiency, avoiding complex iterative parameter estimation. Temporal, geographical, and textual variables

were defined for the online clustering technique, which distinguished event-related documents from unrelated postings. The authors also introduced a TF-IDF weighting method for new document analysis.

Mahmud Hasan and colleagues [123] introduced Twitter New, a real-time evolutionary system for detecting news events on Twitter using incremental clustering. The system operates in two phases: first, tweets undergo pre-treatment, and the model determines if an input tweet pertains to a previously encountered topic by combining Locality Sensitive Hashing (LSH) with a Random Indexing (RI)-based term vector model. In the second phase, clusters are created using a generic incremental clustering algorithm, with dynamic expiration times determined by subsequent tweet arrival times. A series of filters is then applied, and significant events are identified using the Longest Common Subsequence (LCS) method, along with a representative tweet for each event cluster.

Deyu Zhou et al. [169] proposed the Latent Event Extraction and Visualization (LEEV) model, a probabilistic approach for collaborative event extraction and visualization on Twitter. In LEEV, each event is associated with a coordinate (ϕ_e), while each tweet (w_m) is mapped to a latent coordinate (x_m) in the visualization space. The model utilizes normalized Euclidean distance to assess the probability of a tweet (w_m) belonging to the event (e). To remove erroneous occurrences, the correlation coefficient of each event element is calculated, and elements with coefficients below a defined threshold are eliminated.

ATSED (Automatical Targeted-domain Spatiotemporal Event Detection), introduced by Ting Hua et al. [170], is a semi-supervised approach for detecting spatiotemporal events on Twitter. The system automatically generates labels from historical Twitter data, which are used to train a classifier specifically designed for Twitter data analysis. The trained classifier is then applied to real-time Twitter streams to identify event-related tweets. The Automatic Label Generation (ALG) algorithm extracts feature terms from news reports, ranks tweets by similarity, and identifies features relevant to specific events in the targeted domain. Domain weight and event weight are defined to calculate text similarity, alongside spatial and temporal similarity. An EM-based inference algorithm is employed to resolve "inference dependency" problems. The authors also proposed social-ties clustering and mini-tweet-group classification, clustering tweets into mini-groups based on social ties and applying SVM-based classification to these mini-groups.

Table 4.1: Summary of most important related works

Ref	Dataset	Detection Techniques	Best Performance
[143]	Tweet collection	Deep learning	**F1-score** 89.4%
[144]	FA Cup Super Tuesday US Election	EDT_BERT	**F-measure** 85.0%
[145]	Tweet collection	BERT-AdamW	Accuracy 82.0%
[141]	Tweet collection	XLM-RoBERTa	Recall 92.83 %
[38]	MUNLIV(sports) BrexitVote(politics)	Embed2Detect	Keyword recall 98.5%
[146]	Events2012	Chinese restaurant-based clustering(CRP)	Average Purity 0.34
[147]	Freeport Dataset	SensorTree	Precision 85.0%
[115]	RepLab 2013	MCP Clustering	Mutual Information 0.9857
[125]	Event2012	MaTED	Recall 82.0%
[152]	Hurricane Florence	DSTTM	Silhouette score 0.561
[21]	Streaming	Sehaa	Accuracy 86.7%
[116]	Hurricane Florence	VDCT	Dunn Index 0.721
[118]	FA Cup Super Tuesday US Election	WDHG	Keyword-Precision 75.0%
[156]	Collected tweets	KNN, SVM,naive Bayes and decision trees	F1-score 0.790
[28]	Hurricane Irma	LSTM	F1-score 0.751
[157]	Collected Tweets	Decompose-Cluster-Map	Mutual Information 0.873
[158]	Collected Tweets	ST-DBSCAN	**Perplexity** 35.5
[142]	Collected Tweets	Clustering algorithm	F1-score

Table 3.1 continued

Ref	Dataset	Detection Techniques	Best Performance
			0.818
[162]	Collected Tweets	TWCM	Recall 0.844
[117]	Collected Tweets	Clustering K-means	Average purity 0.772
[165]	Events 2012 First Story Detection	RNN	Average purity 88%
[166]	Collected Tweets	Tweet-SCAN	-
[167]	Live Tweet stream	Clustering Algorithm Based on Semantic Classes	Precision 0.963
[168]	Live Tweet stream	Online Clustering Algorithm	F1-score 0.854
[123]	Events2012	TwitterNews	Recall 0.870
[169]	First Story Detection (FSD)	LEEV	**F-measure** 0.898
[170]	Twitter dataset	ATSED	F1-score 0.870
[171]	Tweets Stream	GEOBURST	Precision 0.37

4.6 Conclusion

Chapter 4 has provided an in-depth exploration of event detection and classification techniques by examining relevant datasets from Twitter and general domains, analyzing specified and unspecified detection methods, Approaches and reviewing state-of-the-art advancements.

Chapter 5

Frameworks and Experimentations

Contents

5.1 Introduction

In this chapter, we examine the practical details of the solutions proposed in this thesis for event detection and classification, from dataset building to AI-based models developing. Starting with the construction of datasets and the development of models for event detection and classification. Starting with the building of a dataset, we explore methods for collecting tweets, delivering datasets for event detection, and analyzing dataset statistics. After building datasets, we introduce supervised learning-based models, with a detailed examination of models such as BERT, LSTM and hybrid approaches. Finally, we present unsupervised learning-based models for English and Arabic datasets, along with the experiments and evaluations conducted.

5.2 Building a Dataset

To build datasets for event detection and classification, we applied collection and adaptation methods. The collection process begins by gathering tweets related to specific events using keywords associated with event types. In the adaptation phase, selection and combining play a crucial role in refining the data for effective event classification. Selection ensures high-quality and targeted information by filtering data based on specific keywords or event categories. Meanwhile, combining integrates multiple datasets, aligns formats, and refines labels to create a more comprehensive and diverse dataset.

5.2.1 Collecting Tweets for the Dataset

To assess our event detection and classification models, we begin by building related datasets. To this end, TweetsEvents datasets were collected.

1. ***TweetsEvent:***A new dataset was collected by scraping tweets related to specific events. To classify events, we opted for 50 event types based on Fillmore's theory of semantic frames[129]. This categorization encompasses the majority of events, enhancing the model's classification performance compared to the 168 event types in the MAVEN dataset. The large number of event types in MAVEN reduces the model's performance. The 50 selected types of events are shown in Table 5.1. An algorithm (see Algorithm 1) was specifically designed to scrape the tweets. The collected tweets include three key pieces of information: the type of event, the content, and the date; the operation concluded with the building of a dataset consisting of

29,728 tweets. This dataset was used in experiments to detect and classify events (see section 5.3) [6].

Table 5.1: List of 50 Selected Event Types for Tweet Collection

Accidents	Arranging	Arrest	Arriving
Arts and Culture	Attack	Building	Business and Economy
Catastrophe	Change	Choosing	Collaboration
Competition	Conquering	Convincing	Creating
Crime	Cure	Damaging	Death
Defending	Departing	Destroying	Discovery news
Education Teaching	Elections	Exchange	Financial News
Health	Human Rights	Innovation and Technology	International Relations
Miscellaneous news	Politics	Protest	Publishing
Releasing	Religion	Revenge	Robbery
Science and Technology	Sending	Social event	Sports
Supporting	Terrorism	Theft	Transport
Travelling	Violence	-	-

2. ***TweetsEvent310:*** We used the same algorithm (see Algorithm 1) to build another dataset containing 310,000 tweets for the same event types listed in Table 5.1.

Algorithm 1 Tweet Scraping

```
Require: Events = Selected keywords such as 'Accidents', 'Attack', 'Building', etc.
  data = DataFrame(columns =['Type', 'Content', 'Date'])
Require: Limit the number of tweets to scrape by event type: Limit
  for event in in Events do
      query = event
      i = 0
      for tweet in s.TwitterSearchScraper(query) do
          if i == Limit  then
              break
          else
              data=data.append ('Type':query, 'Content':tweet.content,
              'Date': tweet.date)
              i = i+1
          end if
      end for
  end for
```

Table 5.2: Statistics of the TweetsEvent and TweetsEvent310 Datasets

Period	***# of tweets***	***Dataset Name***
From Sep. 08, 2022 to Sep. 11, 2022	29,728	TweetsEvent
From Sep. 08, 2022 to Dec. 23, 2022	310,000	TweetsEvent310

5.2.2 Adapting Datasets for Event Detection and Classification

1. **TweetsFinal dataset**: A new dataset, titled TweetsFinal, was compiled from disaster-tweets https://www.kaggle.com/datasets/vstepanenko/disaster-tweets)from kaggle repository. This dataset consists of 20 specific keywords that were extracted from a comprehensive analysis of tweets. These keywords were selected to capture the most relevant themes and topics present in the Twitter conversations. The TweetsFinal Dataset contains around 1000 tweets associated with disaster keywords such as "crash", "quarantine", and "bush fires" as well as the location and keyword itself.

2. **Event35 dataset** : To enhance model performance, we selected 35 types of events. Event2012 dataset contains a wide range of event categories, which can decrease the quality of model performance.

3. **Maven39 Dataset:** A significant works on event detection and classification have focused on publicly available datasets such as MAVEN, in order to increase model performance we selected 39 from **MAVEN** [129] using agorithm 2.

4. **Maven66 dataset:** To increase the range of event types we selected 66 types from **MAVEN** [129] using agorithm 2.

5. **MavenEvent66 dataset** To enhance the model performance, the MavenEvent66 dataset is obtained by combining two datasets, the Event2012 and MAVEN datasets, with a selection of 66 event types. This process involved mapping the data to ensure compatibility and refining it to remove any errors or redundancies.

6. **MavenEvent70 dataset:** This dataset is obtained through the same process used to build MavenEvent66, by changing the types of events and increasing the number of events.

7. **Tweets-data dataset:** This dataset contains Twitter data related to disaster events derived from tweet-data-clean(https://datasets.omdena.com/dataset/twitter-data-on-disaster-related-tweets). The types of disasters selected are: 'collapse', 'cyclone', 'meteor', and 'meteorite'.

8. **Arabic_Dataset_Classifiction dataset:** The dataset is organized into five distinct categories: sports, politics, culture, economy, and divers https://www.kaggle.com/datasets/saurabhshahane/arabic-classification. This classification provides a structured way to analyze and evaluate how Arabic text classification models perform across different domains. We used a subset of this dataset in our experiments.

Algorithm 2 Maven Module: Processing JSON events and creating a DataFrame

Import necessary libraries
Open the JSON lines file
Read all lines from the file
Initialize an empty list for storing the results
Initialize a counter `j` to 0
while `j` is less than the number of lines **do**
    Read the `j`th line
    Parse the line as JSON
    Extract the events from the JSON
    Initialize a counter `i` to 0
    while `i` is less than the number of events **do**
        Get the sentence ID of the `i`th event
        Retrieve the sentence using the sentence ID
        Append the event type ID, event type, and sentence to the results list
        Increment `i` by 1
    end while
    Increment `j` by 1
end while
Create a DataFrame from the results list
Save the DataFrame to a CSV file

5.2.3 Dataset Statistics

This table provides a summary of all the datasets used in our research project. Each entry includes documentation of the dataset's name, source and the total number of event types associated with it.

Table 5.3: Dataset Overview: Event Types, Sizes, and Sources

DataSet	**Event types** or **Class**	**Size**	**Source**
TweetsEvent310	50	310.000	Tweet collection (Algorithm 1)
TweetsEvent	50	29.728	Tweet collection (Algorithm 1)
TweetsFinal	20	996	Kaggle
Events35	35	8489	corpus Event2012 [120]
Maven49	49	15304	MAVEN [129]
MavenEvent70	70	23795	MAVEN[129] and Event2012 [120]
Maven66	66	50282	MAVEN[129]
MavenEvent66	66	46741	MAVEN[129] and Event2012 [120]
Tweets-data	4	2922	Omdena
Arabic_Dataset_Classifiction	5	7423	Kaggle

5.3 AI-based Models for Event Detection and Classification

This section presents various models proposed in this thesis for event detection and classification, utilizing both supervised and unsupervised learning techniques. Supervised models include the BERT+LSTM model for events classification, which combines deep contextual embeddings with sequential learning for improved accuracy, as well as LSTM-based multi-class classification for handling complex event structures. Additionally, hybrid similarity and BERT-based models integrate similarity measures with deep learning for refined event identification. For unsupervised approaches, clustering methods such as K-means, K-medoids, and Agglomerative Clustering classify events by grouping similar tweets, supported by embedding techniques like TF-IDF and pretrained BERT models. Arabic event classification was enhanced with ARAELECTRA and CAMeLBERT, which addressed linguistic complexities, while DistilBERT was utilized for English tweets to ensure efficient processing and classification.

5.3.1 Supervised Learning Based Models

Our supervised learning-based approaches for event detection and classification rely on labelled datasets to train algorithms for precise predictions. These methods utilize deep learning techniques to analyze textual data, extract meaningful patterns, and accurately categorize events.

5.3.1.1 BERT+LSTM model for events classification

The proposed model leverages a Long Short-Term Memory (LSTM) network in combination with BERT representations to effectively identify and classify events in tweets. For this model, we used our collected dataset TweetsEvent and TweetsEvent310(Refer to section 5.2.2). Extensive experiments were conducted on the dataset to evaluate the model's performance. The results demonstrated an overall accuracy exceeding 94.3% and an F1 score surpassing 90%, demonstrating state-of-the-art performance across the majority of event categories using TweetsEvent.

Methodology

Combining BERT with LSTM improves the classification task, as they are skilled at recognizing patterns, such as the context and structure of words within a sentence. The model takes the input token IDs with a shape of 280 representing the length of the input sequence. Table IV shows the Tokenizer parameters used in the model. Additionally, an attention mask layer is used to specify which tokens should be attended to (1) or ignored (0). The proposed model uses softmax function for the classification task into 50 event types. The model's performance is evaluated using different measurements such Precision, Recall and F1-Score. Figure 5.1 shows the overall architecture of the proposed model.

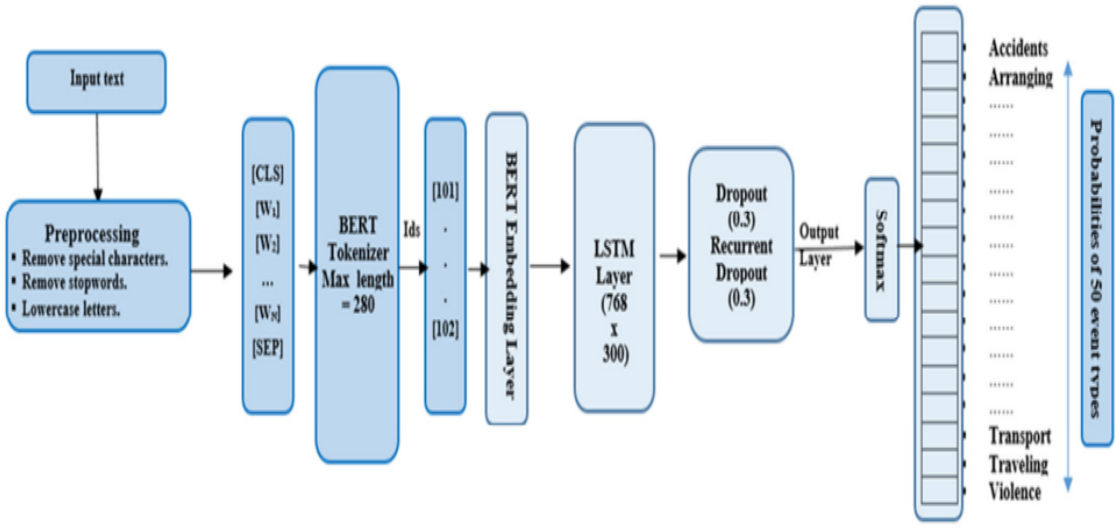


Figure 5.1: The overall architecture of the proposed BERT+LSTM model for events classification.
[6]

Preprocessing

The tweets generated during user conversations often contain various impurities. To enhance data quality for subsequent processing, the dataset underwent the following preprocessing methods:

1. Elimination of Duplicate Tweets: Removing duplicate tweets ensures the cleanliness and accuracy of the data analysis, preventing the distortion of metrics and insights. This step guarantees that each unique tweet makes a meaningful contribution to the analysis or model training.

2. Tweet Cleaning: This process involved removing non-decodable information such as stop words, recurring characters, and hyperlinks from the content, as well as converting all text to lowercase.

3. Standardization of Tweet Length: Tweets that were shorter than 30 characters were removed from the dataset.

Model Parameters

Table 5.4: Tokenizer parameters

Tokenizer	***BertTokenizer 'bert-base-cased'***
Max length	280
Truncation	True
Padding	Max_length
Add special token	True
Return tensors	tf

Table 5.5: Model parameters

Batch	16
Optimizer	Adam
Padding	Max_length
Loss function	CategoricalCrossentropy
Epoch	4
Learning_rate	0.0001
Decay	0.00001

Results, analysis and discussion

We trained our proposed model on two datasets the TweetsEvent and the TweetsEv-

ent310(Refer to section 5.2.1). We obtained the following results.

TweetsEvent dataset: The TweetsEvent dataset has been utilized in research to classify events using deep learning techniques. A proposed model, which combines BERT representations with LSTM layers, demonstrated impressive results. Specifically, it achieved an overall accuracy exceeding 94.3% and an F1 score surpassing 90%, outperforming previous state-of-the-art methods in classifying various event categories. These results demonstrate the impact of using BERT models. In addition, these results show that models that analyze the entire text structure through long-term semantic feature dependencies enhance the performance of event detection and classification tasks. The model was validated using two datasets, both constructed through tweet collection via Algorithm 1. The results of this validation are presented in the corresponding Table 5.6 and through the confusion matrixs for both validation datasets, see the figures (Figure 5.2 and Figure 5.3).

Table 5.6: Performance Metrics for Selected Event Types

Event type	**Precision**	**Recall**	**F1-score**
Validation1.csv: F1 score= 0.927			
Accidents	1.000	1.000	1.000
Arts and Culture	0.920	0.884	0.901
Business and Economy	0.833	1.000	0.909
International relations	0.960	0.888	0.923
Revenge	1.000	0.750	0.857
Validation2.csv: F1_score= 0.926			
Accidents	0.964	0.931	0.947
Arts and Culture	1.000	0.740	0.851
Business and Economy	0.966	1.000	0.983

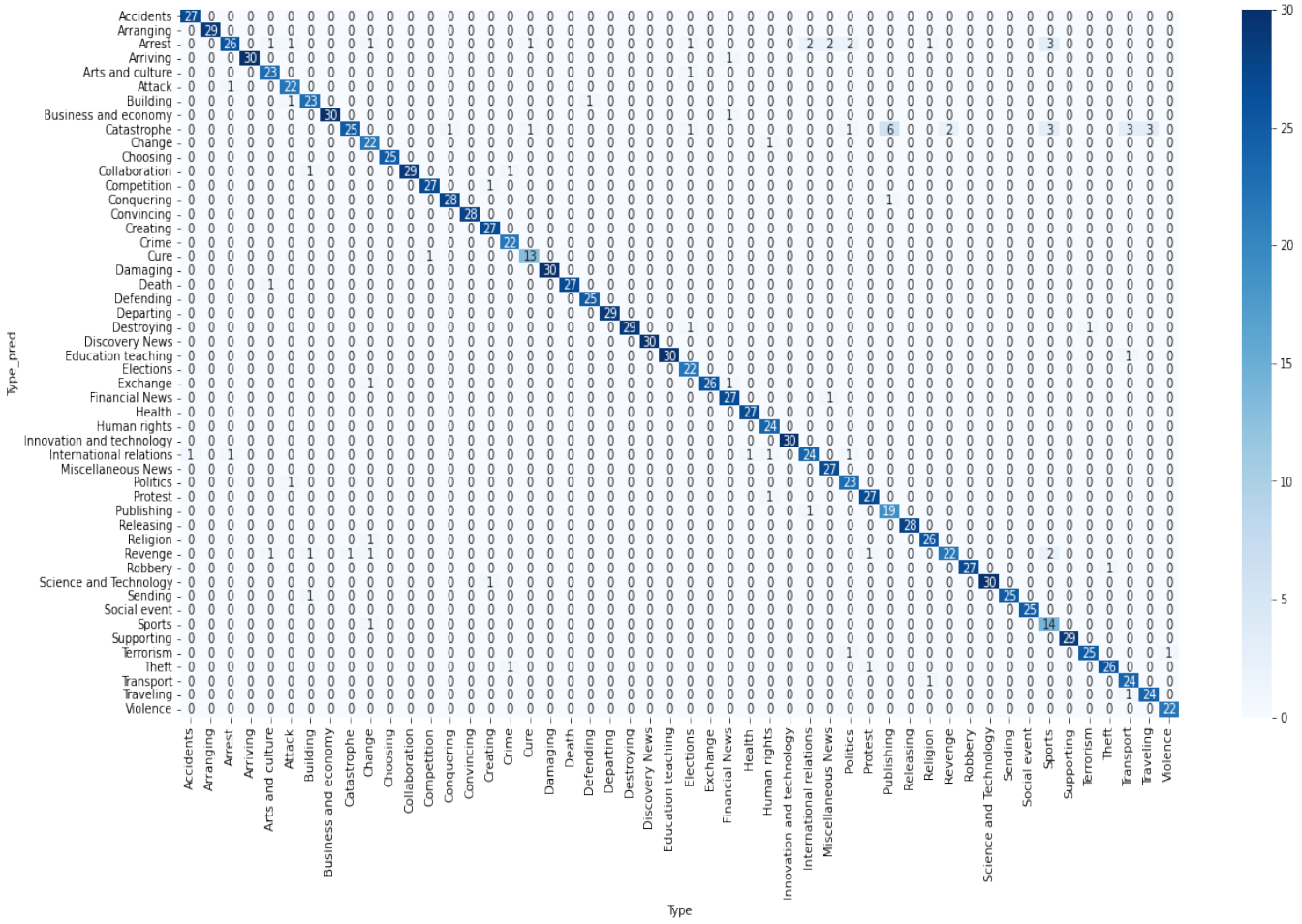


Figure 5.2: Confusion Matrix Analysis for the Validation1 Dataset.

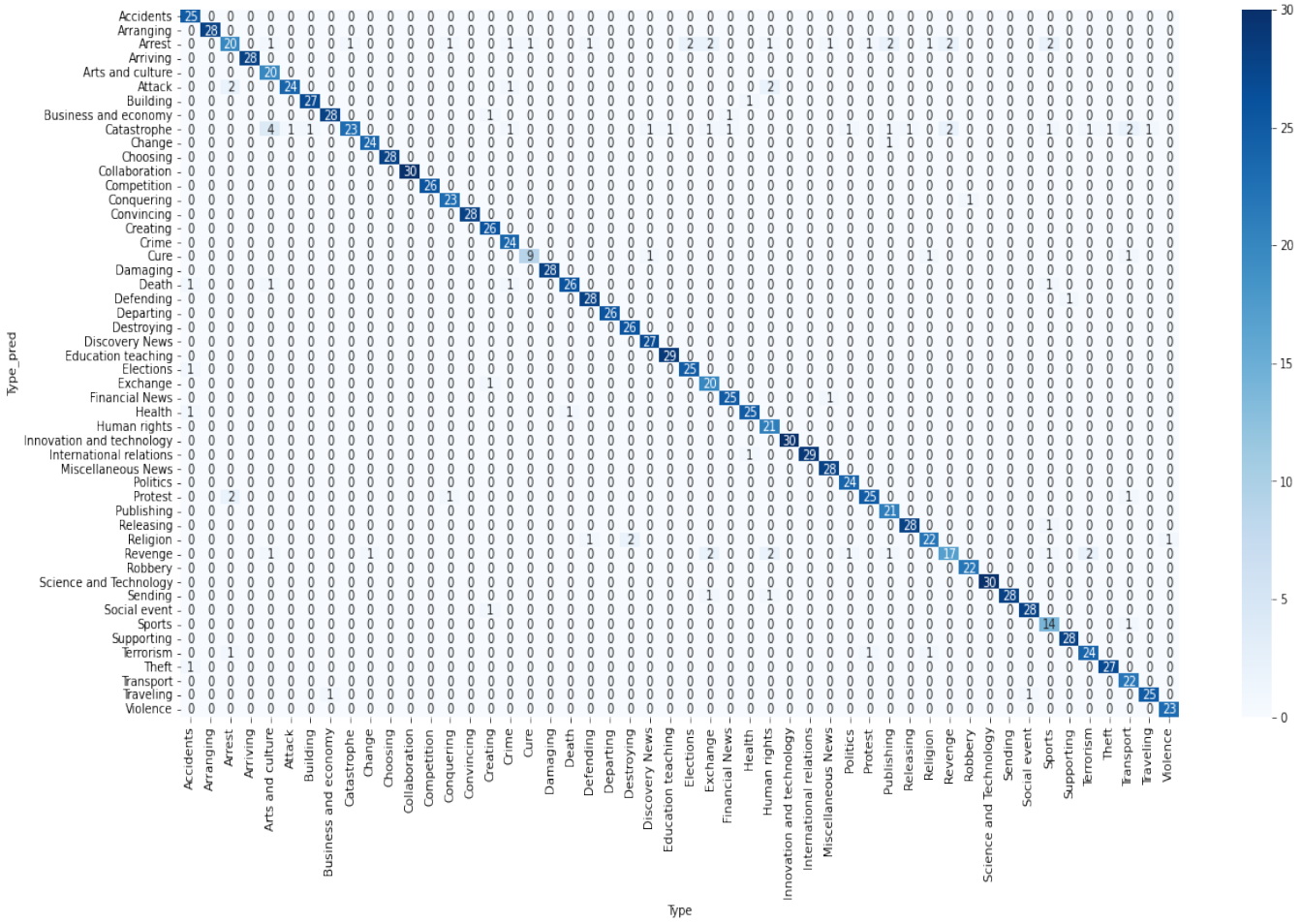


Figure 5.3: Confusion Matrix Analysis for the Validation2 Dataset.

TweetsEvent310 dataset: Table 5.7 presents the performance of the proposed model by type of event. Overall, the experimental results show that the proposed model achieved an overall accuracy greater than **84.85%** and an F1 score over the state-of-the-art results so far **(80%)**, in the classification of most of the event categories.

Table 5.7: Performance Metrics of 50 Event Types

Event type	**Precision**	**Recall**	**F1-score**
Accidents	0.871	0.964	0.915
Arranging	0.800	0.966	0.875
Arrest	0.884	0.821	0.852
Arriving	0.920	0.7667	0.836
Arts and culture	0.8276	0.923	0.873
Attack	0.720	0.720	0.720
Building	0.826	0.731	0.776

Table 5.6 continued

Event type	**Precision**	**Recall**	**F1-score**
Business and economy	0.800	0.933	0.862
Catastrophe	1.000	0.923	0.960
Change	0.690	0.741	0.714
Choosing	0.875	0.840	0.857
Collaboration	0.679	0.655	0.667
Competition	0.833	0.893	0.862
Conquering	1.001	0.966	0.982
Convincing	0.952	0.714	0.816
Creating	0.963	0.897	0.929
Crime	0.556	0.833	0.667
Cure	0.625	0.667	0.645
Damaging	0.829	0.967	0.892
Death	0.885	0.852	0.868
Defending	0.913	0.808	0.857
Departing	0.730	0.931	0.818
Destroying	0.920	0.793	0.852
Discovery News	0.926	0.833	0.877
Education teaching	1.000	0.967	0.983
Elections	0.618	0.808	0.700
Exchange	0.857	0.923	0.889
Financial News	0.967	0.967	0.967
Health	0.931	0.964	0.947
Human rights	0.960	0.889	0.923
Innovation and technology	1.000	0.967	0.983
International relations	0.963	0.963	0.963
Miscellaneous News	1.000	0.833	0.909
Politics	0.786	0.786	0.786
Protest	0.750	0.827	0.787
Publishing	0.952	0.769	0.851
Releasing	0.926	0.893	0.909
Religion	0.697	0.821	0.754
Revenge	0.941	0.667	0.780
Robbery	0.870	0.741	0.800
Science and Technology	0.938	1.000	0.968
Sending	0.846	0.880	0.863
Social event	1.000	1.000	1.000

Table 5.6 continued

Event type	**Precision**	**Recall**	**F1-score**
Sports	0.680	0.773	0.723
Supporting	0.926	0.862	0.893
Terrorism	1.000	0.577	0.732
Theft	0.833	0.741	0.784
Transport	0.676	0.793	0.730
Traveling	0.778	0.778	0.778
Violence	0.913	0.913	0.913

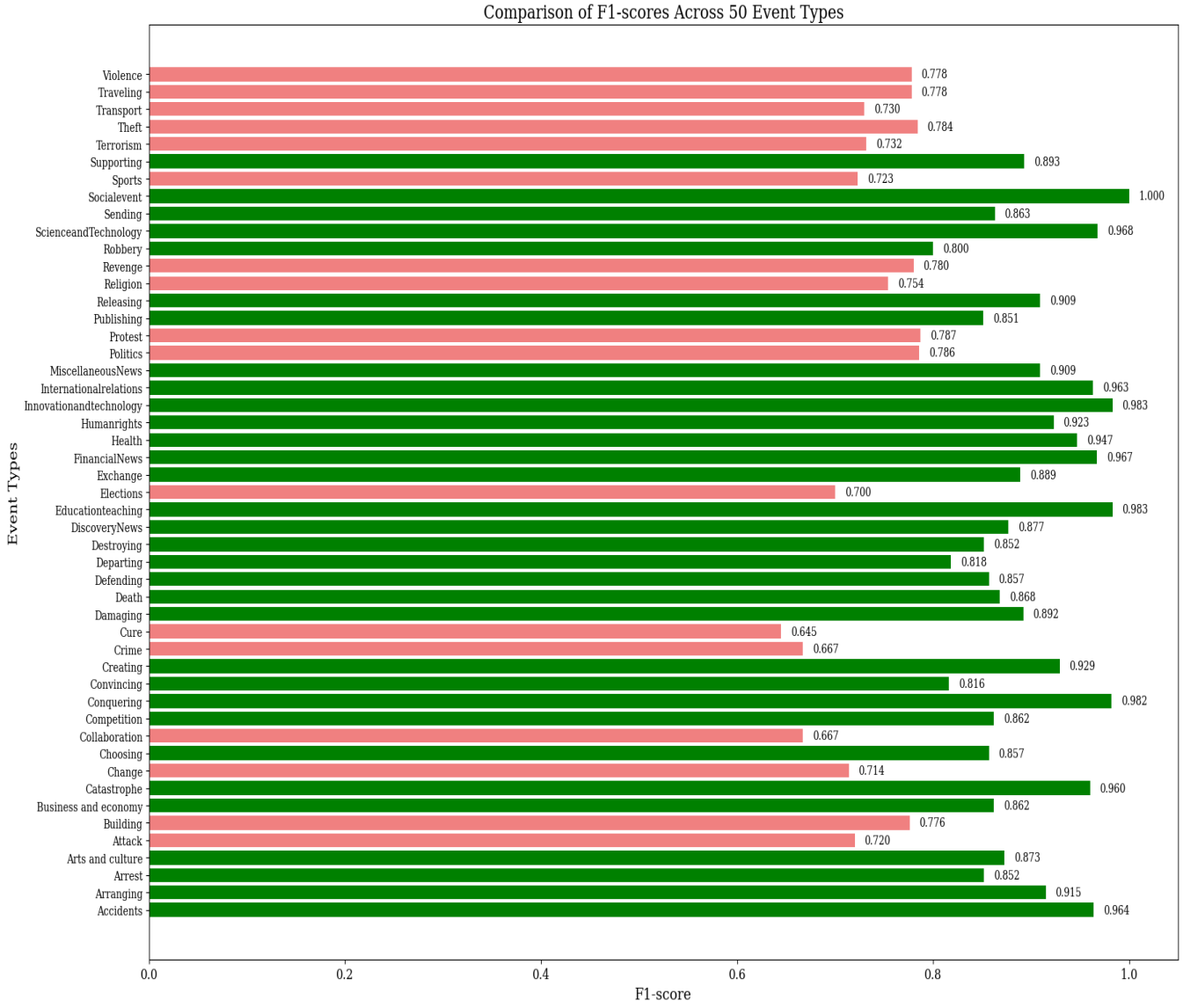


Figure 5.4: Comparison of F1-scores Across 50 Event Types.

- **Best results (F1-score$\geq$ 80%):** A significant majority of events (37 out of 50) achieve an F1-score greater than or equal to 0.80. This suggests that the model performs reliably across a large portion of event types.
- **Results with F1-score<80%:** Only 13 events fall below the threshold. These likely correspond to more ambiguous or challenging categories, such as "Crime," "Terrorism," or "Religion," which might require additional data or better feature engineering.
- **Perfect score:** We obtained a score of 100% Social Events The perfect score suggests a model highly attuned to distinguishing these types of events, likely due to clearly distinguishable data features.
- **High-Performance Event Classification Across 37 Categories:** 37 events demonstrate strong classification performance. This result underscores the advantages of BERT-based models for contextual understanding and the effectiveness of LSTM networks in handling sequential data, highlighting their combined strength in event detection and classification tasks. These events include categories with distinct and well-represented features in the training dataset, such as:
  1. **Education/Teaching (98.3%) :** A high F1-score here may be due to the straightforward and structured nature of educational topics, minimizing ambiguity
  2. **Science and Technology (96.8%) for F1-score:** This category likely benefits from clearly defined patterns, such as specific terminologies or keywords related to advancements and innovations.
- **Events with F1-score < 80%:** 13 events fall below the threshold, indicating areas for potential improvement. Categories such as:
  1. ***Crime (66.7%) and Terrorism (73.%2):*** These may present challenges due to overlapping features with other categories, high variability in data, or insufficient training samples.
  2. **Religion (75.4%):** The lower score here could stem from the diverse and nuanced nature of religious discourse, making classification more complex.

- **Events with Highest F1-Scores ($\geq$ 0.90)**

Table 5.9: Events with Highest F1-Scores ($\geq$ 0.90)

Event	***F1-Score***
Social Event	1.000
Education/Teaching	0.983
Innovation and Technology	0.983
Conquering	0.982
Science and Technology	0.968
Financial News	0.967
Accidents	0.964
International Relations	0.963
Catastrophe	0.960
Health	0.947
Releasing	0.909
Miscellaneous News	0.909

5.3.1.2 LSTM based model for Multi-classifaction

First model

By offering helpful information, real-time occurrences reported in OSM messages issued during catastrophes can aid in disaster management. However, ambiguity, context sensitivity, and event border detection are some of the difficulties that event detection faces. To tackle lengthy sequences issue and efficiently capture temporal dependencies, we consequently suggested an LSTM network. The findings indicate that on a sub-dataset of events produced from 20 keywords TweetsFinal (Refer to Section 5.2.2), the suggested model attains a success accuracy of higher than **71.7%**[7].

Table 5.10: Samples from the TweetsFinal Dataset.

Text	***Keyword***
reminder haiti money donated earthquake 2010	earthquake
ocean rise thunder roar soar storm father king flood	flood
definitely whirlwind emotion	whirlwind

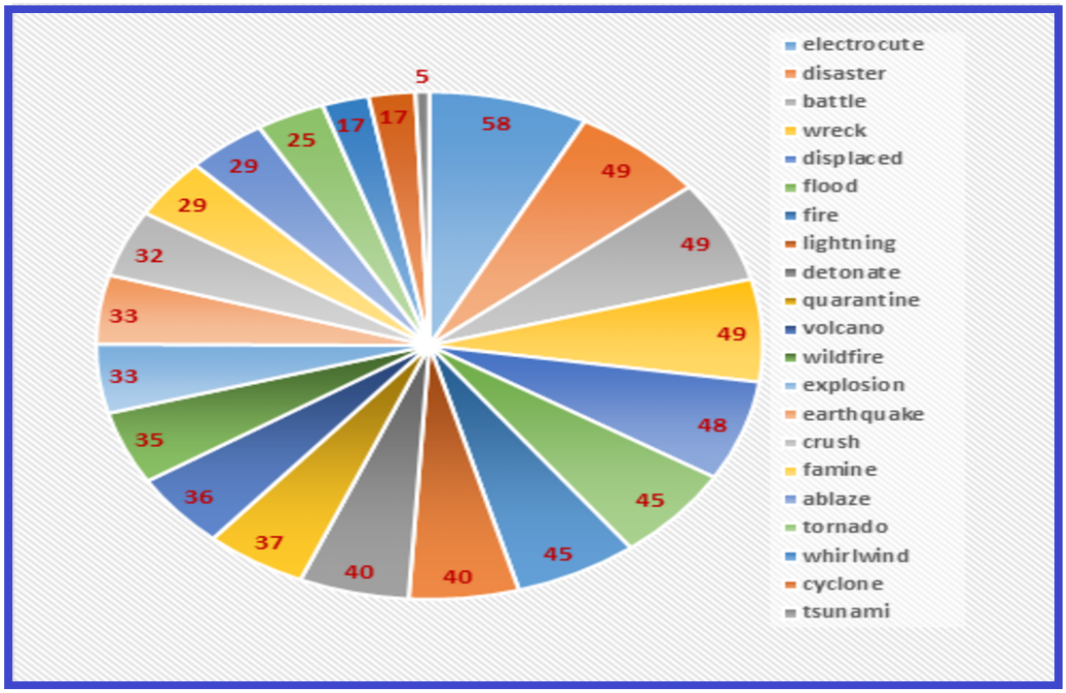


Figure 5.5: Distribution of Keyword Frequencies in TweetsFinal Dataset.

Methodology

We conducted event detection using Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks on the Tweets dataset. A block diagram of the main task is shown in Figure 5.6. In the first block, we performed preprocessing on the text from tweets. Next, we selected 20 event types related to disaster phenomena and split the data into training (size: 916) and validation datasets (size: 80). The model includes a dropout layer to enhance training efficiency and an LSTM layer with 350 units, which utilizes a gating mechanism to manage the memorization process and address the vanishing gradient problem during backpropagation. The activation function transforms the weighted sum of inputs into outputs, significantly influencing the neural network's performance and capability. For the hidden layer, we employed the rectified linear unit (ReLU) activation function, while the softmax activation function was utilized for the output layer. In the final validation block, we evaluated the model using the validation dataset (size: 80). To facilitate event type prediction, we introduced a new field, keywordP. Additionally, we created a confusion matrix and generated a classification report to calculate and analyze performance metrics, including precision, recall, and F1-score.

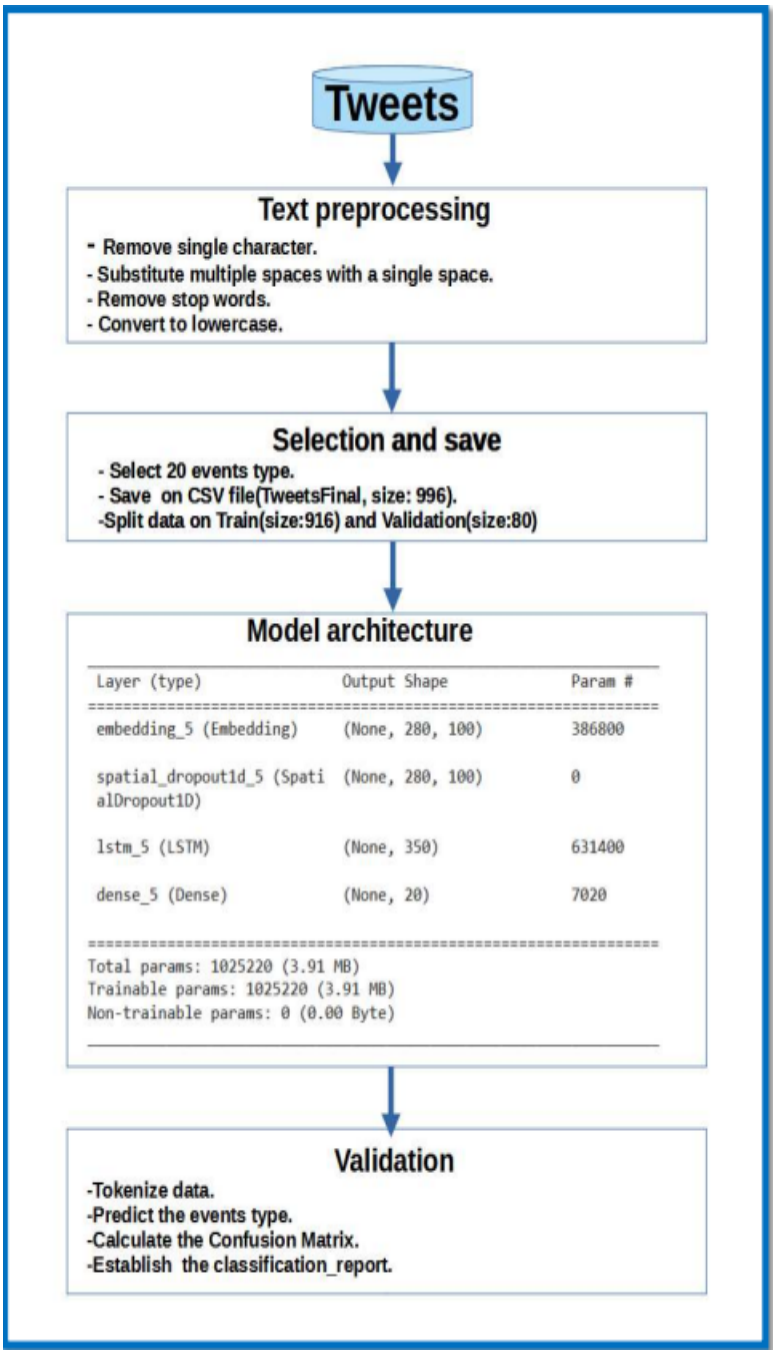


Figure 5.6: Event Detection Framework.

Experiments and results

We conducted a multi-classification experiment utilizing Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks as outlined in the Event Detection Framework (Figure 5.6). The implementation of the proposed model was carried out in Python (version 3.6.5), employing various libraries and packages such as Keras, Numpy, Pandas, Matplotlib, Sklearn, among others, within the Google Colab environment. The model was trained and validated using the Adam optimization algorithm, while the cate-

gorical cross-entropy loss function was utilized to assign probability values to the labels. The dataset, TweetsFinal, was divided into training data (996 rows) and validation data (80 rows). Subsequently, the training data was further split into X_train (824 rows) and X_test (92 rows). The results of the experiment are summarized as follows [7]:

1. Accuracy: The model achieved an accuracy of 71.7% in event detection, which is noteworthy given the dataset's size (996 rows) and the total number of keywords considered (20)..

Table 5.11: Accuracy and loss function

Train Size:916, Keyword:20	
Parameter	***Value***
Train-data-size	824
Test-data-size	92
Loss-value	2.205
Accuracy	71.7%

2. Classification Report: The precision, recall, and F1-score metrics were computed after applying the proposed model to the validation dataset. The results indicate that more than half of the events achieve significant values across all three metrics, demonstrating the model's effectiveness for certain event types. Table 5.11 highlights the performance metrics for selected events.[7]

Table 5.12: Metrics Values for Event Types

Keyword	***Precision***	***Recall***	***F1-score***
cyclone	1.000	0.500	0.666
detonate	0.750	1.000	0.857
earthquake	0.750	1.000	0.857
electrocute	1.000	1.000	1.000
explosion	1.000	0.750	0.857
fire	0.857	1.000	0.923
flood	1.000	1.000	1.000
lightning	1.000	0.600	0.750
tornado	1.000	0.666	0.800
volcano	0.666	1.000	0.800
whirlwind	1.000	1.000	1.000

3. Confusion Matrix: The dataset used to evaluate the proposed model consists of 80 rows. The confusion matrix, created for specific keywords, highlights the significance of diagonal values, which reflect correct classifications. This indicates the model's efficiency in event detection.[7].

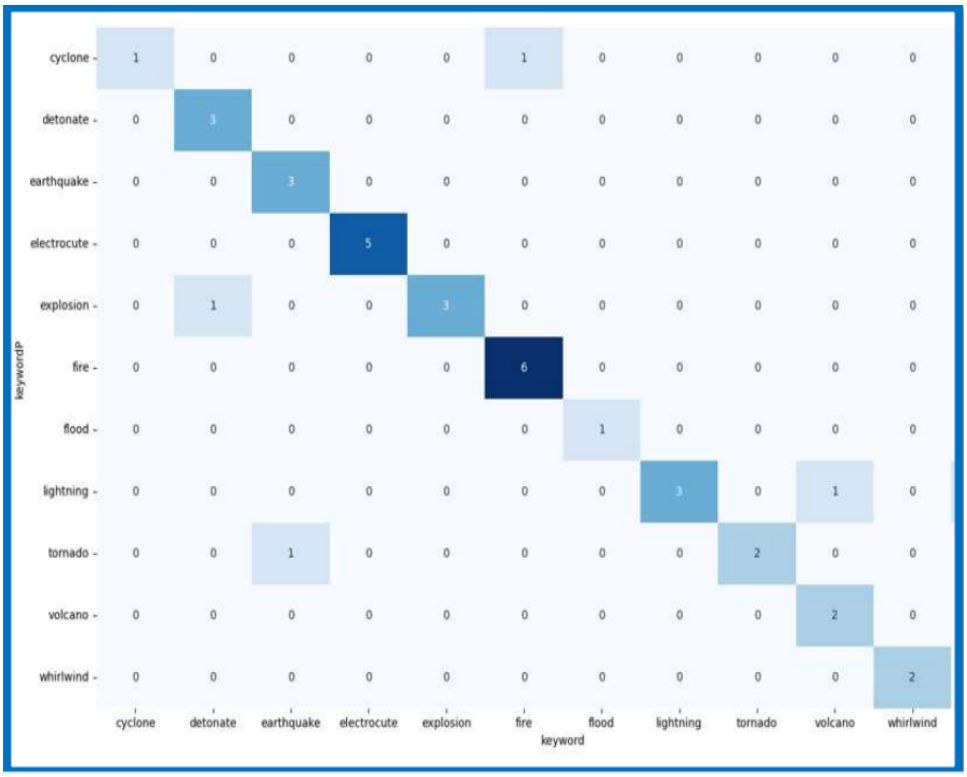


Figure 5.7: Confusion Matrix Analysis of Keyword-Based Classification.

Second model

To tackle the challenge of event detection and classification in short texts, we proposed a semantically deep learning model. These texts encompass diverse contexts, including chat messages, tweets, product descriptions, search queries, and online comments. Notably, this model was presented as a poster at the 2022 5th International Symposium on Informatics and its Applications (ISIA2022) at M'sila University.

- **Maven66** dataset contains 2 fields (Sentence , Event type) the follow table represents some samples:

Table 5.13: Sample Sentences and Event Types: MAVEN66 Dataset

Sentence	***Event type***
There were no direct effects of the earthquake's shaking	Catastrophe
Soon afterwards, the protests spread to many other major cities	Protest
He must also pay $63,000	Commerce_pay

- ***Event35*** dataset contains 2 fields (Description, Event type) the follow table represents some examples:

Table 5.14: Sample Sentences and Event Types: Event35 Dataset

Description	***Event type***
Lenovo takes over HP as the biggest PC maker	Financial News
Ukraine protest fraud on parliamentary election	Politics and Elections
Preparing for Hurricane Sandy	Disasters

- ***MavenEvent66*** Dataset formed by grouping **Maven66** and **Event35**.

Framework Methodology

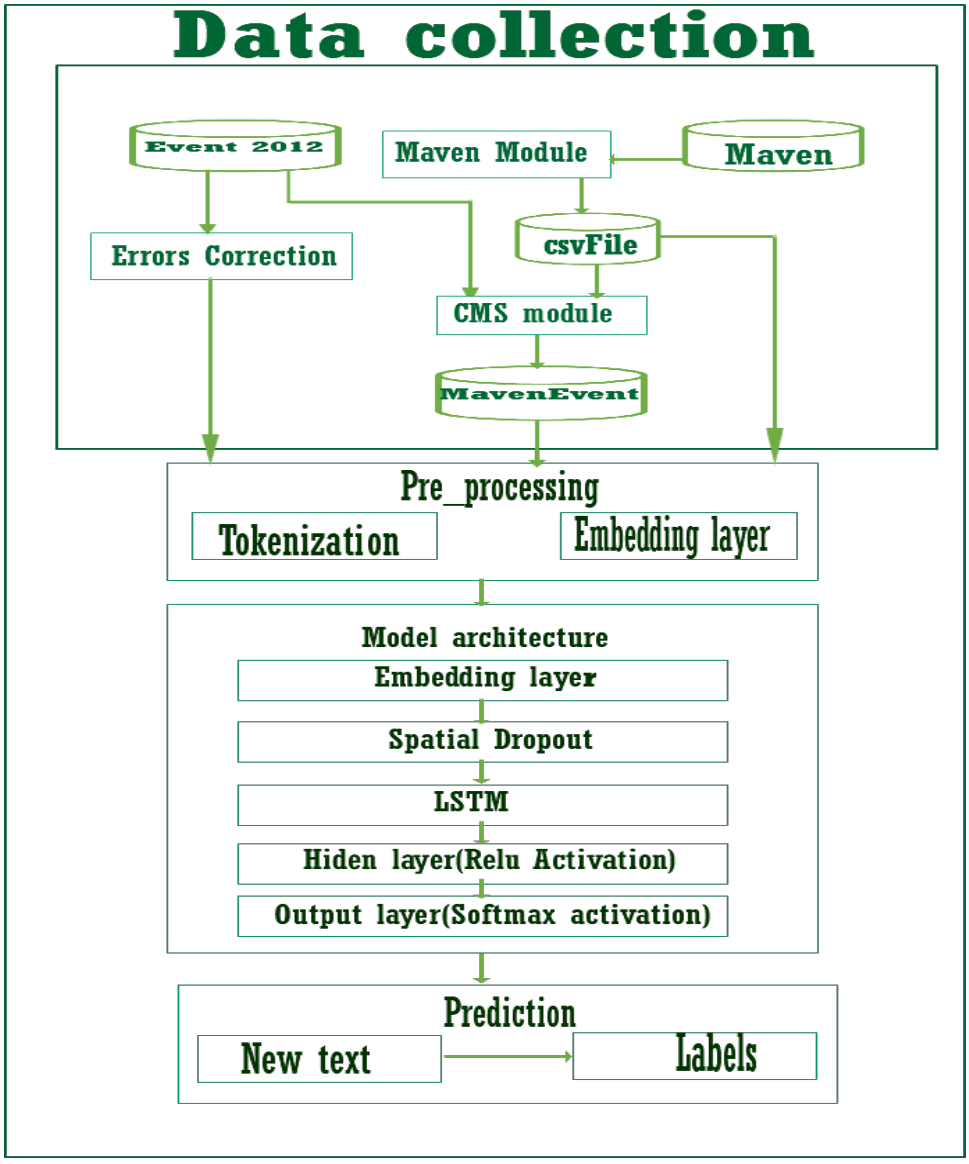


Figure 5.8: Framework for LSTM-Based Multi-Classification Model.

In the Data Collection block, we utilized two datasets: Event2012 and MAVEN. From the MAVEN dataset, we extracted sentences along with their corresponding event types and saved the results into a CSV file for the CMS block (Cleaning, Mapping, and Saving). Within the CMS block, after mapping and grouping the data by event type, we created a refined dataset called MavenEvent66.

We conducted three experiments using the following datasets: Event35 (after error correction), the CSV file containing the Maven66 Dataset, and the MavenEvent66 Dataset. Standard preprocessing steps were applied, including lowercasing and removing stop words. Subsequently, tokenization was performed, and the data was represented using word embeddings before preparing the training and test datasets. The model architecture incorporated a dropout layer, an LSTM layer with 300 units, a ReLU activation function in the hidden layer, and a softmax activation function in the output layer.

In the final Prediction block, we preprocessed the new texts and successfully predicted their corresponding event types.

Experimental Results

We performed classification experiments using Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Networks, obtaining the following results:

a/ First Experiment using Event35 Dataset

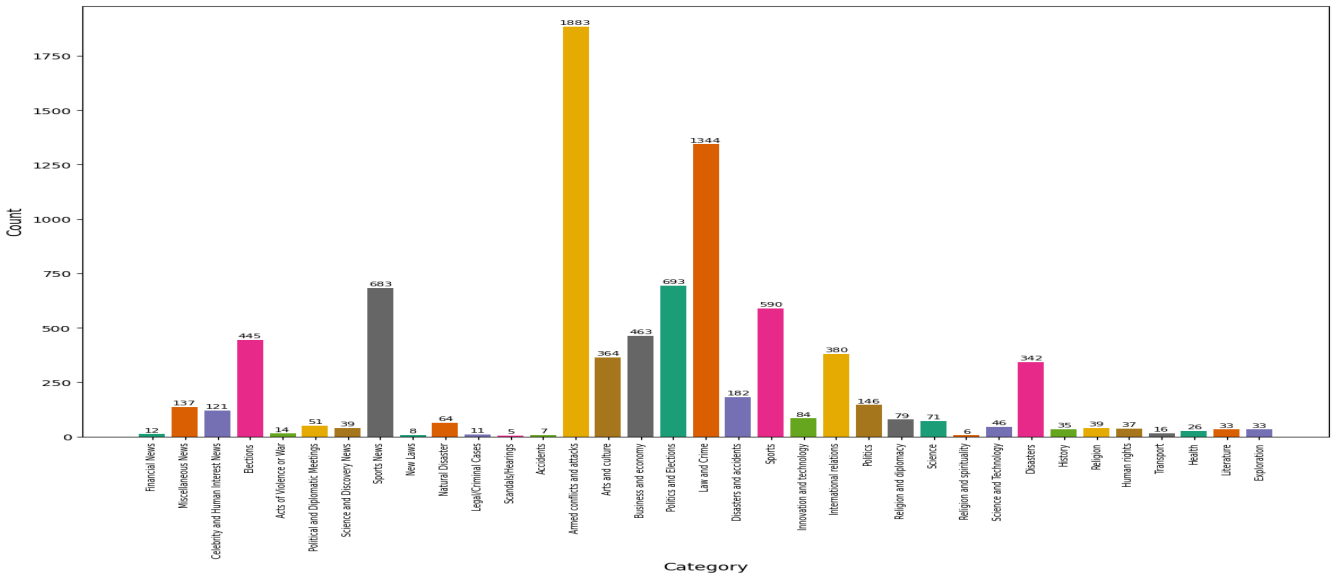


Figure 5.9: Repartition of Events by Category

In this experiment, after correcting errors such as cases where the same category was defined by two different names or empty fields, we obtained a dataset comprising 35 event types.

Table 5.15: Summary of Event35 Dataset and Experimental Results

Dataset Size:8489, Event type:35	
Information	***Value***
Train-data-size	7640
Test-data-size	849
Loss-value	0.523
Accuracy	89.4%

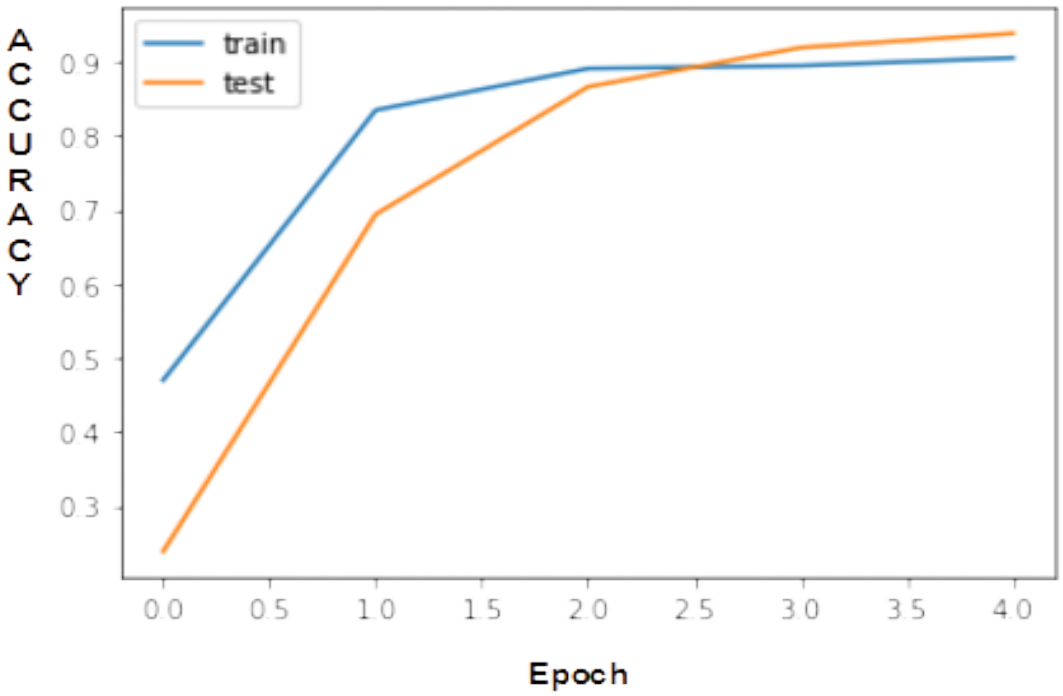


Figure 5.10: Accuracy Evaluation for Event35 Dataset

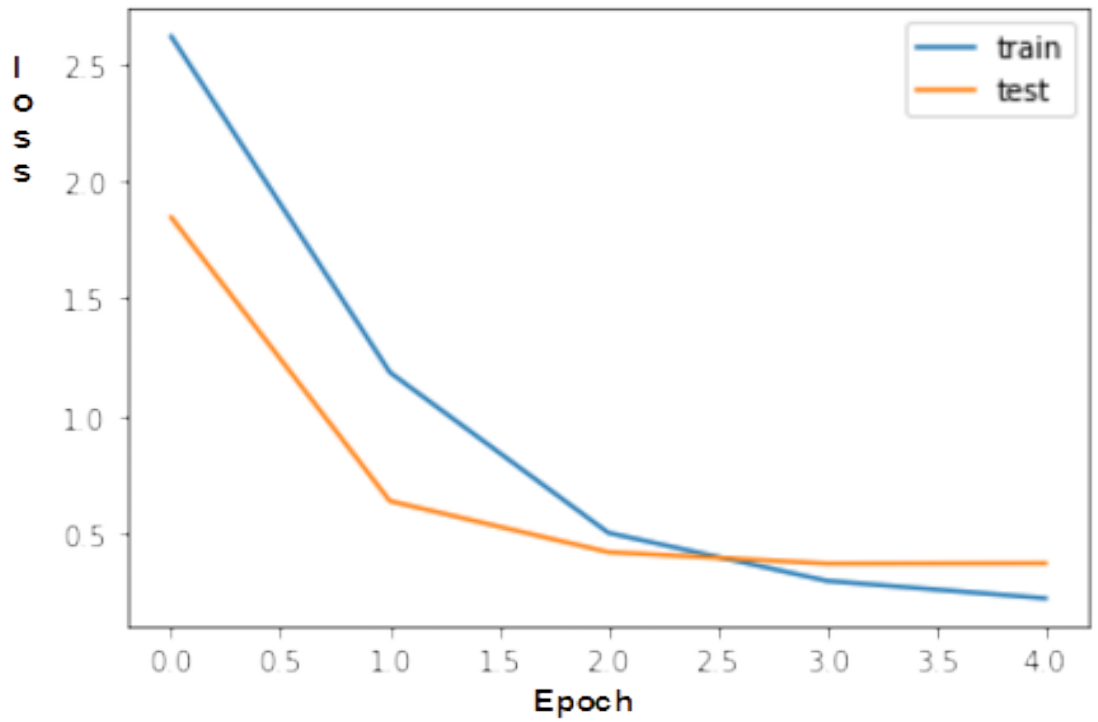


Figure 5.11: Loss Function Analysis for the Event35 Dataset

b/ Second experiment using Maven66 dataset

We selected 66 events from the MAVEN dataset to create the Maven66 dataset. Table 5.15 provides an overview of some of these events along with their frequency of occurrence..

Table 5.16: Example of Event Types and Their Counts in the Maven66 Dataset

Dataset Size:50282, Event type:66	
Event type	***Count***
Theft	198
Legality	270
Violence	293
Robbery	228
Incident	28
Commerce buy	204
Social event	1369
Education teaching	103
Committing crime	216
Traveling	1336

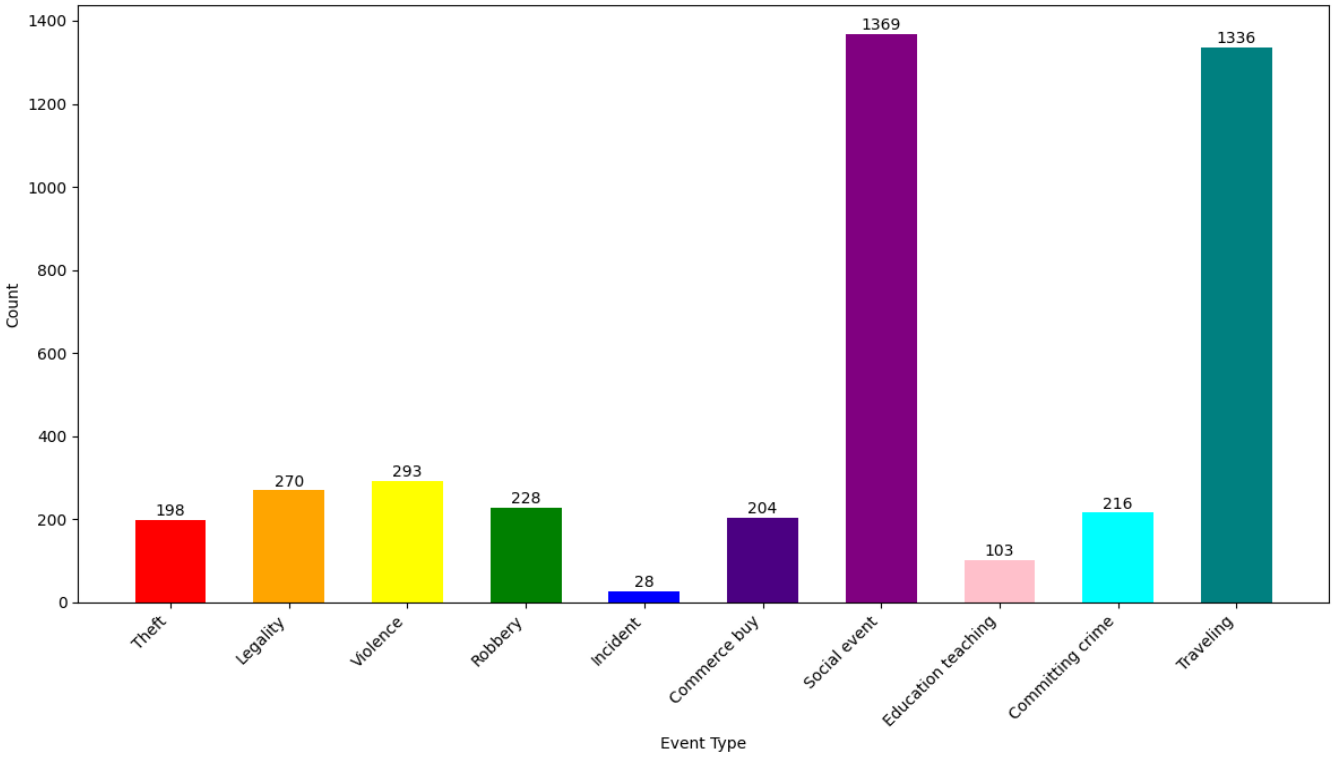


Figure 5.12: Count Distribution of Selected Events in Maven66 Dataset

We performed classification experiments using the same model presented in the Figure 5.8, we obtained follow results:

Table 5.17: Summary of Maven66 Dataset and Experimental Results

Dataset Size:50282, Event type:66	
Information	***Value***
Train-data-size	45253
Test-data-size	5029
Loss-value	1.779
Accuracy	55%

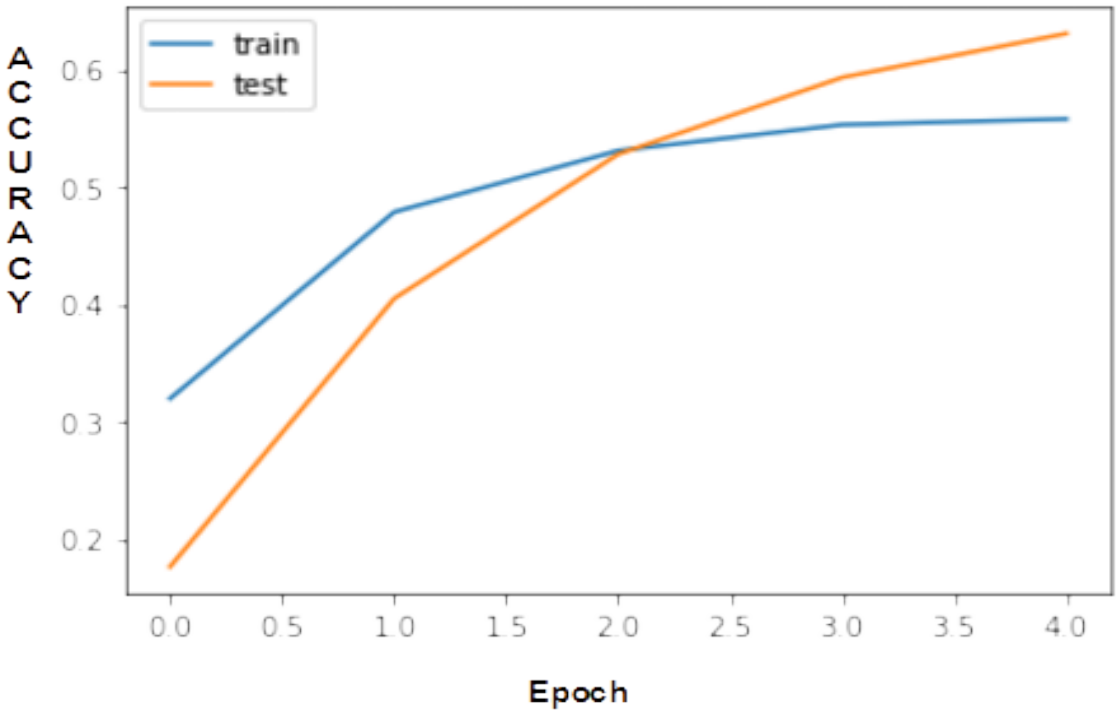


Figure 5.13: Accuracy Function Analysis for the Maven66 Dataset

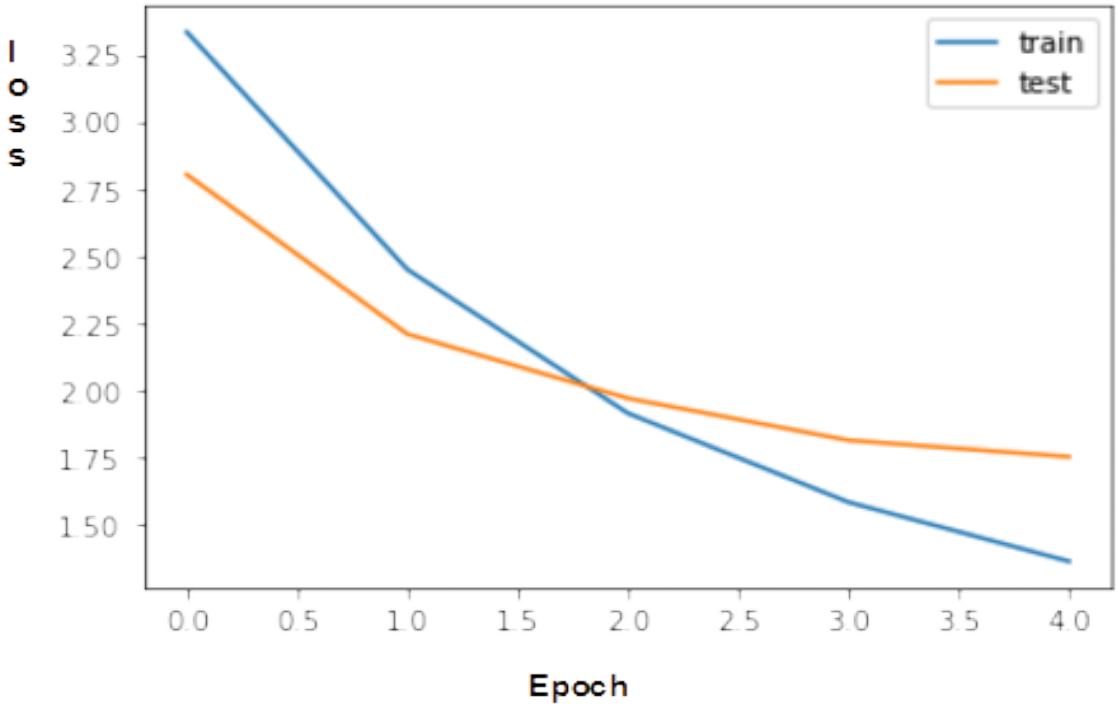


Figure 5.14: Loss Function Analysis for the Maven66 Dataset

c/ Third experiment using MavenEvent66 Dataset

Table 5.18: Example of Event Types and Their Counts in the MavenEvent66 Dataset

Dataset Size:46741, Event type:66	
Event type	***Count***
Theft	198
Legality	270
Social-event	1490
Sports	590
Violence	293
Human rights	37
Politics and Elections	1138
Health	1490
Protest	266
Catastrophe	2236

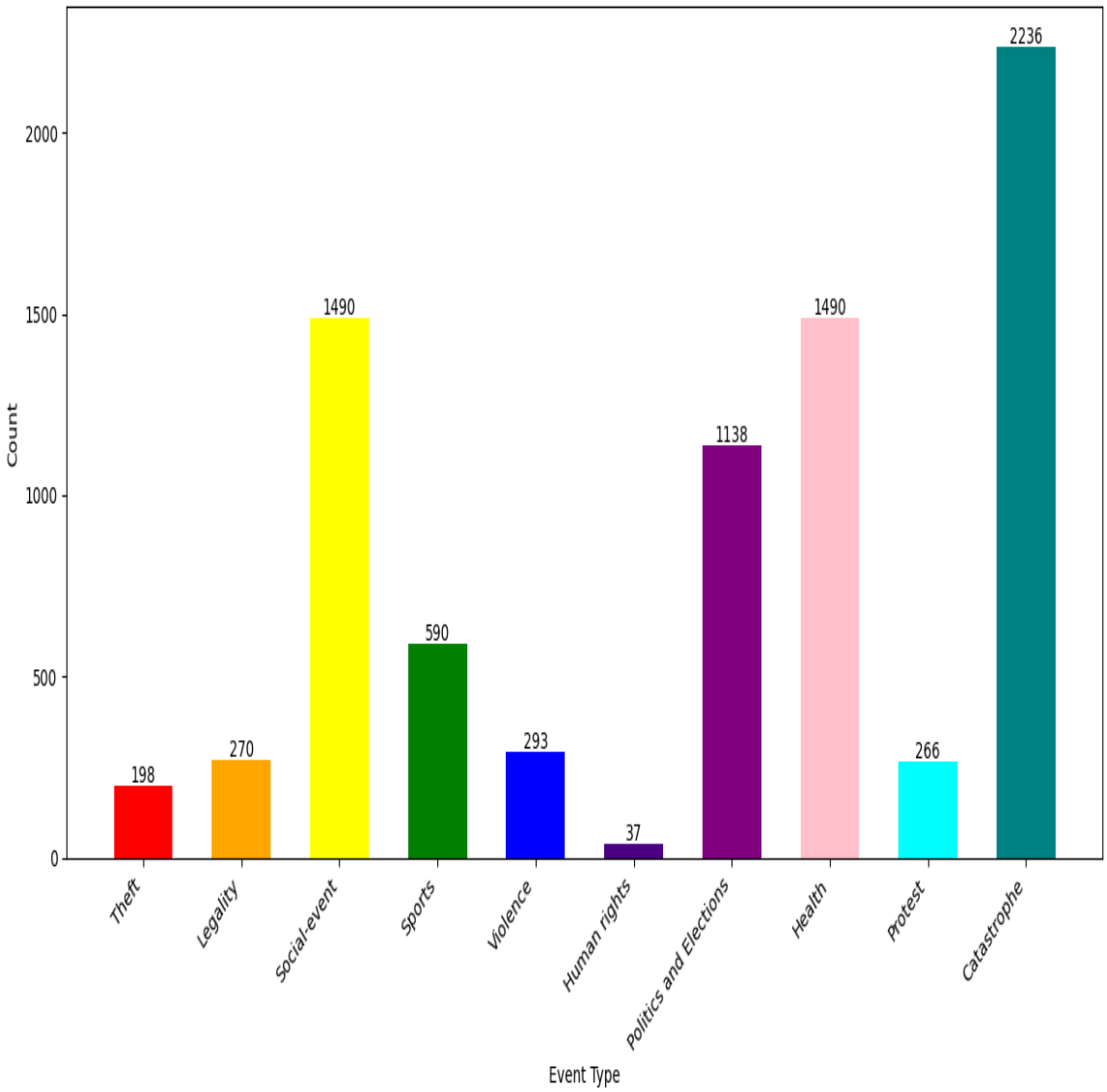


Figure 5.15: Count of Various Event Types in the MavenEvent66 Dataset

In this experiment, our objective was to maintain the same number of event types as in the second experiment (66) while aiming to improve accuracy. To achieve this, we implemented a grouping and mapping strategy between Event35 and Maven66. The following results were obtained as a consequence of this process:

Table 5.19: Summary of MavenEvent66 Dataset and Experimental Results

Dataset Size:46741, Event type:66	
Information	***Value***
Train-data-size	42066
Test-data-size	4675
Loss-value	0.931
Accuracy	**74.4%**

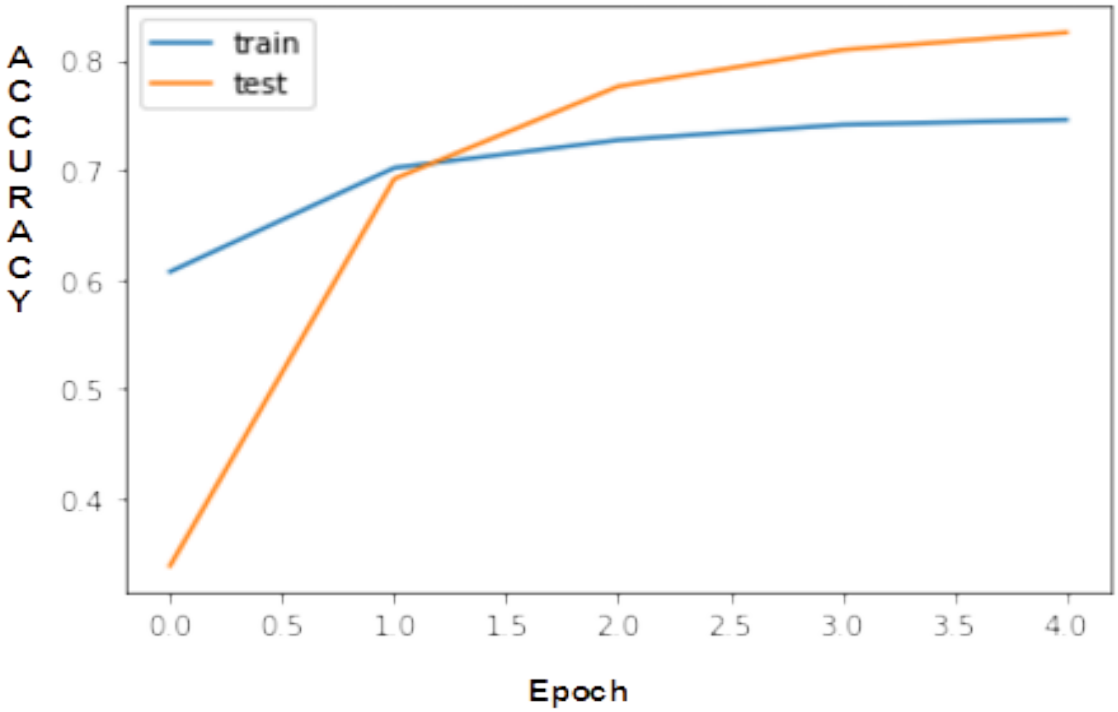


Figure 5.16: Accuracy Function Analysis for the MavenEvent66 Dataset

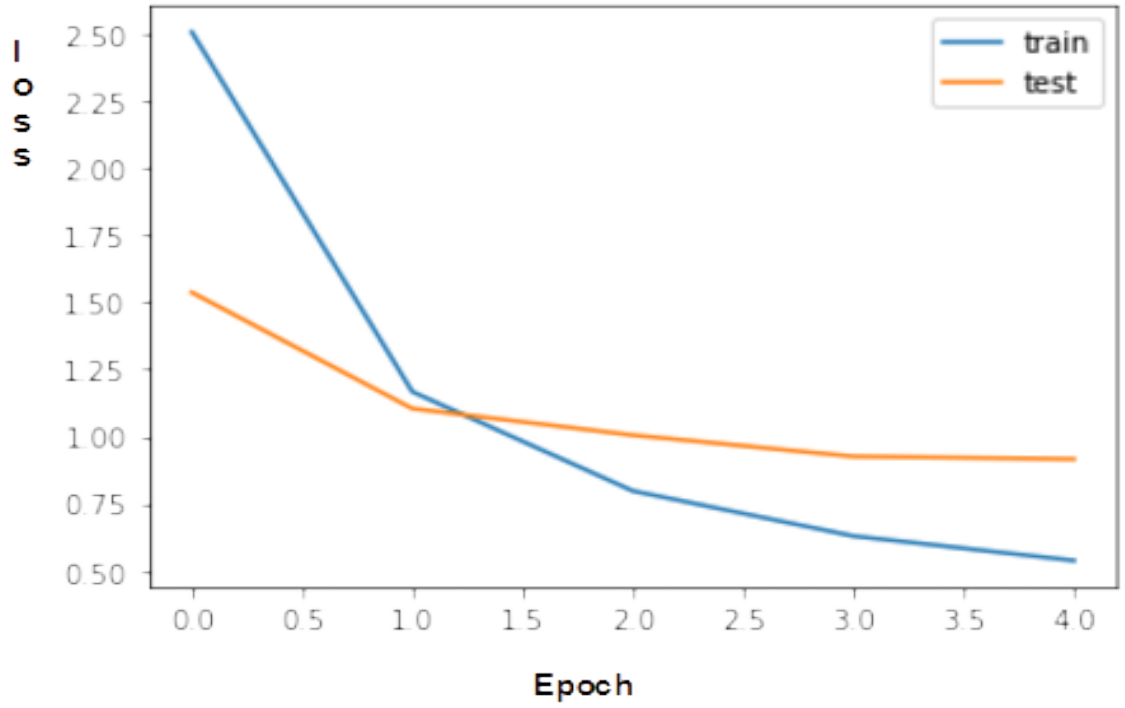


Figure 5.17: Loss Function Analysis for the MavenEvent66 Dataset

4. Analysis

Table 5.20: Comparative Analysis of Experimental Results

Metric	**Event35 Dataset**	**Maven66 Dataset**	**MavenEvent66 Dataset**
Dataset Size	8,489	50,282	46,741
Event Type	35	66	66
Train Size	7,640	45,253	42,066
Test Size	849	5,029	4,675
Loss Value	0.523	1.779	0.931
Accuracy (%)	89.4	55	74.4

The goal of loss and accuracy is to measure how well a model works. Loss sums up the mistakes made by the model, lower loss means fewer mistakes, while higher loss means the model is struggling. In your case, a categorical cross-entropy loss function was used, which is ideal for tasks involving multiple categories. Accuracy, on the other hand, is a percentage showing how often the model's predictions were correct.vThe Event35 dataset achieved the highest accuracy of 89.4% alongside a low loss value of 0.523. However, it includes the smallest number of typical events (35). In contrast, the MavenEvent66 dataset, with 66 event types, like the Maven dataset, exhibited higher accuracy and a lower loss value compared to the Maven66 dataset.

5.3.1.3 Hybrid Similarity and BERT-Based Models for Event Detection

This work proposes a deep learning-based approach for useful event detection, utilizing BERT models for analyzing similarities. Three datasets were introduced: Events35, Maven49, and MavenEvent70, derived from the well-known Event2012 and MAVEN datasets. Events35 consists of 35 selected event types from Event2012, while Maven49 includes 49 event types from MAVEN. To enhance the number of event types and improve performance, Events35 and Maven49 were combined to form MavenEvent70, comprising 70 event types. The study evaluates the performance of models using BERT-based similarity features compared to those incorporating TF.IDF-weighted character n-grams..

Table 5.21: Sample Data for Hybrid Similarity and BERT-Based Event Detection

Dataset	Sentence	type
`Events35`	lenovo takes over hP as the biggest pc maker	Financial News
`Maven49`	there were no direct effects of the earthquake's shaking	Catastrophe
`MavenEvent70`	2012–13 uffa Champions league group stage	Sports

Words Vectors

- **TF-IDF Model:** This model represents text data by assigning importance scores to words based on their frequency in a document and their rarity across the entire dataset.
- **BERT Models:**
  - **Base BERT:** A smaller version of the BERT architecture suitable for general tasks with limited computational resources.
  - **Large BERT:** A more expansive version of BERT that offers improved performance by leveraging a larger number of parameters and layers.

Block Diagrams

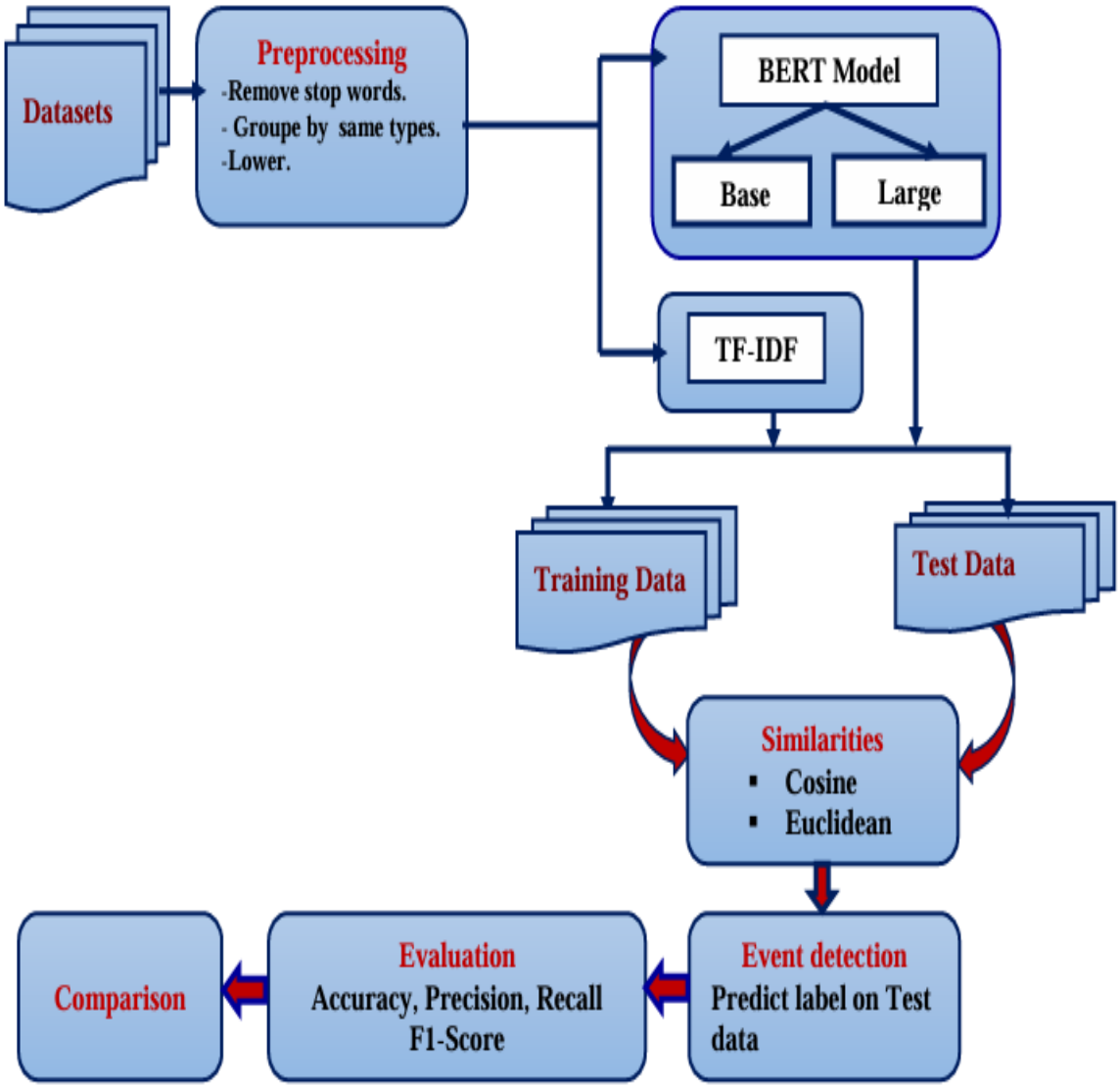


Figure 5.18: Block Diagrams of Event Detection Based on BERT Models, TF-IDF and Similarities.

Algorithm proposed for Useful Event Detection

Algorithm 3 Useful Event Detection with BERT and Similarity

```
// Split data on Train (80%) and Test (20%)
// Encode data with BERT
function BertRepresentation(Train, Test, model)
    // Apply BERT Encoder on column['Sentence']
    return (VecsTrain, VecsTest)
end function
// Predict type event by cosine_similarity
function PredictCosine(VecsTest, VecsTrain, Train, Test)
    // Add empty column to Test data
    Test = typePred[' '], i = 0
    for each vector v in VecsTest do
        a = cosine_similarity(v, Train[0:])
        simMax = max(a)
        if Train['type'][simMax] == Test['type'][i] then
            Test['typePred'][i] = Test['type'][i]
        else
            Test['typePred'][i] = Train['type'][simMax]
        end if
        i = i + 1
    end for
    return Test
end function
// Predict type event by Euclidean_similarity
// Convert distance to similarity
function distanceSimilarity(distance)
    return (1 - distance)
end function
function PredEuclidean(VecsTest, VecsTrain, Train, Test)
    // Add empty column to Test data
    Test = typePred[' '], i = 0
    a = Euclidean(VecsTest, VecsTrain)
    for each vector v in VecsTest do
        s = distanceSimilarity(a[i])
        simMax = max(s)
        if Train['type'][simMax] == Test['type'][i] then
            Test['typePred'][i] = Test['type'][i]
        else
            Test['typePred'][i] = Train['type'][simMax]
        end if
        i = i + 1
    end for
    return Test
end function
```

Experiments and Evaluation

The experiments were conducted using Google Collaboratory, a cloud service that facilitates machine learning research and knowledge sharing through Jupyter notebooks. This platform provides free access to a dependable GPU and an optimized runtime for deep learning tasks. Four experiments were performed using the Events35 and MAVEN49 datasets, applying Cosine and Euclidean similarity measures with both BERT models (Base and Large). For the MavenEvent70 dataset, only Cosine similarity was used with the two BERT models. The algorithm identified type_pred in the x_test data by calculating similarity (either Cosine or Euclidean). To evaluate the performance of useful event detection, the Confusion Matrix and Classification Report (Python) were employed.

- **Data representation:** In our experiments, we began by splitting the data into training (80%) and testing (20%) sets. Subsequently, BERT representations were performed on the data. The time taken and the data shape for each experiment are summarized in the following table.

Table 5.22: Data representation with different models

Dataset	Size	Base Model BERT		Large Model BERT		TF-IDF	
		Cosine sim	Time	Cosine sim	Shape	Time	shape
Events35	8491	16.3 s	(8491, 768)	25.1 s	(8491, 1024)	0.188 s	(8491, 4641)
Maven49	15304	37.9 s	(15304, 768)	2min 12s	(15304, 1024)	1.16 s	(15304, 15731)
MavenEvent70	23795	36.3 s	(23795, 768)	1min 45s	(23795, 1024)	1.06 s	(23795, 17113)

- **Results and analysis**

1.Results

a/ Accuracy

Table 5.23: Accuracy of different models

Dataset	Base Model BERT		Large Model BERT		TF-IDF	
	Cosine sim	Euclidean sim	Cosine sim	Euclidean sim	Cosine sim	Euclidean sim
Events35	87.8%	88%	89.3%	88.8%	48.2%	48.2%
Maven49	72.5%	72.6%	72.4%	71.8%	64.2%	64.2%
MavenEvent70	79.2%	79.3%	79.5%	79.7%	42.1%	42.1%

b/ Classification Report: The Classification Report provides an evaluation of model performance by presenting key metrics(Precision, Recall, and F1-score)for each class in a multiclass classification problem.

Table 5.24: Metric Values with Cosine Similarity Across Different Event Models

Event type	**Base Model BERT**			**Large Model BERT**			**TF-IDF**		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Events35									
Arts and culture	0.961	0.833	0.892	1.000	0.866	0.928	0.300	0.500	0.375
Law and Crime	0.900	0.906	0.903	0.916	0.959	0.937	0.720	0.635	0.675
Politics and Elections	0.901	0.888	0.895	0.915	0.902	0.909	0.135	0.166	0.149
Sports	0.981	0.885	0.931	0.980	0.836	0.902	0.846	0.241	0.376
Maven49									
Death	0.688	0.794	0.738	0.697	0.769	0.731	0.792	0.656	0.718
Exchange	0.444	0.444	0.444	0.571	0.444	0.500	0.953	0.706	0.811
Revenge	0.750	0.600	0.666	0.600	0.600	0.600	0.972	0.734	0.837
Social event	0.547	0.621	0.582	0.704	0.775	0.738	0.912	0.567	0.699
MavenEvent70									
Arts and culture	0.904	0.974	0.938	1.000	0.974	0.987	0.689	0.188	0.296
Attack	0.905	0.875	0.890	0.921	0.879	0.900	0.602	0.739	0.664
Politics and Elections	0.930	0.930	0.930	0.920	0.904	0.912	0.338	0.209	0.258

Table 5.25: Events Metrics Values with Euclidean Similarity across different models

Event type	**Base Model BERT**			**Large Model BERT**			**TF-IDF**		
	Precision	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Events35									
Arts and culture	1.000	0.833	0.909	1.000	0.866	0.928	0.300	0.500	0.375
Law and Crime	0.878	0.926	0.901	0.921	0.946	0.933	0.720	0.635	0.675
Politics and Elections	0.901	0.888	0.895	0.888	0.888	0.888	0.135	0.166	0.149
Sports	0.981	0.885	0.931	0.981	0.852	0.912	0.846	0.241	0.376
Maven49									
Death	0.710	0.692	0.701	0.642	0.692	0.666	0.792	0.656	0.718
Exchange	0.444	0.444	0.444	0.625	0.555	0.588	0.953	0.706	0.811
Revenge	0.800	0.666	0.727	0.615	0.666	0.615	0.972	0.734	0.837
Social event	0.475	0.513	0.493	0.487	0.540	0.487	0.912	0.567	0.699
MavenEvent70									
Arts and culture	0.926	0.974	0.950	1.000	0.974	0.987	0.689	0.188	0.296
Attack	0.901	0.871	0.886	0.918	0.887	0.902	0.602	0.739	0.664
Politics and Elections	0.929	0.921	0.925	0.912	0.904	0.908	0.937	0.954	0.945
Sports	0.909	0.918	0.914	0.937	0.954	0.945	0.664	0.689	0.677

2. Results analysis and discussion: This analysis focuses on event detection and classification, comparing the effectiveness of different text representation models and similarity measures in these tasks. Specifically, it evaluates the performance of BERT (base and large variations) and TF-IDF approaches using cosine and Euclidean similarity measures across three datasets (see Figure 5.18). The Figure 5.19 and the Figure 5.20 extend the exploration by presenting the accuracy of different models alongside the number of events processed for each dataset using the proposed algorithm (Algorithm 3 Useful Event Detection with BERT and Similarity). The evaluation of the suggested method was conducted on a personal computer equipped with an Intel(R) Core(TM) i5-6300U CPU operating at speeds of 2.40 GHz and 2.50 GHz, with 16.0GB of installed RAM. The system runs Microsoft Windows 10 Professional as the operating system, and Python was employed as the programming language for implementation.

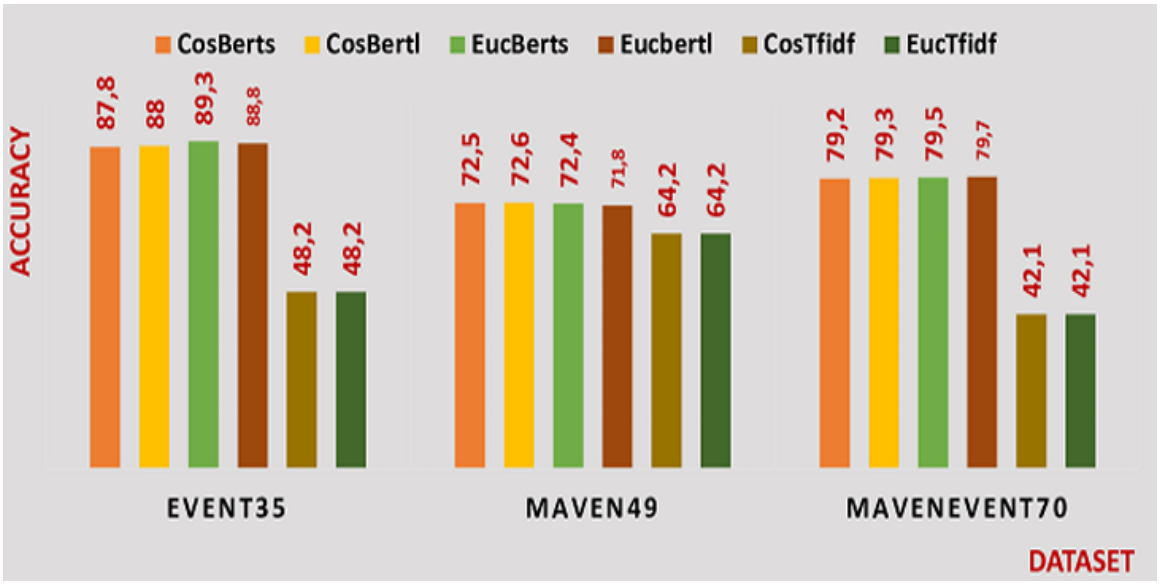


Figure 5.19: Comparative Performance: BERT, TF-IDF, and Similarity Measures Across Datasets.

Where:

- **CosBerts:** Cosine similarity using the base BERT model.
- **CosBertl:** Cosine similarity using the large BERT model.
- **EucBerts:** Euclidean similarity using the base BERT model.
- **Eucbertl:** Euclidean similarity using the large BERT model.
- **CosTfidf:** Cosine similarity with the TF-IDF model.
- **EucTfidf:** Euclidean similarity with the TF-IDF model.

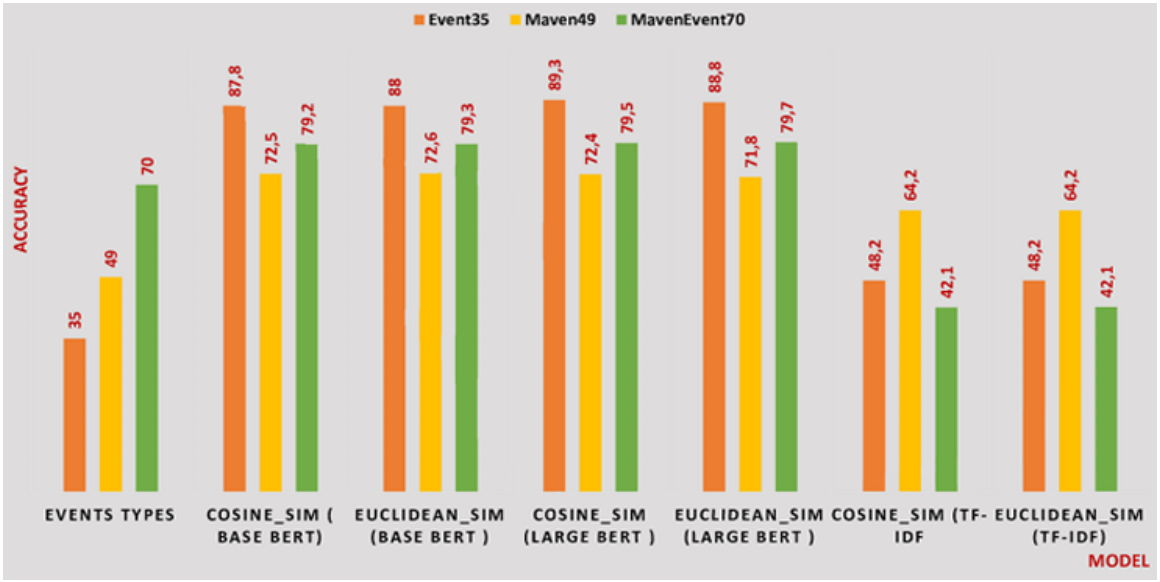


Figure 5.20: Model Accuracy and Event Distribution Among Datasets.

Data representation: Table 5.21 highlights the differences in data representation times and dimensions among BaseModel BERT, LargeModel BERT, and TF-IDF. The base BERT model demonstrates efficiency, with representation times ranging from 9 seconds for smaller datasets like "Events35" to 3 minutes for larger datasets. The large BERT model, while requiring 1 to 10 minutes for representation, provides higher-dimensional outputs (1024 dimensions). On the other hand, TF-IDF processes datasets in under 2 seconds, though with significantly larger output dimensions.

This analysis suggests that the base BERT model strikes an optimal balance between speed and representation quality, making it suitable for tasks requiring both efficiency and robust feature extraction. The large model offers richer representations but at a higher computational cost, whereas TF-IDF excels in speed, ideal for lightweight tasks.

Event types Metrics Values with different Similarities across different models: The Table 5.22 presents the use of two text similarity measurements, Cosine and Euclidean, to evaluate model accuracy across datasets. The "Events35" dataset achieved the highest accuracy (90.5%) using Cosine similarity with the large model, while the base model yielded the highest accuracy (88.5%) with Euclidean similarity. For the "MAVEN49" dataset, the large model recorded the highest accuracy (68.1%) with Euclidean similarity, whereas the base model achieved slightly lower accuracy (67.9%) with Cosine similarity. The "MavenEvent70" dataset showed its peak accuracy (71.8%) using Cosine similarity in the large

model. These results underscore the varying effectiveness of similarity measures and models depending on the dataset's characteristics.

As previously mentioned, the objective of our work is to expand the number of event types. The MavenEvent70 dataset, which encompasses the largest variety of event types (70), achieves an acceptable accuracy of 71.8%. Referring to Table 5.22, we can conclude the following: For the Events35 and Maven49 datasets, the base BERT model is more efficient and faster in processing. However, for the MavenEvent70 dataset, the large BERT model is preferred as it enhances the precision of event detection, making it better suited for datasets with a broader range of event types.

Evaluating Event Types Using Precision, Recall, and F1 Scores: The Table 5.23 compares the performance of Base Model BERT, Large Model BERT, and TF-IDF using precision, recall, and F1 score across different event types. Large Model BERT consistently achieves higher F1 scores across most categories, such as "Arts and Culture," "Law and Crime," and "Politics and Elections," showcasing its effectiveness. Base Model BERT performs well, though slightly behind the Large Model. TF-IDF struggles significantly in comparison, particularly in "Politics and Elections" where its F1 score drops to 0.149.

The Table 5.24 shows similar metrics (precision, recall, and F1 score) for the same models but with Euclidean similarity. Large Model BERT once again outshines the others in most categories, emphasizing its superior performance. Base Model BERT shows strong results but slightly lags in certain cases. TF-IDF, while performing moderately in some categories like "Exchange," remains less effective overall, with consistently lower F1 scores.

In conclusion we observed the dominance of BERT-based models, particularly the Large Model, in achieving high precision, recall, and F1 scores, making them better suited for tasks requiring nuanced understanding of events.

5.3.2 Unsupervised based Models

In this section, we emphasize the application of clustering algorithms, K-means, K-medoids, and Agglomerative Clustering, to classify events in tweets. It focuses on comparing the performance of these methods in clustering both Arabic and English tweets. Two embedding approaches were employed: TF-IDF and pretrained BERT models. For

Arabic tweets, advanced models like ARAELECTRA and CAMeLBERT were leveraged to address linguistic complexities, while DistilBERT was utilized for English tweets.

Methodology

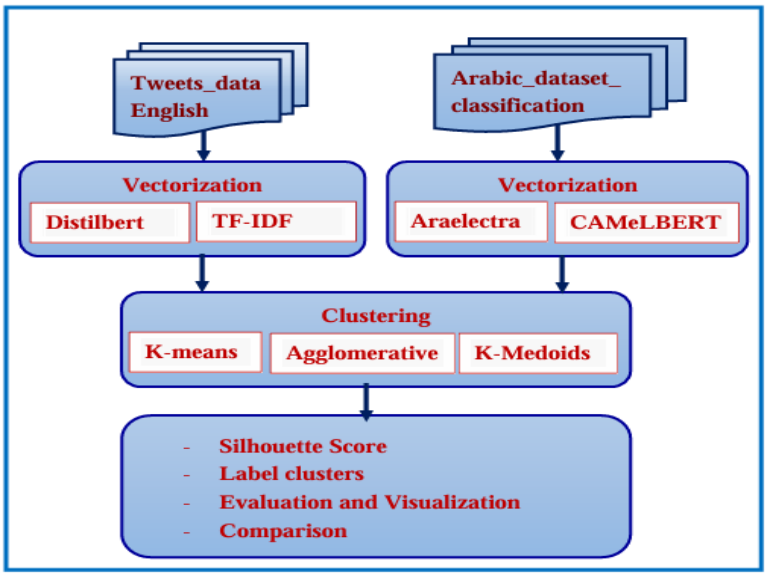


Figure 5.21: Clustering Framework.

The proposed approach involves embedding textual datasets using advanced models, followed by clustering and labeling processes tailored for distinct datasets.

Step 1: Vectorization

Textual data is transformed into numerical representations using embedding techniques:

- **TF-IDF**: A classical approach for text representation, emphasizing term relevance within documents.
- **DistilBERT**: Efficient contextual embeddings for English text, capturing semantic nuances.
- **ARAELECTRA**: Designed for Arabic, leveraging pre-training for efficient representation in morphologically rich languages.
- **CAMeLBERT**: Tailored for Arabic text, capturing standard and dialectal nuances.

Step 2: Clustering

The clustering process involves grouping data points based on their embeddings:

1. **Algorithms Applied:**
    - **K-Means**: Minimizes intra-cluster variance for well-defined clusters.
    - **K-Medoids**: More robust to noise and outliers, as cluster centers are actual data points.
    - **Agglomerative Clustering**: Hierarchical clustering based on linkage criteria.
2. **Number of Clusters:**
    - For the English tweets dataset: $k = 4$.
    - For the Arabic dataset: $k = 5$.
3. **Evaluation Metric:**
    - The **Silhouette Score** is computed to evaluate clustering quality, ensuring well-defined and interpretable clusters.

Step 3: Labeling Clusters

Distinct approaches are applied to label the clusters:

- **English Dataset:** Algorithm 4 derives descriptive labels by analyzing frequent terms or latent topics.
- **Arabic Dataset:** Algorithm 5 incorporates linguistic features unique to Arabic, enhancing cultural and semantic accuracy.

5.3.2.1 Clustering and labeling Algorithm for English Dataset

This algorithm uses Python's itertools package to determine all possible permutations of disaster labels without repetition. For the dataset "Tweets-data," which includes labels such as 'collapse,' 'cyclone,' 'meteor,' and 'meteorite,' the total number of permutations is calculated as ($n! = 4 \times 3 \times 2 \times 1 = 24$.) Each permutation is assigned to clusters [0, 1, 2, 3], and metric values are computed for each permutation using a classification report. Finally, the permutation with the highest accuracy is selected, ensuring optimal clustering performance. This method effectively explores different permutations and identifies the one yielding the best results.

Algorithm 4 Label cluster for English dataset

Require: Input: Label list of disaster types
Ensure: Output: Maximum result from classification report
1: Import itertools
2: Define label list: ['collapse', 'cyclone', 'meteor', 'meteorite']
3: Generate all permutations of the label list
4: Initialize cluster-df['type-pr'] to an empty string
5: **for** each permutation in permutations **do**
6: Assign perm[0] to cluster-df['type-pr'] for cluster 0
7: Assign perm[1] to cluster-df['type-pr'] for cluster 1
8: Assign perm[2] to cluster-df['type-pr'] for cluster 2
9: Assign perm[3] to cluster-df['type-pr'] for cluster 3
10: Print perm[0], perm[1], perm[2], perm[3]
11: Create Classification Report for cluster-df['type-of-disaster'] vs. cluster-df['type-pr']
12: Convert the report to a DataFrame and store it in R
13: **end for**
14: Find the maximum result from R and store it in Result

5.3.2.2 Clustering and labeling Algorithm for Arabic Dataset

The algorithm provided outlines an effective method for assigning labels to textual data based on the presence of predefined keywords. Given a set of labels, each associated with a list of keywords, and a target text to be classified, the algorithm calculates the level of association between the text and each label by counting keyword matches. Initially, a data structure is prepared to store the count of matches for each label, ensuring it is initialized to zero for all labels. The algorithm then iterates through each label and its corresponding keywords, incrementally tallying the occurrences of each keyword within the given text. This process results in an updated count of matches for every label. Subsequently, the label with the highest match count is identified as the best-fitting label for the text. In cases where no keywords are matched across all labels, the algorithm assigns a default label, "Divers," to indicate the absence of any strong association. This algorithm is particularly useful in text classification tasks, enabling efficient labeling of data such as documents, feedback, or articles based on predefined categorical criteria. Its structured approach ensures clarity, adaptability, and effectiveness in organizing unstructured textual content.

Algorithm 5 Algorithm for Labelling Arabic Text Based on Keywords

Inputs:
Labels: $\{C_1, C_2, \ldots, C_n\}$, where each label C_i contains a list of keywords $\{K_{i1}, K_{i2}, \ldots, K_{im}\}$
Text: T
Outputs:
Best Label C_b (or *Divers* if no matches are found)
Initialize: MatchCount$(C_i) \leftarrow 0$ for all $i = 1, 2, \ldots, n$
for $i = 1$ to n **do**
    $count \leftarrow 0$
    for each keyword K_{ij} in C_i **do**
        $count \leftarrow count +$ Number of occurrences of K_{ij} in T
    end for
    MatchCount$(C_i) \leftarrow count$
end for
$C_b \leftarrow \arg\max_i$ MatchCount(C_i)
if MatchCount$(C_b) = 0$ **then**
    $C_b \leftarrow$ Divers
end if
return C_b

5.3.2.3 Experiments and evaluations

Several experiments were conducted using a range of machine learning models with both the Tweets-data and Arabic Dataset Classification datasets. Each model was applied to evaluate and optimize the classification tasks specific to the datasets. For the Tweets-data, models were tested to accurately classify disaster types (e.g., 'collapse,' 'cyclone,' 'meteor,' and 'meteorite') based on tweet content. On the other hand, for the classification on Arabic dataset, models were evaluated on their ability to effectively classify Arabic text into categories such as sports, politics, culture, economy, and divers.

5.3.2.4 Clustering algorithms for English tweets

The Figure 5.22 shows value distributions for the English dataset ***Tweets-data***.

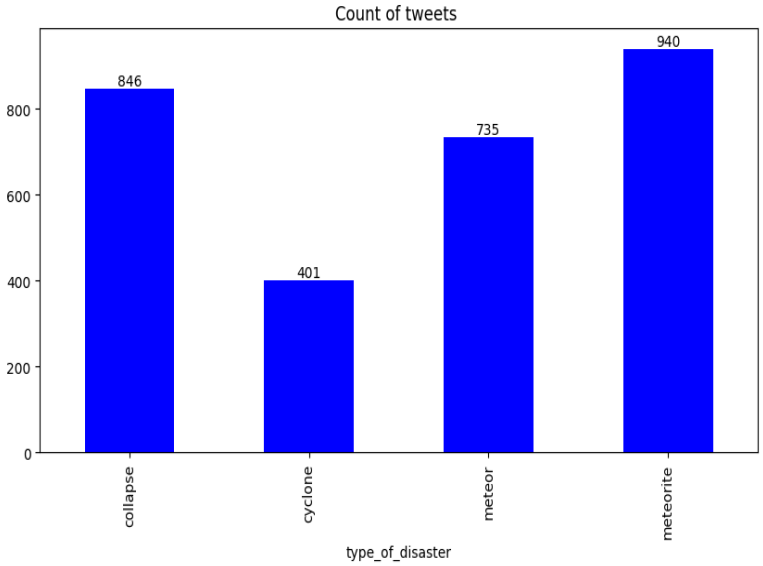


Figure 5.22: Count of Disaster Types in the Tweets-Data Dataset.

a./K-means Clustering

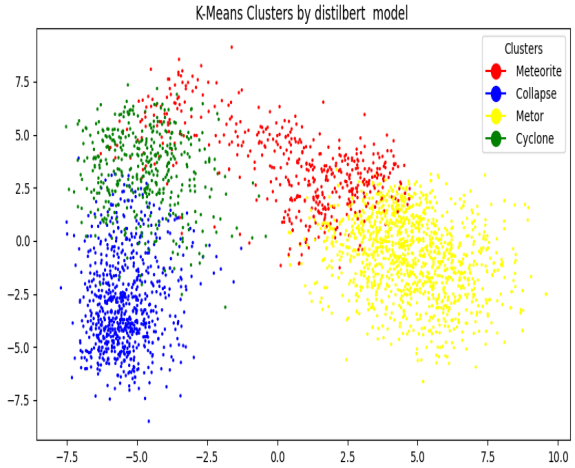


Table 5.26: Metric values K-means by Distil-BERT

Event	Precision	Recall	F1-score
Collapse	0.992	0.766	0.864
Cyclone	0.562	1.000	0.720
Meteor	0.443	0.208	0.283
Meteorite	0.527	0.679	0.593
Accuracy	**0.733**	**0.733**	**0.733**

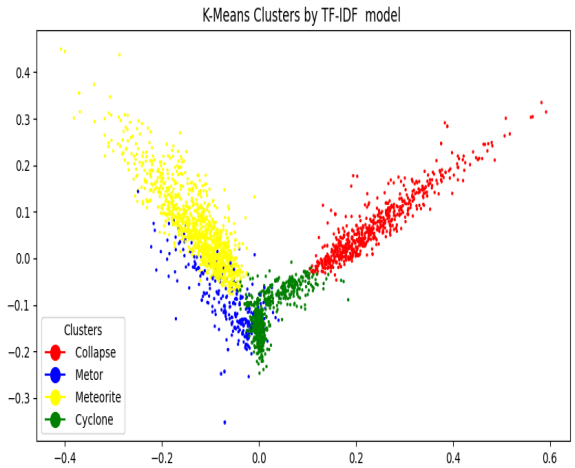


Table 5.27: Metric values K-means by TF-IDF

Event	Precision	Recall	F1-score
Collapse	0.992	0.765	0.864
Cyclone	0.562	1.000	0.719
Meteor	0.443	0.208	0.283
Meteorite	0.526	0.678	0.593
Accuracy	**0.629**	**0.629**	**0.629**

b./Agglomerative Clustering

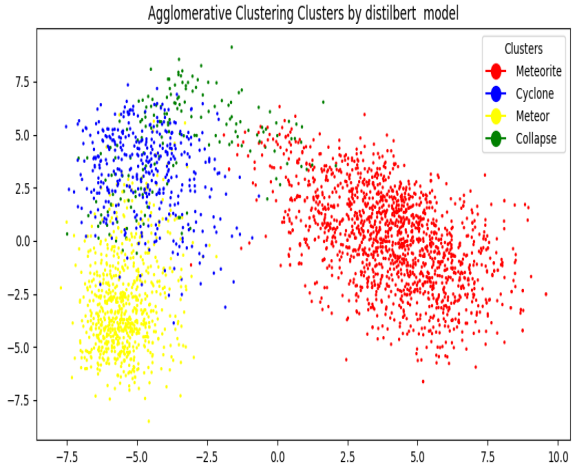


Table 5.28: Metric values Agglomerative clustering by DistilBERT

Event	Precision	Recall	F1-score
Collapse	0.995	0.871	0.929
Cyclone	0.951	0.985	0.968
Meteor	0.005	0.001	0.002
Meteorite	0.537	0.909	0.675
Accuracy	**0.680**	**0.680**	**0.680**

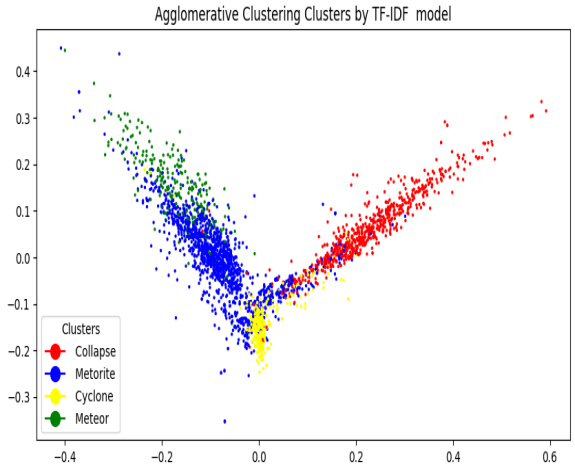


Table 5.29: Metric values Agglomerative clustering by TF-IDF

Event	Precision	Recall	F1-score
Collapse	0.973	0.774	0.862
Cyclone	0.924	0.940	0.932
Meteor	0.478	0.183	0.265
Meteorite	0.500	0.830	0.625
Accuracy	**0.666**	**0.666**	**0.666**

c./K-Medoids Clustering

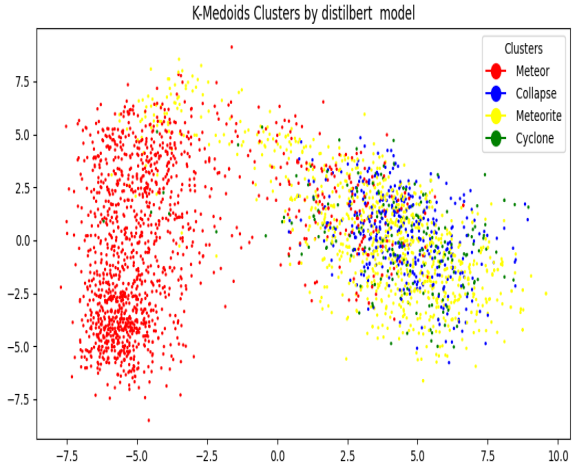


Table 5.30: Metric values K-Medoids clustering by DistilBERT

Event	Precision	Recall	F1-score
Collapse	0.741	0.968	0.839
Cyclone	0.014	0.014	0.014
Meteor	0.436	0.404	0.419
Meteorite	0.615	0.469	0.532
Accuracy	**0.534**	**0.534**	**0.534**

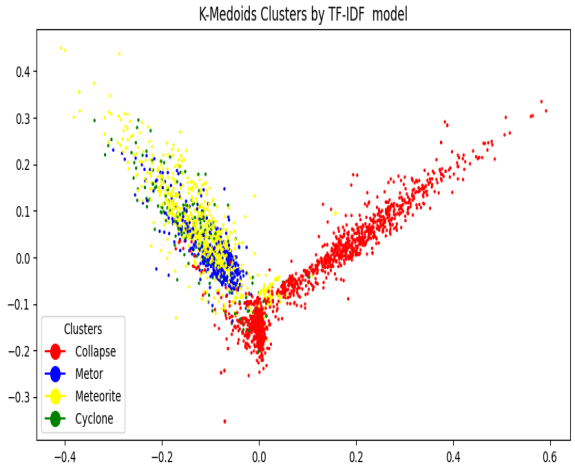


Table 5.31: Metric values K-Medoids clustering by TF-IDF

Event	Precision	Recall	F1-score
Collapse	0.570	0.951	0.713
Cyclone	0.023	0.012	0.016
Meteor	0.459	0.236	0.312
Meteorite	0.569	0.557	0.563
Accuracy	**0.516**	**0.516**	**0.516**

d./Comparison

Table 5.32: Silhouette score Comparison

Method	DistilBERT	TF-IDF
K-means	0.124	**0.02**
K-Medoids	0.06	0.01
Agglomerative	**0.143**	0.01

- By utilizing **DistilBERT**, the best Silhouette score was achieved using Agglomerative clustering. This approach likely benefits from DistilBERT's ability to generate dense and meaningful embeddings, which enable the hierarchical clustering to efficiently group similar data points.
- When employing the **TF-IDF** model, the highest Silhouette score was obtained with the K-means method. This result aligns well with TF-IDF's sparse vector representations, as K-means is particularly effective in handling high-dimensional, sparse data.

Table 5.33: Accuracy values Comparison

Method	DistilBERT	TF-IDF
K-means	**0.733**	0.630
K-Medoids	0.535	0.516
Agglomerative	0.680	**0.667**

- **K-means with DistilBERT** achieves the best accuracy for the English dataset "Tweets-data."

- **Agglomerative with TF-IDF** yields the highest accuracy when using the TF-IDF model.

5.3.2.5 Clustering performance of DistilBERT and TF-IDF

DistilBERT and TF-IDF represent contrasting approaches to text representation, showcasing their strengths in clustering tasks. DistilBERT consistently outperforms TF-IDF due to its ability to capture semantic context, particularly excelling in hierarchical methods like Agglomerative clustering. TF-IDF, while simpler and computationally efficient, is effective in scenarios requiring less emphasis on deep contextual understanding. This comparison highlights the importance of choosing the right method based on clustering goals and resource constraints.

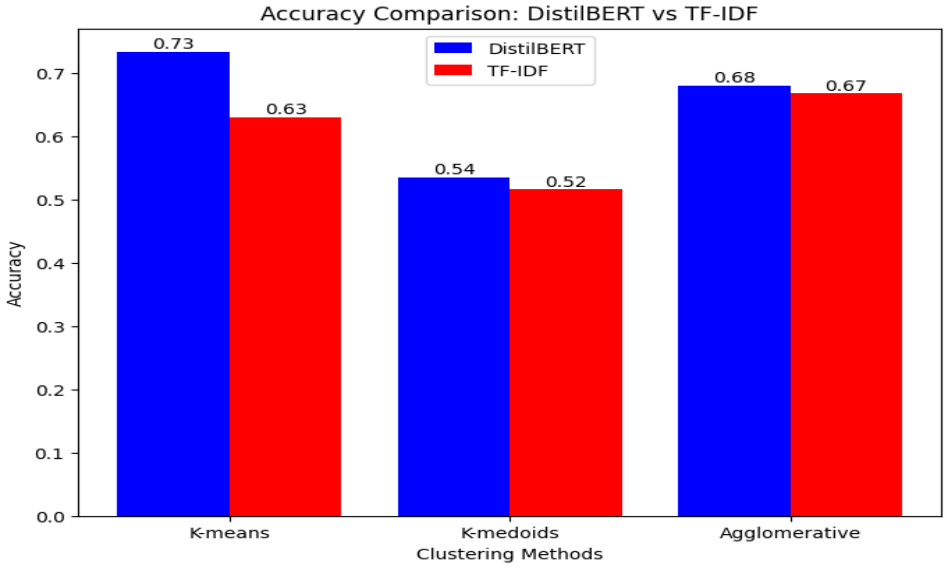


Figure 5.23: Clustering Accuracy Comparison: DistilBERT vs TF-IDF.

5.3.2.6 Clustering algorithms for Arabic tweets

The Figure 5.24 shows value distributions for the Arabic dataset ***Arabic_Dataset_Classifiction***.

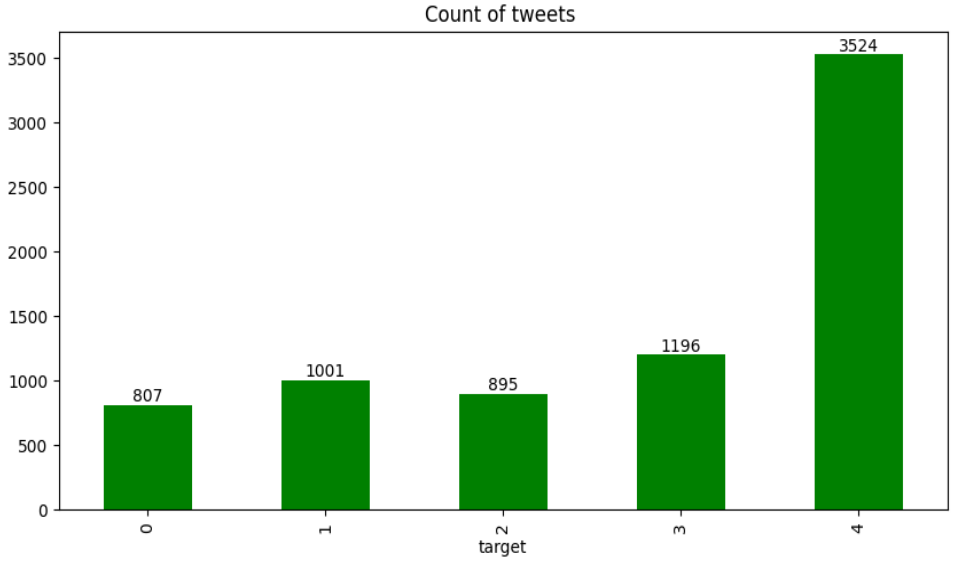


Figure 5.24: Count of Different Targets in the Arabic Dataset for Clustering.

a./K-means Clustering

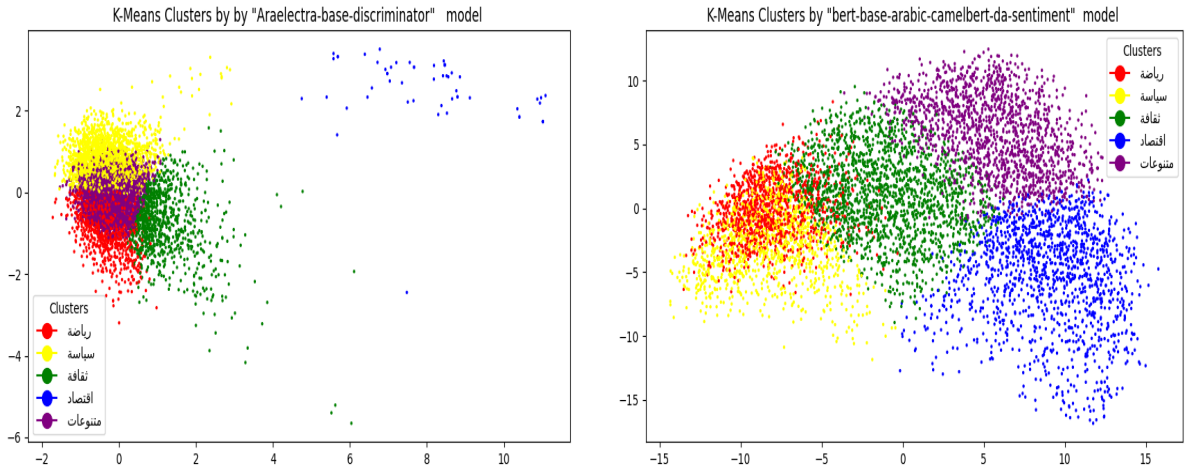
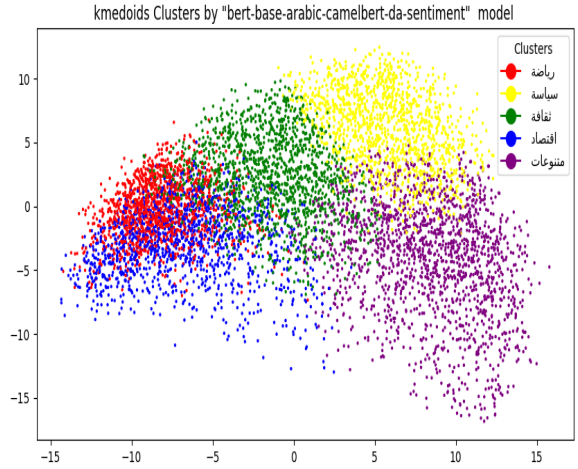


Table 5.34: Silhouette score Comparison using K-means

ARAELECTRA	CAMeLBERT
0.108	0.121

b./K-Medoids Clustering



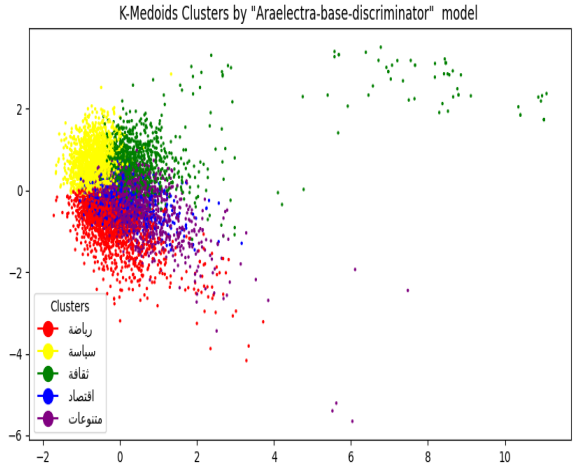
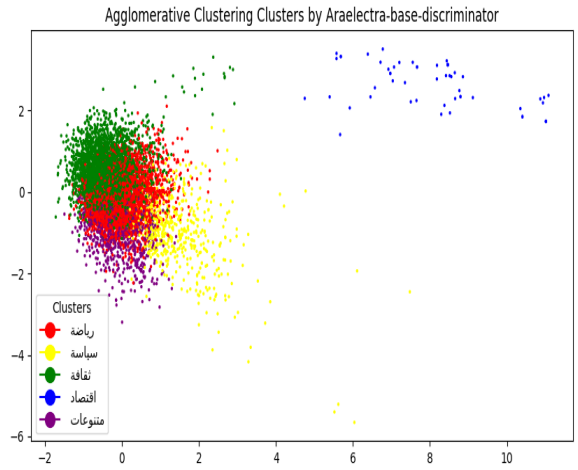


Table 5.35: K-Medoids Silhouette score Comparisons

ARAELECTRA	CAMeLBERT
0.011	0.108

c./Agglomerative Clustering



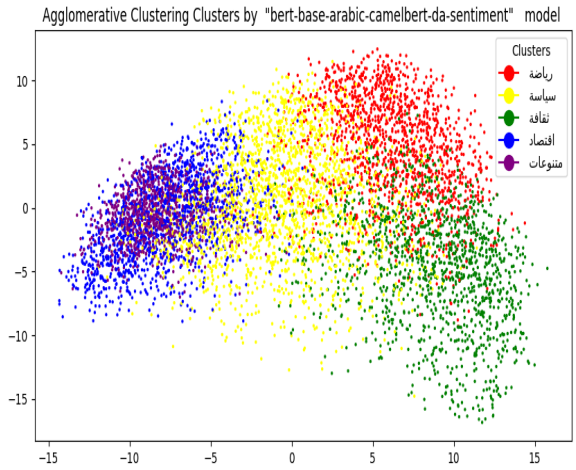


Table 5.36: Agglomerative Silhouette score Comparison

ARAELECTRA	CAMeLBERT
0.076	0.084

d./ Analyse

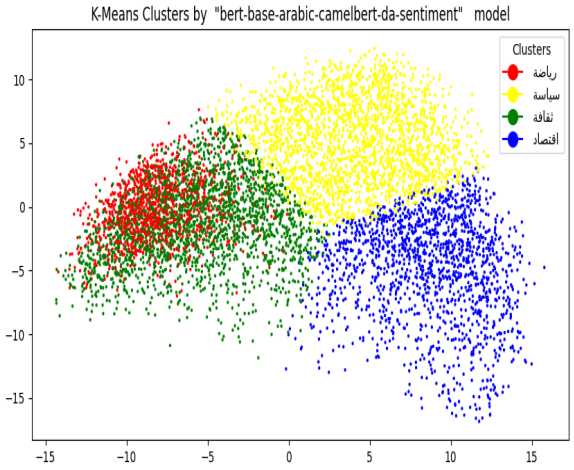
Table 5.37: Silhouette score Comparison

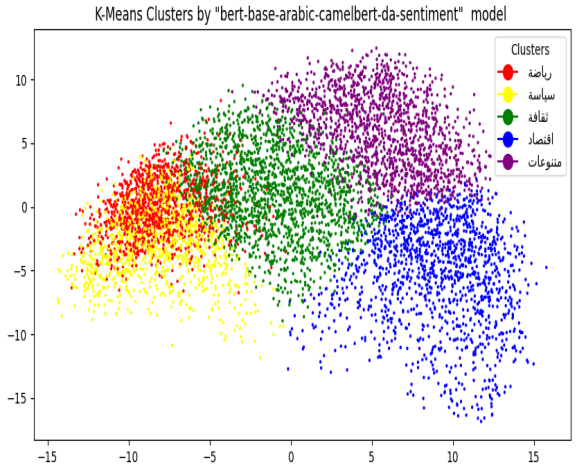
Method	ARAELECTRA	CAMeLBERT
K-means	**0.108**	**0.121**
K-Medoids	0.011	0.108
Agglomerative	0.076	0.084

- **K-means with CAMeLBERT** delivers the highest Silhouette score, suggesting its effectiveness in clustering tasks when paired with this model.
- K-means with ARAELECTRA also provides the best Silhouette score for that model, highlighting the consistency of K-means across different embedding techniques.
- **Algorithms with CAMeLBERT** benefits from adjustments in cluster numbers, implying that fine-tuning this parameter can significantly enhance performance.

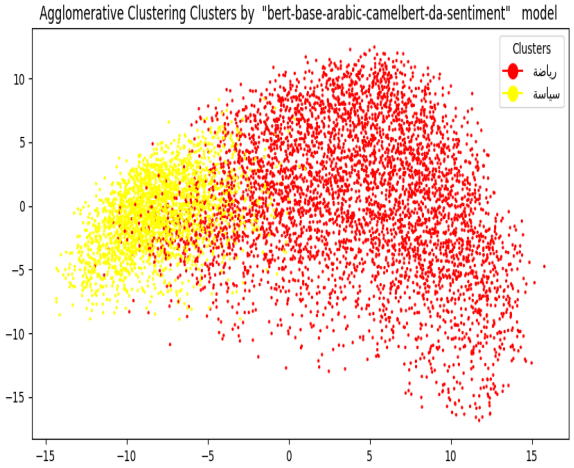
5.3.2.7 Analysis result According to the Number of Clusters

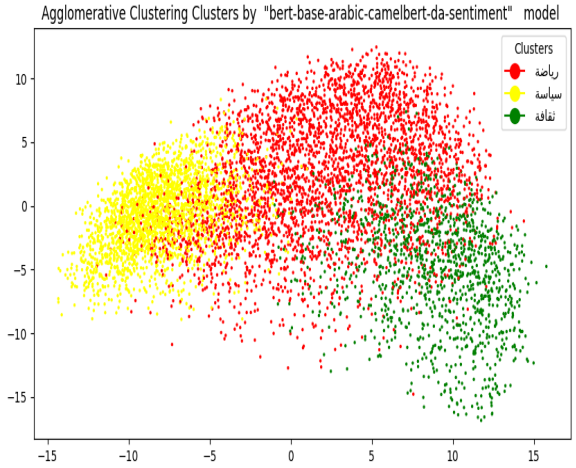
K-means with CAMeLBERT

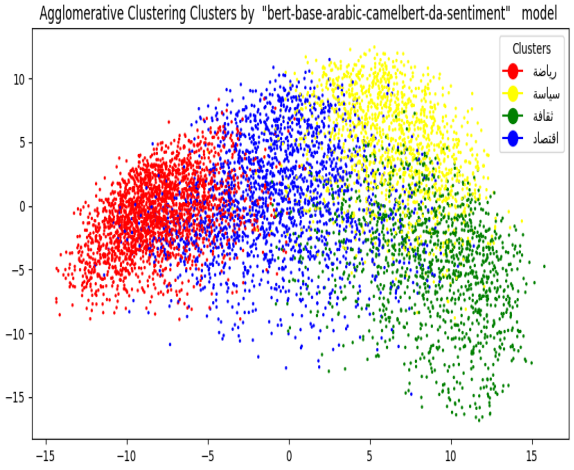


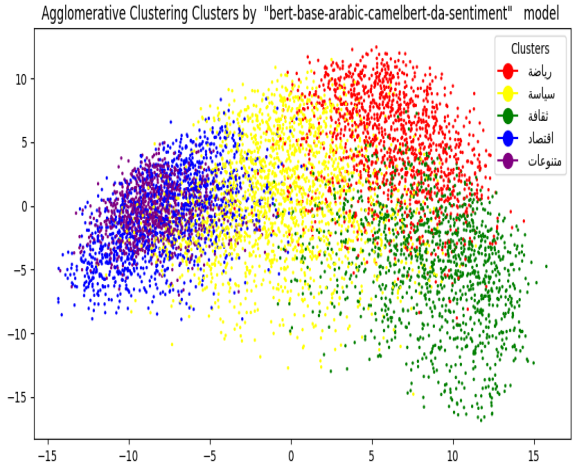


Agglomerative clustering with CAMeLBERT









K-Medoids clustering with CAMeLBERT

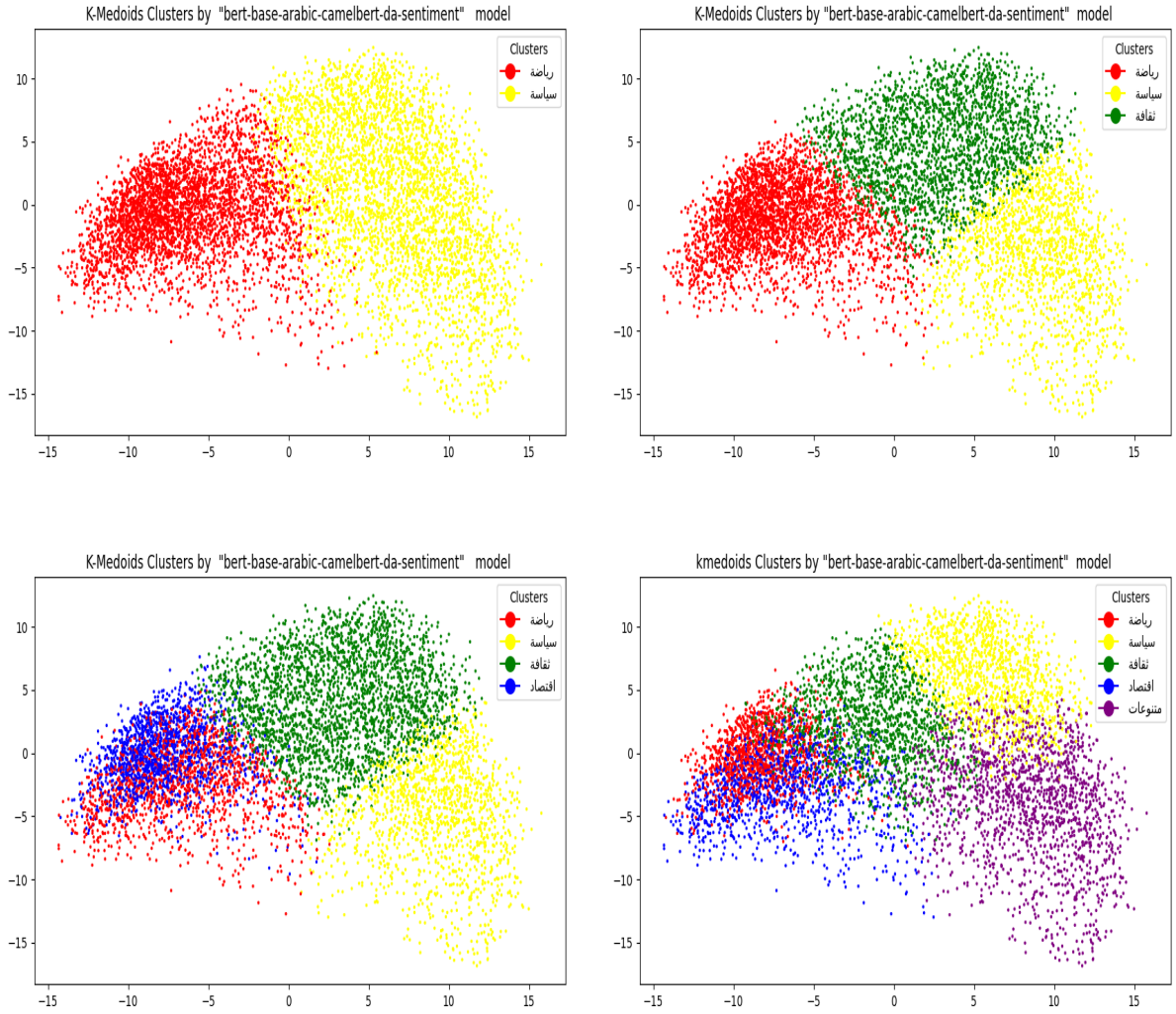
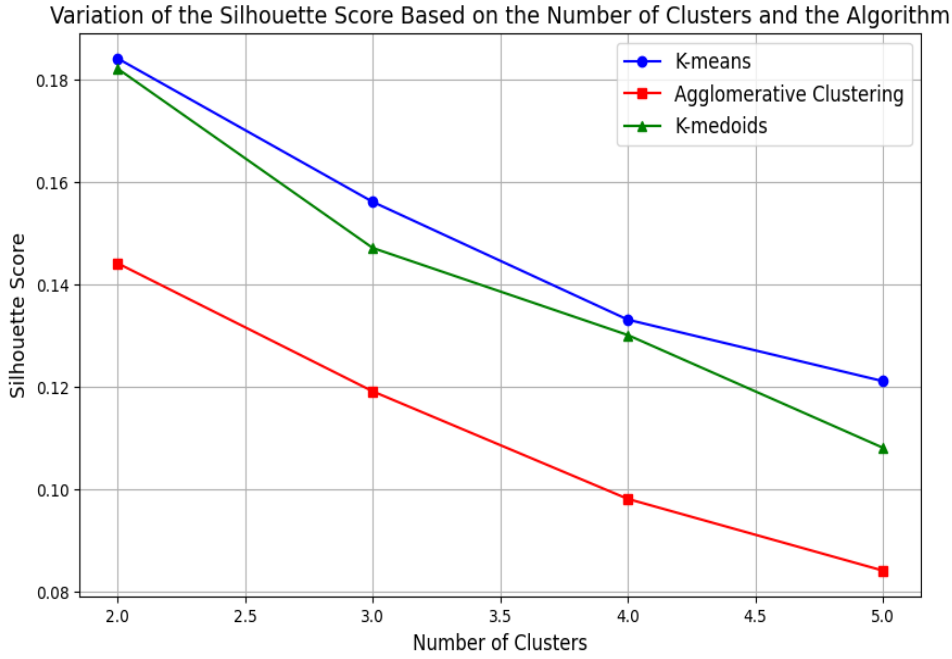


Table 5.38: Variation of the Silhouette Score According to the Number of Clusters and the Algorithm

Cluster Number	2	3	4	5
K-means	0.184	0.156	0.133	0.121
Agglomerative	0.144	0.119	0.098	0.084
K-Medoids	0.182	0.147	0.130	0.108

Figure 5.25: Variation of the Silhouette Score According to the Number of Clusters and the Algorithm



The silhouette scores provided for the three clustering algorithms—K-Means, Agglomerative Clustering, and K-Medoids—reveal a noticeable trend as the number of clusters increases. For all algorithms, the scores consistently decline, indicating a decrease in cluster cohesion and separation with a higher number of clusters. Among the algorithms, K-Means consistently outperforms the other two in terms of silhouette score, suggesting it provides better-defined clusters overall. K-Medoids follows closely behind K-Means, while Agglomerative Clustering scores the lowest in all cases. This suggests that K-Means might be the most suitable algorithm for this dataset if the objective is to achieve higher-quality clusters.

5.4 Conclusion

In this chapter, we presented our experimental part of our thesis, including the use of supervised and unsupervised methods for event detction and classification. We built and analyzed datasets, explored models like BERT and LSTM, and examined unsupervised approaches for English and Arabic data. Our experiments provided valuable insights, emphasizing methodological details, evaluation metrics, and result interpretation.

Chapter 6

Conclusions and Future Works

Contents

6.1 Conclusions

Social media has become a vital source of real-time information, shaping communication and public discourse. However, the vast volume of unstructured textual data presents significant challenges for event detection and classification. This research addressed these challenges by exploring deep learning techniques, particularly transformer-based models such as BERT and its variants. Through the construction of datasets and the development of advanced detection frameworks, this study demonstrated the effectiveness of these models, achieving competitive accuracy rates and improving classification performance.

In order to improve the process of valuable event detection and classification ,we proposed appropriate datasets, we proposed deep learning and transformers based models .We performed various experiments on 9 datasets using several deep learning models, such as RNN, LSTM and transformer based BERT model.

Our contibutions in this thesis consist of :

1. The development and generation of datasets for efficient event detection, classification and clustering. To this end:

    - Two new datasets, named *TweetsEvent and TweetsEvent310*, were collected by scraping tweets related to specific events. Events were categorized into 50 types based on Fillmore's theory of semantic frames.

- Five datasets from the well-known corpus Event2012 and the MAVEN dataset.
- One dataset from omdena.
- Two datasets from kaggle.

2. The developement of deep learning models for event detection and classification:

   a) We proposed an LSTM and BERT based model for event detection and classification in tweets. The experimental results show that the proposed model achieved an overall accuracy greater than 94.3% and hight F1 score values for the most of the event types.

   b) We proposed multi-classification method of disaster tweets to improve the analysis of social media data during emergencies. In this work we used LSTM networks on small data about 20 event types. Results show that our model demonstrates an accuracy of 71.7% in detecting events, which may be significant given the total amount of keywords (20) and the dataset is not very large(996 rows).

   c) We proposed a similarity based model, utilizing BERT models, base and large model, and two similarities(Cosine and Euclidean). This model used 3 datasets, Events35 dataset, that presents 35 event types, proved highest accuracy (90.5%)with Cosine similarity in large model. Maven49 proved, that presents 49 event types, (68.1%)with Euclidean in large model. MavenEvent70 dataset, that presents 70 event types, proved 71.8% with Cosine similarity in large model.

   d) We proposed a deep learning based model for short text with one-hot encoding achieved results 89.4%, 55% and 74.4% respectively for the datasets Event35, Maven66 and MavenEvent66.

   e) We proposed a clustering algorithms have been widely used for classifying tweets, we compared their performance on Arabic and English tweets using TF-IDF and pretrained BERT models. The analyzed algorithms include K-means, K-medoids, and Agglomerative clustering. The BERT models used are ARAELECTRA, CAMeLBERT, and DistilBERT. We conducted several experiments using various models with both Arabic and English tweets. The best result for Arabic dataset "Arabic_dataset_classification" obtained by K-means method using CAMeLBERT model. The best accuracy for English dataset "Tweets-data" obtained by K-means method using DistilBERT model.

6.2 Future Works

For the future avenue of this thesis, it will be interesting to consider the following extensions:

1. The construction of a Domain-Specific Corpus for Event Detection
To improve the accuracy and efficiency of event detection systems, a domain-specific corpus is essential. Such a corpus would be tailored to the language, terminology, and context of a specific field (e.g., healthcare, finance, or environmental studies), which would significantly enhance machine learning models. Key aspects include:

- **Data Collection:** Curating a diverse and representative dataset from the target domain to ensure comprehensive coverage of real-world scenarios.
- **Annotation Process:** Employing experts to label the data accurately for various event categories, ensuring high-quality training material.
- **Updates and Maintenance:** Continuously enriching the corpus to reflect emerging trends and changes in the domain, allowing models to stay relevant over time.

2. The building of new Arabic data representation models
Arabic NLP presents unique challenges due to the language's rich morphology, complex syntax, and wide range of dialects. Developing novel data representation methods can significantly enhance the processing of Arabic texts. Potential innovations include:

- **Morphological Analysis:** Designing representations that capture the nuances of Arabic morphology, such as root-based structures and inflectional variations.
- **Dialect Handling:** Developing embeddings or models that can differentiate between Modern Standard Arabic (MSA) and regional dialects while addressing code-switching issues.
- **Neural Approaches:** Leveraging state-of-the-art neural networks (e.g., transformers) to create pre-trained models specifically optimized for Arabic. These advancements can bridge gaps in Arabic NLP, making systems more inclusive and accurate across diverse applications like event detection sentiment analysis, machine translation.

3. Enhancement of Prediction Models Efficient prediction is vital for real-time applications, and optimizing both speed and accuracy ensures practical usability. Key areas of focus include:

- **Enhencing both the speed and quality of predictions:** Exploring alternative algorithms could indeed revolutionize the prediction phase, making it faster and more efficient.

- **Parallel Processing:** Implementing parallel computing techniques to accelerate the prediction process, particularly for large-scale datasets.

Bibliography

[1] Y. Venugeetha, R. Rathod, and R. Kumar, "Social networking sites in daily life: Benefits and threats," *Artificial intelligence and communication technologies*, pp. 51–64, 2022.

[2] L. Sigerson and C. Cheng, "Scales for measuring user engagement with social network sites: A systematic review of psychometric properties," *Computers in Human Behavior*, vol. 83, pp. 87–105, 2018.

[3] N. DUGUÉ, "Analyse du capitalisme social sur twitter," *thesis*, 2015.

[4] D. M. Boyd and N. B. Ellison, "Social network sites: Definition, history, and scholarship," *Journal of computer-mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

[5] J. Piskorski, J. Haneczok, and G. Jacquet, "New benchmark corpus and models for fine-grained event classification: To bert or not to bert?" in *Proceedings of the 28th international conference on computational linguistics*, 2020, pp. 6663–6678.

[6] M. Noui, A. Lakhfif, and M. A. Laouadi, "Event detection and classification in tweets using deep learning," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19 977–19 982, 2025.

[7] M. Noui, A. Lakhfif, and M. A. Laouadi, "Towards useful event detection and sentiment analysis of osn for disaster management," in *2024 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. IEEE, 2024, pp. 1–7.

[8] B. F. Mateusz Fedoryszak, Vijay Rajaram and C. Zhong, "Real-time event detection on social data streams," *article*, 2019.

[9] I. K. Nikolaos Panagiotou and D. Gunopulos, "Detecting events in online social networks: De
nitions, trends and challenges," *article*, 2016.

[10] J. M. J. Andrew J. McMinn, Yashar Moshfeghi, "Building a large-scale corpus for evaluating event detection on twitter," *article*, 2013.

[11] H. Becker, "Identi
cation and characterization of events in social media," *Thesis*, 2011.

[12] S. B. B. G. Laurie Serrano, Thierry Charnois and M. Bouzid, "Combinaison d'approches pour l'extraction automatique d'événements," *article*, 2012.

[13] Z. Saeed, R. A. Abbasi, and I. Razzak, "Evesense: what can you sense from twitter?" *Advances in Information Retrieval*, vol. 12036, p. 491, 2020.

[14] M. Nasim, A. Nguyen, N. Lothian, R. Cope, and L. Mitchell, "Real-time detection of content polluters in partially observable twitter networks," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 1331–1339.

[15] E. Tunggawan and Y. E. Soelistio, "And the winner is...: Bayesian twitter-based prediction on 2016 us presidential election," in *2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. IEEE, 2016, pp. 33–37.

[16] U. Yaqub, S. A. Chun, V. Atluri, and J. Vaidya, "Analysis of political discourse on twitter in the context of the 2016 us presidential elections," *Government Information Quarterly*, vol. 34, no. 4, pp. 613–626, 2017.

[17] P. Sharma and T.-S. Moh, "Prediction of indian election using sentiment analysis on hindi twitter," in *2016 IEEE international conference on big data (big data)*. IEEE, 2016, pp. 1966–1971.

[18] M. Adedoyin-Olowe, M. M. Gaber, C. M. Dancausa, F. Stahl, and J. B. Gomes, "A rule dynamics approach to event detection in twitter with its application to sports and politics," *Expert Systems with Applications*, vol. 55, pp. 351–360, 2016.

[19] J. Wang, L. Zhao, Y. Ye, and Y. Zhang, "Adverse event detection by integrating twitter data and vaers," *Journal of biomedical semantics*, vol. 9, no. 1, pp. 1–10, 2018.

[20] I. Arpaci, S. Alshehabi, M. Al-Emran, M. Khasawneh, I. Mahariq, T. Abdeljawad, and A. E. Hassanien, "Analysis of twitter data using evolutionary clustering during the covid-19 pandemic," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 193–204, 2020.

[21] S. Alotaibi, R. Mehmood, I. Katib, O. Rana, and A. Albeshri, "Sehaa: A big data analytics tool for healthcare symptoms and diseases detection using twitter, apache spark, and machine learning," *Applied Sciences*, vol. 10, no. 4, p. 1398, 2020.

[22] M. Akbari, X. Hu, N. Liqiang, and T.-S. Chua, "From tweets to wellness: Wellness event detection from twitter streams," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[23] N. Thapen, D. Simmie, and C. Hankin, "The early bird catches the term: combining twitter and news data for event detection and situational awareness," *Journal of biomedical semantics*, vol. 7, no. 1, pp. 1–14, 2016.

[24] M. A. Magumba and P. Nabende, "Ontology driven disease incidence detection on twitter," *arXiv preprint arXiv:1611.06671*, 2016.

[25] C. Arachie, M. Gaur, S. Anzaroot, W. Groves, K. Zhang, and A. Jaimes, "Unsupervised detection of sub-events in large scale disasters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 354–361.

[26] J. Yang, M. Yu, H. Qin, M. Lu, and C. Yang, "A twitter data credibility framework—hurricane harvey as a use case," *ISPRS International Journal of Geo-Information*, vol. 8, no. 3, pp. 111 https://www.mdpi.com/2220--9964/8/3/111, 2019.

[27] M. Mendoza, B. Poblete, and I. Valderrama, *EPJ Data Science*, vol. 8, no. 1, p. 3, 2019.

[28] M. A. Sit, C. Koylu, and I. Demir, "Identifying disaster-related tweets and their semantic, spatial and temporal context using deep learning, natural language processing and spatial analysis: a case study of hurricane irma," *International Journal of Digital Earth*, vol. 12, no. 11, pp. 1205–1229 https://doi.org/10.1080/17\protect\protect\leavevmode@ifvmode\kern+.1667em\relax538\protect\protect\leavevmode@ifvmode\kern+.1667em\relax947.2018.1\protect\protect\leavevmode@ifvmode\kern+.1667em\relax563\protect\protect\leavevmode@ifvmode\kern+.1667em\relax219, 2019.

[29] Z. Chen and S. Lim, "Collecting typhoon disaster information from twitter based on query expansion," *ISPRS International Journal of Geo-Information*, vol. 7, no. 4, p. 139, 2018.

[30] Y. Fang, J. Gao, Z. Liu, and C. Huang, "Detecting cyber threat event from twitter using idcnn and bilstm," *Applied Sciences*, vol. 10, no. 17, p. 5922, 2020.

[31] V. Vargas-Calderón, J. E. Camargo, H. Vinck-Posada *et al.*, "Event detection in colombian security twitter news using fine-grained latent topic analysis," *arXiv preprint arXiv:1911.08370*, 2019.

[32] S. Zimmerman, U. Kruschwitz, and C. Fox, "Improving hate speech detection with deep learning ensembles," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018, pp. 2546–2558.

[33] R. Kshirsagar, T. Cukuvac, K. McKeown, and S. McGregor, "Predictive embeddings for hate speech detection on twitter," *arXiv preprint arXiv:1809.10644*, 2018.

[34] Z. Zhang, Q. He, J. Gao, and M. Ni, "A deep learning approach for detecting traffic accidents from social media data," *Transportation research part C: emerging technologies*, vol. 86, pp. 580–596, 2018.

[35] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th international conference on World Wide Web companion*, 2017, pp. 759–760.

[36] Q. Le Sceller, E. B. Karbab, M. Debbabi, and F. Iqbal, "Sonar: Automatic detection of cyber security events over the twitter stream," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1–11 https://doi.org/10.1145/3\protect\protect\leavevmode@ifvmode\kern+.1667em\relax098\protect\protect\leavevmode@ifvmode\kern+.1667em\relax954.3\protect\protect\leavevmode@ifvmode\kern+.1667em\relax098\protect\protect\leavevmode@ifvmode\kern+.1667em\relax992.

[37] J. Lin, M. Efron, Y. Wang, and G. Sherman, "Overview of the trec-2014 microblog track," 2014.

[38] H. Hettiarachchi, M. Adedoyin-Olowe, J. Bhogal, and M. M. Gaber, "Embed2detect: Temporally clustered embedded words for event detection in social media," *Machine Learning*, no. 1, pp. 49–87 https://doi.org/10.1007/s10\protect\protect\leavevmode@ifvmode\kern+.1667em\relax994--021--05\protect\protect\leavevmode@ifvmode\kern+.1667em\relax988--7, 2022.

[39] P. T. J. Walker Orr and X. Fern, "Event detection with neural networks: A rigorous empirical evaluation," *article*, 2018.

[40] N. Panagiotou, I. Katakis, and D. Gunopulos, "Detecting events in online social networks: Definitions, trends and challenges," *Solving Large Scale Learning Tasks. Challenges and Algorithms: Essays Dedicated to Katharina Morik on the Occasion of Her 60th Birthday*, pp. 42–84, 2016.

[41] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," pp. 37–45, 1998.

[42] J. Yang and S. Counts, "Predicting the speed, scale, and range of information diffusion in twitter," 2010.

[43] N. Panagiotou, I. Katakis, and D. Gunopulos, "Detecting events in online social networks: Definitions, trends and challenges," in *Solving Large Scale Learning Tasks. Challenges and Algorithms*. Springer, 2016, pp. 42–84.

[44] Z. Saeed, R. A. Abbasi, O. Maqbool, A. Sadaf, I. Razzak, A. Daud, N. R. Aljohani, and G. Xu, "What's happening around the world? a survey and framework on event detection techniques on twitter," *Journal of Grid Computing*, vol. 17, no. 2, pp. 279–312, 2019.

[45] C. C. Leif Sigerson, "Scales for measuring user engagement with social network sites: A systematic review of psychometric properties," *article*, 2018.

[46] N. B. E. Danah m. boyd, "Social network sites: Definition, history, and scholarship," *article*, 2007.

[47] B. A. Seyedezahra Shadi Erfani, "Impacts of the use of social network sites on users' psychological well-being: A systematic review," *article*, 2018.

[48] P. C. T. Bogdan Batrinca, "Social media analytics: a survey of techniques, tools and platforms," *article*, 2014.

[49] S. Alhabash and M. Ma, "A tale of four platforms: Motivations and uses of facebook, twitter, instagram, and snapchat among college students?" *article*, 2017.

[50] D. K. Sibo Zhang, Yuan Cheng, "Event-radar: Real-time local event detection system for geo-tagged tweet streams," *article*, 2017.

[51] M. G. ANDREAS WEILER and M. H. SCHOLL, "Survey and experimental analysis of event detection techniques for twitter," *article*, 2017.

[52] A. P. B. Josimar Edinson Chire Saire, "Text mining approach to analyze coronavirus impact: Mexico city as case of study," *article*, 2020.

[53] M. O. Takeshi Sakaki and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," *article*, 2010.

[54] A. Culotta, "Towards detecting influenza epidemics by analyzing twitter messages," *article*, 2010.

[55] G. H. N. M. B. K. P. N. H. T. L. Caio Machado, Beatriz Kira and V. Barash, "News and political information consumption in brazil: Mapping the first round of the 2018 brazilian presidential election on twitter," *article*, 2018.

[56] D. D. Prasetya, A. P. Wibawa, and T. Hirashima, "The performance of text similarity algorithms," *International Journal of Advances in Intelligent Informatics*, vol. 4, no. 1, pp. 63–69, 2018.

[57] P. T. Pinky Sitikhu1, Kritish Pahi2 and S. Shakya, "A comparison of semantic similarity methods for maximum human interpretability," *article*, 2019.

[58] O. K. Pavel Stefanovi and R. Štrimaitis, "The n-grams based text similarity detection approach using self-organizing maps and similarity measures," *article*, 2019.

[59] M. Afzali and S. Kumar, "Comparative analysis of various similarity measures for finding similarity of two documents," *article*, 2017.

[60] P. F. Najlah Gali, Radu Mariescu-Istodor, "Similarity measures for title matching," *article*, 2016.

[61] P. Malakasiotis and I. Androutsopoulos, "Learning textual entailment using svms and string similarity measures," *article*, 2007.

[62] G. Anitha and S. Kuldeep, "Neural network approach for processing substation alarms," *International Journal of Power Electronics Controllers and Converters*, vol. 1, no. 1, pp. 21–28, 2015.

[63] D. Puri, R. Kumar, P. Sihag, M. S. Thakur, K. Perveen, F. M. Alfaisal, and D. Lee, "Analytical investigation of the impact of jet geometry on aeration effectiveness using soft computing techniques," *ACS omega*, vol. 8, no. 35, pp. 31 811–31 825, 2023.

[64] O. Alagöz and T. Uçkan, "Text clustering with pre-trained models: Bert, roberta, albert and mpnet," *NATURENGS*, vol. 5, no. 2, pp. 37–46, 2024.

[65] P. Arora, S. Varshney *et al.*, "Analysis of k-means and k-medoids algorithm for big data," *Procedia Computer Science*, vol. 78, pp. 507–512, 2016.

[66] A. Hodorog, I. Petri, and Y. Rezgui, "Machine learning and natural language processing of social media data for event detection in smart cities," *Sustainable Cities and Society*, vol. 85, p. 104026, 2022.

[67] S. Gadal, R. Mokhtar, M. Abdelhaq, R. Alsaqour, E. S. Ali, and R. Saeed, "Machine learning-based anomaly detection using k-mean array and sequential minimal optimization," *Electronics*, vol. 11, no. 14, p. 2158, 2022.

[68] R. L. Rosa, M. J. De Silva, D. H. Silva, M. S. Ayub, D. Carrillo, P. H. Nardelli, and D. Z. Rodriguez, "Event detection system based on user behavior changes in online social networks: Case of the covid-19 pandemic," *Ieee Access*, vol. 8, pp. 158 806–158 825, 2020.

[69] J. Ranganathan, N. Hedge, A. S. Irudayaraj, and A. A. Tzacheva, "Automatic detection of emotions in twitter data: a scalable decision tree classification method," in *Proceedings of the Workshop on Opinion Mining, Summarization and Diversification*, 2018, pp. 1–10.

[70] M. Z. Fauzi, A. Abdullah *et al.*, "Clustering of public opinion on natural disasters in indonesia using dbscan and k-medoids algorithms," in *Journal of Physics: Conference Series*, vol. 1783, no. 1. IOP Publishing, 2021, p. 012016.

[71] W. Sharif, S. Mumtaz, Z. Shafiq, O. Riaz, T. Ali, M. Husnain, and G. S. Choi, "An empirical approach for extreme behavior identification through tweets using machine learning," *Applied Sciences*, vol. 9, no. 18, p. 3723, 2019.

[72] S. Tam and Ö. Ö. Tanriöver, "Multimodal deep learning crime prediction using tweets," *IEEE Access*, vol. 11, pp. 93 204–93 214, 2023.

[73] J. Schmidhuber, S. Hochreiter *et al.*, "Long short-term memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[74] B. C. Mateus, M. Mendes, J. T. Farinha, R. Assis, and A. M. Cardoso, "Comparing lstm and gru models to predict the condition of a pulp paper press," *Energies*, vol. 14, no. 21, p. 6958, 2021.

[75] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A survey of text representation and embedding techniques in nlp," *IEEE Access*, vol. 11, pp. 36 120–36 146, 2023.

[76] S. Carta, S. Consoli, L. Piras *et al.*, "Event detection in finance using hierarchical clustering algorithms on news and tweets. peerj comput sci 7: e438," 2021.

[77] Z. Nasim and S. Haider, "Cluster analysis of urdu tweets," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 2170–2179, 2022.

[78] S. Akuma, T. Lubem, and I. T. Adom, "Comparing bag of words and tf-idf with different models for hate speech detection from live tweets," *International Journal of Information Technology*, vol. 14, no. 7, pp. 3629–3635, 2022.

[79] Z. Jiang, B. Gao, Y. He, Y. Han, P. Doyle, and Q. Zhu, "Text classification using novel term weighting scheme-based improved tf-idf for internet media reports," *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 6619088, 2021.

[80] T. Mikolov, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, vol. 3781, 2013.

[81] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[82] H. Liu, "Sentiment analysis of citations using word2vec," *arXiv preprint arXiv:1704.00177*, 2017.

[83] H. Peng, Y. Song, and D. Roth, "Event detection and co-reference with minimal supervision," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 392–402.

[84] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[85] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th international conference on World Wide Web companion*, 2017, pp. 759–760.

[86] A. Ollagnier and H. Williams, "Classification and event identification using word embedding," *neural networks*, vol. 6, p. 7, 2019.

[87] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[88] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[89] T. Nugent, F. Petroni, N. Raman, L. Carstens, and J. L. Leidner, "A comparison of classification models for natural disaster and critical event detection from news," in *2017 IEEE international conference on big data (Big Data)*. IEEE, 2017, pp. 3750–3759.

[90] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, "Transformer models for text-based emotion detection: a review of bert-based approaches," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 5789–5829, 2021.

[91] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[92] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[93] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He *et al.*, "A comprehensive survey on pretrained foundation models: A history from bert to chatgpt," *International Journal of Machine Learning and Cybernetics*, pp. 1–65, 2024.

[94] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[95] P. López-Úbeda, M. C. Díaz-Galiano, L. A. U. López, and M. T. Martín-Valdivia, "Pre-trained language models to extract information from radiological reports." in *CLEF (Working Notes)*, 2021, pp. 794–803.

[96] D. Wadden, U. Wennberg, Y. Luan, and H. Hajishirzi, "Entity, relation, and event extraction with contextualized span representations," *arXiv preprint arXiv:1909.03546*, 2019.

[97] A. Adhikari, "Docbert: Bert for document classification," *arXiv preprint arXiv:1904.08398*, 2019.

[98] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[99] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," *arXiv preprint arXiv:2003.00104*, 2020.

[100] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[101] A. R. Nair, R. P. Singh, D. Gupta, and P. Kumar, "Evaluating the impact of text data augmentation on text classification tasks using distilbert," *Procedia Computer Science*, vol. 235, pp. 102–111, 2024.

[102] J. Rashid, J. Kim, U. Naseem, and A. Hussain, "A distilbertopic model for short text documents," in *Proceedings of the 20th Annual Workshop of the Australasian Language Technology Association*, 2022, pp. 84–89.

[103] W. Antoun, F. Baly, and H. Hajj, "Araelectra: Pre-training text discriminators for arabic language understanding," *arXiv preprint arXiv:2012.15516*, 2020.

[104] A. Wadhawan, "Dialect identification in nuanced arabic tweets using farasa segmentation and arabert. arxiv 2021," *arXiv preprint arXiv:2102.09749*.

[105] K. E. Daouadi, Y. Boualleg, and O. Guehairia, "Systematic investigation of recent pre-trained language model for hate speech detection in arabic tweets," *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2024.

[106] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, "The interplay of variant, size, and task type in arabic pre-trained language models. arxiv 2021," *arXiv preprint arXiv:2103.06678*.

[107] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.

[108] C. Zhang, D. Lei, Q. Yuan, H. Zhuang, L. Kaplan, S. Wang, and J. Han, "Geoburst+ effective and real-time local event detection in geo-tagged tweet streams," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 3, pp. 1–24, 2018.

[109] H. Kayesh, M. S. Islam, and J. Wang, "On event causality detection in tweets," *arXiv preprint arXiv:1901.03526*, 2019.

[110] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[111] P. F. Wiemann, T. Kneib, and J. Hambuckers, "Using the softplus function to construct alternative link functions in generalized linear models and beyond," *Statistical Papers*, vol. 65, no. 5, pp. 3155–3180, 2024.

[112] I. Markoulidakis, G. Kopsiaftis, I. Rallis, and I. Georgoulas, "Multi-class confusion matrix reduction method and its application on net promoter score classification problem," in *Proceedings of the 14th PErvasive technologies related to assistive environments conference*, 2021, pp. 412–419.

[113] Y. George, S. Karunasekera, A. Harwood, and K. H. Lim, "Real-time spatio-temporal event detection on geotagged social media," *Journal of Big Data*, vol. 8, no. 1, p. 91, 2021.

[114] L. Ferrer, "Analysis and comparison of classification metrics," *arXiv preprint arXiv:2209.05355*, 2022.

[115] A. Dhiman and D. Toshniwal, "An approximate model for event detection from twitter data," *IEEE Access*, vol. 8, pp. 122 168–122 184, 2020.

[116] Z. Ghaemi and M. Farnaghi, "A varied density-based clustering approach for event detection from heterogeneous twitter data," *ISPRS international journal of geo-information*, vol. 8, no. 2, p. 82, 2019.

[117] S.-F. Yang and J. T. Rayz, "An event detection approach based on twitter hashtags," *arXiv preprint arXiv:1804.11243*, 2018.

[118] Z. Saeed, R. A. Abbasi, M. I. Razzak, and G. Xu, "Event detection in twitter stream using weighted dynamic heartbeat graph approach [application notes]," *IEEE Computational Intelligence Magazine*, vol. 14, no. 3, pp. 29–38, 2019.

[119] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[120] A. J. McMinn, Y. Moshfeghi, and J. M. Jose, "Building a large-scale corpus for evaluating event detection on twitter," pp. 409–418, 2013.

[121] A. Pandya, M. Oussalah, P. Kostakos, and U. Fatima, "Mated: metadata-assisted twitter event detection system," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems: 18th International Conference, IPMU 2020, Lisbon, Portugal, June 15–19, 2020, Proceedings, Part I 18*. Springer, 2020, pp. 402–414.

[122] M. Hasan, M. A. Orgun, and R. Schwitter, "Real-time event detection from the twitter data stream using the twitternews+ framework," *Information Processing & Management*, vol. 56, no. 3, pp. 1146–1165, 2019.

[123] M. Hasan, M. Orgun, and R. Schwitter, "Twitternews: real time event detection from the twitter data stream; 2016."

[124] A. Edouard, E. Cabrio, S. Tonelli, and N. Le Thanh, "Semantic linking for event-based classification of tweets," *International Journal of Computational Linguistics and Applications*, p. 12, 2017.

[125] A. Pandya, M. Oussalah, P. Kostakos, and U. Fatima, "Mated: Metadata-assisted twitter event detection system," in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 2020, pp. 402–414.

[126] B. Mazoyer, N. Hervé, C. Hudelot, and J. Cagé, "Représentations lexicales pour la détection non supervisée d'événements dans un flux de tweets: étude sur des corpus français et anglais," *arXiv preprint arXiv:2001.04139*, 2020.

[127] M. Hasanain, R. Suwaileh, T. Elsayed, M. Kutlu, and H. Almerekhi, "Evetar: building a large-scale multi-task test collection over arabic tweets," *Information Retrieval Journal*, vol. 21, no. 4, pp. 307–336, 2018.

[128] J. H. Zendah and A. Y. Maghari, "Detecting significant events in arabic microblogs using soft frequent pattern mining." *Journal of Engineering Research & Technology*, vol. 6, no. 1, 2019.

[129] X. Wang, Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin, and J. Zhou, "Maven: A massive general domain event detection dataset," *arXiv preprint arXiv:2004.13590*, 2020.

[130] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998.

[131] W. Zhang, B. Ingale, H. Shabir, T. Li, T. Shi, and P. Wang, "Event detection explorer: An interactive tool for event detection exploration," in *Companion Proceedings of the 28th International Conference on Intelligent User Interfaces*, 2023, pp. 171–174.

[132] T. Parekh, A. Mac, J. Yu, Y. Dong, S. Shahriar, B. Liu, E. Yang, K.-H. Huang, W. Wang, N. Peng *et al.*, "Event detection from social media for epidemic prediction," *arXiv preprint arXiv:2404.01679*, 2024.

[133] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Computational Intelligence*, vol. 31, no. 1, pp. 132–164, 2015.

[134] A.-M. Popescu and M. Pennacchiotti, "Detecting controversial events from twitter," pp. 1873–1876, 2010.

[135] M. Naaman, H. Becker, and L. Gravano, "Hip and trendy: Characterizing emerging trends on twitter," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 5, pp. 902–918, 2011.

[136] Z. Rezaei, B. Eslami, M. A. Amini, and M. Eslami, "Event detection in twitter by deep learning classification and multi label clustering virtual backbone formation," *Evolutionary Intelligence*, vol. 16, no. 3, pp. 833–847, 2023.

[137] Q. Li, Y. Chao, D. Li, Y. Lu, and C. Zhang, "Event detection from social media stream: Methods, datasets and opportunities," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 3509–3516.

[138] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," in *Proceedings of the international AAAI conference on web and social media*, vol. 5, no. 1, 2011, pp. 438–441.

[139] R. McCreadie, C. Macdonald, I. Ounis, M. Osborne, and S. Petrovic, "Scalable distributed event detection for twitter," in *2013 IEEE international conference on big data*. IEEE, 2013, pp. 543–549.

[140] D. Corney, C. Martin, and A. Göker, "Spot the ball: Detecting sports events on twitter," in *Advances in Information Retrieval: 36th European Conference on IR Research, ECIR 2014, Amsterdam, The Netherlands, April 13-16, 2014. Proceedings 36*. Springer, 2014, pp. 449–454.

[141] S. Fontalis, A. Zamichos, M. Tsourma, A. Drosou, and D. Tzovaras, "A comparative study of deep learning methods for the detection and classification of natural disasters from social media." in *ICPRAM*, 2023, pp. 320–327.

[142] Ø. Repp and H. Ramampiaro, "Extracting news events from microblogs," *Journal of Statistics and Management Systems*, vol. 21, no. 4, pp. 695–723, 2018.

[143] H. Adel, A. Amin Ali, and M. Kayed, "Event extraction from twitter using bert: A deep learning approach," *Kafrelsheikh Journal of Information Sciences*, vol. 5, no. 1, pp. 1–22, 2025.

[144] A. K. Pradhan, H. Mohanty, and R. P. Lal, "Edtbert: Event detection and tracking in twitter using graph clustering and pre-trained language model," *Procedia Computer Science*, vol. 233, pp. 481–491, 2024.

[145] R. Prasad, A. U. Udeme, S. Misra, and H. Bisallah, "Identification and classification of transportation disaster tweets using improved bidirectional encoder representations from transformers," *International journal of information management data insights*, vol. 3, no. 1, p. 100154, 2023.

[146] J. Kersten, J. Bongard, and F. Klan, "Combining supervised and unsupervised learning to detect and semantically aggregate crisis-related twitter content," in *International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, 2021.

[147] J. Ansah, L. Liu, W. Kang, J. Liu, and J. Li, "Leveraging burst in twitter network communities for event detection," *World Wide Web*, vol. 23, no. 5, pp. 2851–2876, 2020.

[148] A. P. G. P. Matteo Ceccarello, Carlo Fantozzi and F. Vandin, "Clustering uncertain graphs," *Proc. VLDB Endow.*, vol. 11, no. 4, pp. 472–484, https://doi.org/10.1145/3\protect\protect\leavevmode@ifvmode\kern+.1667em\relax186\protect\protect\leavevmode@ifvmode\kern+.1667em\relax728.3\protect\protect\leavevmode@ifvmode\kern+.1667em\relax164\protect\protect\leavevmode@ifvmode\kern+.1667em\relax143, year=2017.

[149] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," pp. 155–164, 2012.

[150] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "Dbpedia spotlight: shedding light on the web of documents," pp. 1–8, 2011.

[151] A. Guille and C. Favre, "Mention-anomaly-based event detection and tracking in twitter," pp. 375–382, 2014.

[152] M. Farnaghi, Z. Ghaemi, and A. Mansourian, "Dynamic spatio-temporal tweet mining for event detection: a case study of hurricane florence," *International Journal of Disaster Risk Science*, vol. 11, pp. 378–393, 2020.

[153] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[154] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," pp. 1532–1543, 2014.

[155] E. Schubert and M. Gertz, "Improving the cluster structure extracted from optics plots." in *LWDA*, 2018, pp. 318–329.

[156] A. H. Hossny and L. Mitchell, "Event detection in twitter: A keyword volume approach," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 1200–1208.

[157] A. H. Hossny, T. Moschuo, G. Osborne, L. Mitchell, and N. Lothian, "Enhancing keyword correlation for event detection in social networks using svd and k-means: Twitter case study," *Social Network Analysis and Mining*, vol. 8, no. 1, pp. 1–10, 2018.

[158] Y. Huang, Y. Li, and J. Shan, "Spatial-temporal event detection from geo-tagged tweets," *ISPRS International Journal of Geo-Information*, vol. 7, no. 4, p. 150, 2018.

[159] D. Birant and A. Kut, "St-dbscan: An algorithm for clustering spatial–temporal data," *Data & knowledge engineering*, vol. 60, no. 1, pp. 208–221, 2007.

[160] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[161] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," pp. 181–189, 2010.

[162] M. Garg and M. Kumar, "Twcm: Twitter word co-occurrence model for event detection," *Procedia computer science*, vol. 143, pp. 434–441, 2018.

[163] S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[164] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang, "Streamcube: Hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream," in *2015 IEEE 31st international conference on data engineering*. IEEE, 2015, pp. 1561–1572.

[165] A. Edouard, E. Cabrio, S. Tonelli, and N. Le Thanh, "Semantic linking for event-based classification of tweets," *International Journal of Computational Linguistics and Applications*, p. 12, 2017.

[166] J. Capdevila, J. Cerquides, J. Nin, and J. Torres, "Tweet-scan: An event discovery technique for geo-located tweets," *Pattern Recognition Letters*, vol. 93, pp. 58–68, 2017.

[167] Q. Li, A. Nourbakhsh, S. Shah, and X. Liu, "Real-time novel event detection from social media," pp. 1129–1139, 2017.

[168] N. Alsaedi, P. Burnap, and O. Rana, "Can we predict a riot? disruptive event detection using twitter," *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 2, pp. 1–26, 2017.

[169] D. Zhou, T. Gao, and Y. He, "Jointly event extraction and visualization on twitter via probabilistic modelling," in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, 2016, pp. 269–278.

[170] T. Hua, F. Chen, L. Zhao, C.-T. Lu, and N. Ramakrishnan, "Automatic targeted-domain spatiotemporal event detection in twitter," *GeoInformatica*, vol. 20, no. 4, pp. 765–795, 2016.

[171] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han, "Geoburst: Real-time local event detection in geo-tagged tweet streams," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016, pp. 513–522.