5W1H-Aware Approach for Retrieving Semantically Rich Multidimensional Events in Social Media Ecosystem

PhD Dissertation

by

Siraj Mohammed Bilal

A Dissertation Submitted to IT Doctoral Program, Addis Ababa University as Part of the Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Information Technology
(Information Retrieval)

Under the guidance of

Prof. Richard Chbeir

and

Dr. Fekade Getahun

Addis Ababa University
Addis Ababa, Ethiopia

August 2024

Addis Ababa University

School of Graduate Studies

IT Doctoral Program

This is to certify that the dissertation prepared by Siraj Mohammed Bilal, entitled "**Using 5W1H-Aware Approaches for Retrieving Semantically Rich Multidimensional Events in Social Media Ecosystem**" and submitted in fulfillment of the requirement for the Degree of Doctor of Philosophy in Information Technology (Information Retrieval) complies with the regulations of the university and meets the accepted standards with respect to quality and originality.

Signed by the advisors and examiners committee:

Examiner: ______________________________ Signature_______________ Date______________

Examiner: ______________________________ Signature_______________ Date______________

Advisor: ______________________________ Signature_______________ Date______________

Advisor: ______________________________ Signature_______________ Date______________

Abstract

5W1H-Aware Approach for Retrieving Semantically Rich Multidimensional Events in Social Media Ecosystem

Nowadays, multimedia digital ecosystems (e.g., *social media sites*) become a great source of user-contributed multimedia documents for many types of real-world events. Social media documents about events are multimedia (e.g., *image, video, text, and others)*, contain multiple features (describing different characteristics in it such as *5W1H*), and come from multiple sources. Such events can be used to construct Event Knowledge Graph (EKG) which is basic to retrieve semantically rich multidimensional events. However, social media documents cannot be used directly to construct such a knowledge graph. First, social media documents must be represented in order to detect multimedia events as well as their different types of semantic relationships. To achieve these tasks, it is necessary to carry out preliminary event-related tasks, such as detecting, linking, and representing events. By doing these, we can provide an event search API that presents a concise summary of events focused on temporal, spatial, semantic, and participant aspects. For this, we proposed a novel 5W1H-aware framework consisting of six modules. Each of these modules uses 4W elements. More specifically, we first represent social media documents that have been used to detect real-world events with their 4W elements based on event-only descriptive features. The detected events are used to identify the three main event relations categories (such as, spatial, temporal, and sematic) and many relation types under these categories by comparing dimensions over multimedia events. We then used a graph database to store event detection outputs as nodes and relationship identification outputs as edges to construct an Event Knowledge Graph (EKG). Finally, we integrated the EKG with an event retrieval API to retrieve events.

Each of these event-related tasks were evaluated using various datasets. For instance, the effectiveness of an incremental event detection algorithm was evaluated using the MediaEval 2013 dataset. The algorithm achieved an NMI score of 0.9914 and an F-score value of 0.9928 when the feature weights were assigned as 0.40 for participant, 0.30 for temporal, 0.35 for spatial, and 0.45 for semantic. Manually crafted spatial and temporal datasets are also used for evaluating the effectiveness of event relationship identification algorithm. Finally, the effectiveness of the developed EKG was evaluated through its downstream tasks, like retrieving similar nodes, whereas the event search API was evaluated via PageRank and search result relevance analysis. Results from the experiments showed that the proposed approach was more effective compared with alternative solutions.

Keywords: Multimedia Event, Event Detection, Event Relationships, Event Relationships Identification, Social Media Platforms, Social Media Documents Representation, Event Representation, and Event Knowledge Graph.

Dedication

To my beloved mother, **Ruqia Abdu Sitotaw**

Acknowledgement

First and foremost, I would like to express my gratitude to almighty God for enabling me to accomplish this task. I want to extend my sincere gratitude to my principal advisor ***Prof. Richard Chbeir***, for his valuable guidance, exceptional knowledge, consistent support, unwavering commitment, and constructive feedback provided during my PhD journey.

I would also like to express my appreciation to my co-advisor, ***Dr. Fekade Getahun***, for his insightful discussions, feedback, patience, motivation, and immense knowledge that assisted me throughout the preparation of this dissertation. His friendly approach, encouragement and guidance played a major role in the completion of this work. Indeed, without his support, the success of this work would have been unattainable. Thank you so much.

Beside my advisors, I would like to express my sincere appreciation to my beloved wife, **Jemila Kemal**, for her patience, support, and encouragement. I could not have achieved it without you, thank you so much for being by my side throughout my journey.

Last but not least, my humble gratitude goes to my family, for their endless love, support, morale and encouragement.

Declaration

I clearly declare that this dissertation entirely my original work and does not include any content that has been submitted for a degree in this or any other University. I clearly stated that all previously published materials used in this dissertation have been presented in accordance with academic rules. Furthermore, I understand that uploading an electronic copy of my dissertation to Addis Ababa University's research collection repository is required and subject to the policy and procedures of Addis Ababa University.

Siraj Mohammed Bilal

Publications

Journal Articles:

- Mohammed, S., Getahun, F. & Chbeir, R. ***Semantic Event Relationships Identification and Representation Using Hypergraph in Multimedia Digital Ecosystem***. Journal of Intelligent Information Systems, 1-31. Springer. https://doi.org/10.1007/s10844-022-00732-6.
- Mohammed, S., Getahun, F. & Chbeir, R. (2023). ***Events Management in Social Media: A Systematic Literature Review***. Social Network Analysis and Mining, 13, 70 (2023). Springer. https://link.springer.com/article/10.1007/s13278-023-01079-8.

Conference Proceeding:

- Mohammed, S., Getahun, F., Chbeir, R. (2021). ***5W1H Aware Framework for Representing and Detecting Real Events from Multimedia Digital Ecosystem***. In: Bellatreche, L., Dumas, M., Karras, P., Matulevičius, R. (eds) Advances in Databases and Information Systems. ADBIS 2021. Lecture Notes in Computer Science, vol 12843. Springer, Cham. https://doi.org/10.1007/978-3-030-82472-3_6.

List of Abbreviations

Abbreviations	Descriptions
5W1H	"Who, What, When, Where, Why, and How"
AI	Artificial Intelligence
API	Application Programming Interface
AQL	ArangoDB Query Language
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM-CRF	Bidirectional long short-term memory with a conditional random field
CNN	Convolutional neural network
DB	Database
DSRM	Design Science Research Methodology
EKG	Even Knowledge Graph
ER	Event Retrieval
GPS	Global Positioning System
GQL	Graph query Language
HTML	HyperText Markup Language
IBM Graph	International Business Machines Corporation Graph
TF-IDF	Term Frequency - Inverse Document Frequency
IR	Information retrieval
JSON	JavaScript Object Notation
KB	Knowledge base
KG	Knowledge graph
LOD	Linked Open Data
LR	Logistic regression
MED	Multimedia Event Detection
MMDES	Multimedia Digital EcoSystem
NER	Named-Entity-Recognition
NMI	Normalized Mutual Information
OEKG	Open Event Knowledge Graph
POS	Part-of-speech
RDF	Resource Description Framework
SGPA	Graph Partitioning Algorithm

SPARQL	SPARQL Protocol and RDF Query Language
SVM	Support Vector Machine
UNIX	Uniplexed Information and Computing System
URL	Uniform Resource Locator
UTC	Universal Time Coordinated
XML	Extensible Markup Language
YAGO	Yet Another Great Ontology

Contents

List of Tables

List of Figures

CHAPTER 1
INTRODUCTION

In contemporary society, several forms of Digital EcoSystems have emerged due to the advancements in digital technologies and innovations. These include Bank-oriented Digital EcoSystems [**1**], Healthcare-oriented Digital EcoSystems [**2**], Industry-oriented Digital EcoSystems [**3**], and Multimedia Digital EcoSystems [**4**]. Among all these digital ecosystems, this work focuses only on Multimedia Digital EcoSystems. Multimedia Digital EcoSystems (MMDES) refers to an interconnected network of digital technologies and platforms that enable the creation, sharing, and consumption of multimedia data [**5**]. Multimedia data usually refers to any type of data that utilizes a combination of image, text, video, number, audio, and animation to convey details information about objects (or events) [**6**]. It also consists of a variety of file formats (*e.g., text file format, audio file format, video file format, image file format*) used to store specific types of information.

Social media platforms, such as Instagram, Facebook, Twitter, YouTube, Flickr, and others, are some examples of popular Multimedia Digital EcoSystems (MMDES). These platforms have become part of our daily lives. According to the DIGITAL 2023 report, more than 4.76 billion people (representing 58.4% of the global population) use social media platforms in early 2023[1]. Such large numbers of social media users upload, share, and view enormous amounts of multimedia contents. For example, 694, 000 hours of videos are uploaded to YouTube every minute and there are 15 billion daily views on YouTube for YouTube shorts[2]. On the photo-sharing website Flicker, 25 million photos are uploaded daily and a total of 10 billion photos have been shared[3]. Twitter has over 353 million people who watched 2 billion videos every day[4]. Currently, photo and video posts outperform other types of content, such as text-based posts. In other words, multimedia content has taken the biggest part and has been growing year-on-year because of a variety of reasons: 1) multimedia content is more engaging than text-only content and can be used

[1] https://datareportal.com/reports/digital-2023-global-overview-report
[2] https://blog.hootsuite.com/youtube-stats-marketers/
[3] https://photutorial.com/flickr-statistics/
[4] https://www.bankmycell.com/blog/how-many-users-does-twitter-have

to represent/capture complex concepts in a simple and clear way; 2) the availability of social media sites and affordability of smartphones have made it easier for individuals and organizations to create and share multimedia content through social media platforms; and 3) social media platforms have features that allow users to create, upload and share multimedia contents. In general, on social media platforms, multimedia content posts receive the highest engagement of social media users than text-based posts. For example, photos along with text receive 91% more engagement among all types of content, followed by video posts (86%) and then text-based posts[5]. This shows that sharing multimedia data via social media platforms is increasing and getting more and more popular. In this way, multimedia data are being created, uploaded, shared, viewed, and coming from a variety of sources with diverse representations.

Although social media platforms offer opportunities to create and share multimedia data of events, they also present several challenges about event management. The first challenge is that there are different data representation techniques across social media sites. For instance, for representing multimedia contents and their metadata, Flicker and YouTube use XML, whereas Twitter uses JSON. The second challenge is data heterogeneity, which refers to different data types and formats. For example, the content of social media data includes videos, photos, texts, and audio which all require different types of data handling and processing techniques. The key reason for data heterogeneity is that social media contents are often generated by different users using varied technologies and resources. A third challenge is data incompleteness, where social media data may be incomplete in terms of dimensions, pictures, audios, and videos, due to non-expert users who generate, define, post, and share multimedia objects without following standard protocols. The fourth challenge is the unstructured nature of social media data. Additionally, social media data are typically characterized by inaccurate contents, ungrammatical text, irregular expressions/abbreviations, and often consist of short sentences [**8, 9**]. All the above challenges contribute to the difficulties associated with detecting relevant events from social media documents. To overcome these, the authors of [**8, 9, 10, 11**] have developed different machine learning approaches (such as *supervised, unsupervised, and hybrid*) that are capable of managing data heterogeneity, dealing with unstructured data, and accommodating diverse data representations in the event detection process.

---

[5] https://www.contentstadium.com/blog/2022-social-media-statistics-media-publishing-industry/

Although detecting events in social media documents have been widely/extensively studied, identifying the relationship between events and building Event Knowledge Graphs (EKGs) is challenged due to the following reasons:

- The existence of several types of relationships between two events, which are naturally independent but interrelated with each other. For example, temporal vs. causal, in which the cause of an event must occur before its effect in time.
- Social media data is multidimensional and can be represented in many ways. This makes it difficult to build a simple model of the relationships between events. Instead, we need a more complex model that can capture different types of relationships between events.
- The development of event-oriented knowledge graphs from social media data is still in its early stages and lacks a solid theoretical foundation and practical applications.
- The creation of semantically rich event knowledge graph is dependent on the quality of various event-related tasks, such as representing the semantic aspect of a multimedia object (or event), detecting events using semantic features of events, identifying semantic relationships between events, selecting graph database and graph model to store events and their relationships.
- Current Event Knowledge Graphs (EKGs) focus on representing entities indicating events and they do not capture all aspects of events, such as participant, location, time, purpose, and outcome. Additionally, they do not capture all types of relationships between events, such as semantic, temporal, and spatial.
- Events on social media sites are associated to multimedia and have multi-features (*e.g., temporal (When), spatial (Where), textual (What), and others*), come from several sources, and can be linked with temporal, spatial, and semantic relationships. Moreover, a single event can be broken into smaller parts and certain event relationships may involve more than two events. These properties of events pose challenges to construct an Event Knowledge Graph (EKG).
- Choosing the right graph database and graph data model for storing and querying multimedia events remains a difficult task.

To overcome the above challenges, in this study, we propose a 5W1H-aware framework, which consists of six major modules: 1) social media data crawling; 2) social media document representation; 3) social event detection; 4) relationship identification; 5) Event KG construction; and 6) event search API development. Each of these modules is presented in Chapters 4, 5, 6, and 7.

1.1. Motivation Scenario

Assume someone searching for relevant information that includes who is affected, when it happened, why an earthquake happened and related events that occurred in Hawassa, Ethiopia. Responding to this query can be done in two ways:

1. This person may conduct a search using search engines (*e.g., Yahoo, Google, Ask, etc.*). Unfortunately, the search results are often lists of information snippets (or lists of ranked URLs) that may not be relevant to the user query. This implies that the user is responsible for browsing, interpreting, and combining the results. Additionally, web search results do not consider basic event-related features or dimensions (*such as temporal, spatial, semantic, participant, and others*), making it difficult for users to find the specific information they need from search results.
2. A well-defined event search engine can be used to find specific events that occur in Hawassa, Ethiopia. The engine must extract location information (Hawassa, Ethiopia), the specific event (earthquake), and events related to it, as well as a list of information related to the event. A person can pose a query in an event search engine that has a well-defined event knowledge base capturing events and their relationships. Recently, Minal et al. [**9**] developed an approach that extracts events from social media documents. However, their approach did not associate events with other events and social events were not well organized.

Moreover, the advancement of social media platforms has made it easier for users to upload and share multimedia contents, such as texts, audios, photos, and videos. Consequently, there are massive amounts of multimedia contents (it may be event-related). These multimedia contents, however, are diverse in terms of data types, domains, dimensions, data presentations, and relationships. Due to these complex characteristics, it is challenging to detect events, establish connections between them and construct an event knowledge graph for retrieving events from it. Each of these challenges motivates us to undertake this research.

1.2. Statement of the Problem

Real-world events can be characterized using the five W's (Who, Where, When, Why, and What) and one H (How) dimensions, which have been well-researched in textual narrative forms like news articles [**12, 13, 14, 9**]. Extracting event-expressive features from such texts to identify an event is generally a straightforward process. In comparison with rich texts, multimedia content posted on social media platforms often does not have detailed textual descriptions. Additionally, there are various challenges related social media documents. These challenges affect both the representation of social media documents and approaches for detecting events. The first challenge, for example, is the existence of different data representation techniques across social media sites. The second challenge is data heterogeneity, which refers to different data types. A third challenge is data incompleteness, where social media data may be incomplete in terms of dimensions, due to users defining dimensions of multimedia objects without adhering standard protocols. The fourth challenge involves the presence of inaccurate contents, ungrammatical text, irregular expressions/abbreviations, and often consist of short sentences [**8, 9**]. All of these challenges collectively contribute to the difficulties associated with detecting relevant events from social media documents. To overcome these, the authors of [**8, 9, 10, 11**] have developed different machine learning approaches utilizing metadata that can be represented as 5W1H elements for detecting real-word events. Notably, some types of metadata, such as textual descriptions (*e.g., content description, titles, or tags*), are often unstructured, heterogeneous, ungrammatical, and noisy (*including, special symbols, abbreviations, non-standard words, and spelling errors*) [**8**]. Due to these, syntactical-based existing event detection methods are unsuitable. As a solution, researchers have used dimensions associated with each multimedia object to identify real-world events within social media documents. Some of these dimensions are: 1) semantic dimension (*or content descriptions, titles, and tags*); 2) temporal dimension (e.g., content capture time); 3) geospatial dimension (*or longitude and latitude values*); and 4) participant dimension (e.g., who created it and who participated in it). Overall, existing event detection approaches within social media documents, which rely on dimensions associated with each multimedia object to detect events, can be broadly categorized into three groups: 1) Classification [**15**]; 2) Cluster [**9**]; and 3) Hybrid [**8**]. These three machine learning approaches have been extensively researched as described in Chapter 3.

However, previous works did not address other important tasks related to events, such as identifying various types of event relationships, constructing an Event Knowledge Graph (EKG), and developing an event searching API that can retrieve semantically meaningful and multi-dimensional events from the EKG. Additionally, traditional pair-based knowledge graph representation techniques are insufficient to represent our complex events that involve multiple temporal, spatial, and causal relationships and some of these relationships may link more than two events at the same time. Therefore, a comprehensive knowledge graph database and knowledge graph model that can incorporate hypernode and hyperedge is needed. In addition to this, it also requires the identification of events and their different types of relationships.

1.2.1. Research Questions

To fill the gaps discussed earlier, this research aims to address the following questions:

- **RQ1:** How can the 5W1H approach be applied to multimedia event-related problems/tasks?
- **RQ2:** What are the key components of 5W1H event management framework?
- **RQ3:** How can real-world multimedia events and their different types of semantic relationships be identified?
- **RQ4:** What type of knowledge graph database and knowledge graph model should be constructed to represent multi-links, multi-dimensions, and multi-sources events?
- **RQ5:** How can users retrieve semantically meaningful multi-dimensional events?

1.3. Objective of the Research

1.3.1. General Objective

The general objective of this research is to study how the 5W1H approach can be applied to separate modules/ stages to provide users with more relevant information about events within Social Media Ecosystem.

1.3.2. Specific Objectives

The specific objectives are:

- Review the most relevant related works conducted in multimedia events;
- Study the characteristics of multimedia events for identifying event descriptive dimensions;

- Construct appropriate multimedia data representation model;
- Identify multimedia events and their different types of semantic relationships;
- Construct an event knowledge graph;
- Develop event searching API for retrieving events from its knowledge graph database;
- Evaluate the effectiveness of the proposed methods at each stage.

1.4. Methodology

1.4.1. Research Methodology

Scientific research is a systematic investigation that aims to find solutions to particular questions. The fundamental goals of scientific research are finding definitive answers and discovering new knowledge. Scientific research must be organized with plan, objective, systematic approach and follow a series of steps [**16**, **17**]. Often the terms "method" and "methodology" are used interchangeably, but these terms have different meanings and purposes. In [**16**, **17**], the authors define method as "*a specific research approach used to accumulate evidence regarding a phenomenon*". In other words, a method is a research tool that we use to gain deeper insights into a particular phenomenon in our studies. We can take surveys, interviews, and participant observations as an example. On the other hand, "methodology" refers to the general principles and procedures we adopt in addressing a problem. It encompasses several stages from problem identification to communication. In general, implementation driven approach, observational studies, applied studies, and simulations are widely applicable research methodology commonly used in computer science. Additionally, there are two commonly used methodologies, namely qualitative research and quantitative research. Quantitative research is methodology that utilizes logical, statistical, and mathematical methods to yield numerical data and facts. Its aim is to examine the causal connections between variables. Surveys, questionnaires, and observations are structured methods that are often employed in quantitative approaches. Qualitative research, on the other hand, is a research methodology that aims to comprehend human and social sciences by exploring the thoughts and feelings of individuals. Un-structured techniques, such as interviews, group discussions, and others are frequently utilized in this methodology. The ultimate objective of qualitative research is to explore discover ideas, and establish preliminary understanding used in the ongoing processes [**16**].

Design Science Research Methodology (DSRM) [**18**, **19**, **20**, **21**], which is common in disciplines such as computer science, information science, information technology, and computer engineering, focuses on artifact creation (*e.g., algorithm, prototypes, concepts, tools, and others*) to solve a particular problem in the identified domain in a unique and innovative way. DSRM focuses on the creation of new knowledge through the development and evaluation of innovative artifacts. This differs from traditional empirical research, which focuses on the discovery of existing knowledge through observation (or scientific data collection methods) and experimentation. DSRM typically involves six main stages from problem identification to disseminating research findings. This research adopts Design Science Research Methodology (DSRM) and its six stages [**18**]: *1) problem identification and motivation; 2) objectives of the solution; 3) design and develop artifact; 4) demonstration; 5) evaluation; and 6) communication*. Each of these stages is described and presented in Figure 1.1.

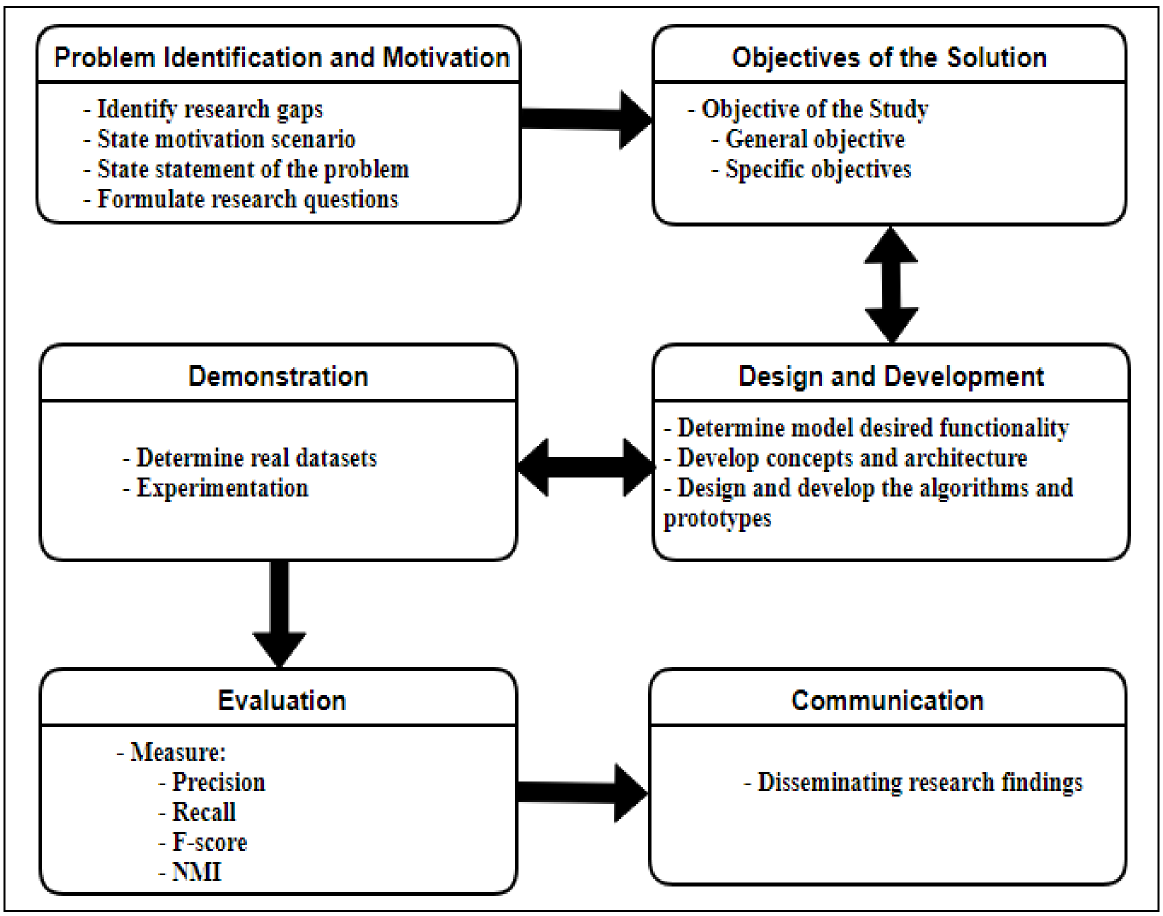


Figure 1.1: Stages and their activities used in this dissertation (adopted from [**18, 19, 20**]

Stage-1: Problem Identification and Motivation: This stage consists of three steps. Firstly, the research gap that needs to be addressed is identified. Secondly, the research motivation scenario and statement of the problem are indicated. Finally, research questions are formulated as presented in this chapter.

Stage-2: Objectives of the Solution: This stage utilized the problem definition established in stage-1 to propose the objectives of the research. In this research, our objective is to develop architecture/framework to identify, connect, represent, and retrieve multimedia events from social media platforms.

Stage-3: Design and Development: The stage aims to design and develop an artifact that addresses the identified problem in Stage-1. This stage explained in Chapters 5, 6, and 7. The entire design of the framework is outlined in Chapter 4. In Chapter 5, we present an incremental clustering algorithm for detecting and clustering multimedia events based on their dimensions similarities. For this, we used the combinations of four naturally different dimensions (*participant, temporal, spatial, and semantic*). We also design and develop an unsupervised machine learning algorithm, which takes the results of event detection as input for identifying different types of relationships among events, as described in Chapter 6. Finally, the design and the development of Knowledge graph and event retrieval system are presented in Chapter 7.

Stage-4: Demonstration: In this stage, our artifact was demonstrated. More specifically, this stage consists of two sub-steps. In the first step, real datasets are identified and prepared, as presented in Section 1.4.2. In the second step, various experiments are conducted using different kinds of datasets as presented in Chapters five, sex, and seven.

Stage-5: Evaluation: In this stage, we used Precision, Recall, F-score, and NMI metrics to assess the effectiveness of our approaches through testing and refinement.

Stage-6: Communication: The key contributions of this dissertation have been published in prominent peer-reviewed journals. This dissertation resulted in three publications, including one conference proceeding paper and two journal articles (*cf. list of publications section*).

1.4.2. Dataset

In social media ecosystem, multimedia event analysis involves the study and analysis of user-generated and shared multimedia contents, such as videos, images, texts, and audio files. Usually, this is done via machine learning-based analysis. For this specific study, we used two different unsupervised machine learning approaches: one for event detection and the other is event relationship identification. For evaluating the results of these two machine learning approaches, we used existing and manually crafted datasets (*extracted from social media source)*. These datasets are described in Table 1.1.

Table 1.1: Dataset description used in this dissertation

Event related tasks	Dataset used in our methods at each stage
Event detection	In order to assess the effectiveness of our event detection method, we used the MediaEval 2013 dataset [**22**], which contains more than 21,169 multimedia events with 437,370 images from Flickr. Every image in this dataset was described by metadata, such as upload time (100%), capture time (98.3%), spatial dimension (45.9%), uploader information (100%), and textual dimension (*including content description (37.9%), content titles (97.9%) and tags (95.6%)*). We focused on four of these dimensions for our evaluation: *uploader, spatial, capture time, and textual dimensions.*
Event relationship identification	• **Temporal Relationship Dataset:** Within this dataset, 310 multimedia events are annotated based on three temporal data types: image Timestamp, video Timespan and sentence-based time expressions from textual descriptions. Additionally, the dataset includes seven temporal relationship types between pairs of events, as detailed in Section 6.4.1. • **Spatial Relationship Dataset:** Similar to the temporal relationship dataset, we created 200 event representations containing event spatial features, such as points, lines, polygons, and spatial coverage representative points for each line and polygon coverage as well as all spatial relationship types between pairs of events (see, Section 6.4.2). • **Semantic Relationship Dataset:** For evaluating semantic relationship identification, we utilized the dataset developed by [**23**]

	containing 913 pairs of causal events that were annotated as either "causal" or "non-causal."

1.4.3. Machine Learning Approaches

Machine learning is a subfield of Artificial Intelligence and cognitive psychology that focuses on using data and algorithms to learn from experience and efficiently perform some human activities [**24**]. Machine learning algorithms were applied to identify events and their relationships. Such tasks can be solved with supervised, unsupervised, or hybrid learning approaches. This study adopts an unsupervised machine learning approach to detect events and their different types of relationships, as presented in Chapters 5 and 6.

1.4.4. Event knowledge Graph Construction Approach

There are two approaches to knowledge graph construction [**25**]: top-down and bottom-up. A top-down approach typically starts by creating a knowledge graph and then populating it with data. This approach may be more suitable if the intended knowledge graph is properly defined. On the other hand, a bottom-up approach starts by collecting data and uses it to build a knowledge graph. If there is no predefined structure for the knowledge representation (or if there is a large amount of heterogeneous information that needs to be integrated in the knowledge graph), bottom-up approach may be more suitable. An example of a bottom-up approach is building a knowledge graph for real-word occurrences in social media platforms. In this case, first, social media documents are collected and analyzed to identify events and their relationships. After this, the Event Knowledge Graph (EKG) is constructed. This study follows a bottom-up approach as described below.

- ***Step-1: Social Media Data Collection:*** This step involves gathering multimedia data from social media sites via Twitter[6], YouTube[7] and Flicker[8] APIs.

[6] https://www.twitter.com
[7] https://www.youtube.com
[8] http://www.flickr.com

- ***Step-2: Identify Multimedia Events:*** Here, we used the collected multimedia data in step-1 to identify real-world events along with their dimensions.
- ***Step-3: Define and Identify semantic Relationships:*** In this step, identified events in step-2 and their dimensions values were used to define and identify different semantic relationships among events.
- ***Step-4: Define the Knowledge Graph Formation:*** For this, we use hypergraph formation since we are dealing with complex events where one relationship involves many nodes (or events) and two events can be connected by different types of relationships.
- ***Step-5: Construct the Event Knowledge Graph:*** This step uses the results of the above steps (*2, 3, and 4*) and graph database model (e.g., neo4j) in order to store events and their links.
- ***Step-6: Query the Event Knowledge Graph:*** This step uses the knowledge graph constructed in Step 5 and searching API for retrieving multimedia events using Cypher query language.

1.5. Contribution of this Study

The key contributions of this dissertation are summarized as follows:

- A newly developed framework that considers 5W1H elements is presented. It has the ability to manage a variety of tasks, from social media document representation to event retrieval.
- We have provided basic definitions for each type of semantic relationship that can exist between events. Using these definitions, we then established a set of rules that serve as a basis for extracting semantic relationships. Finally, we introduced a machine learning algorithm that takes dimension values associated with each event as input and produces spatial, temporal, and causal relationships among events, as well as their sub-relationships as output.
- We presented an advanced Event-centric Knowledge Graph
- We introduced event search API for searching events from the EKG.
- We experimentally demonstrated the efficiency of each stages of the proposed approach.

1.6. Organization of the Report

The dissertation report is organized into eight chapters. This chapter states the problem statement, general objective, methodology, and contribution of the study. In Chapter 2, we provide background information and some basic concepts on detecting, linking, and representing events, as well as the interaction between information retrieval and knowledge graph are presented. Chapter 3 examines relevant works related to our study. Chapter 4 describes the problem definition and our proposed framework. Chapter 5 addresses the problem of multimedia event detection using an incremental clustering approach and multi-featured similarity functions. Chapter 6 presents the event relationship identification approach that identifies various semantic relationships between events. In Chapter 7, we explain the procedures for constructing an event knowledge graph and a search API. Finally, in Chapter 8, we present our conclusions and thoughts on future research directions.

CHAPTER 2

LITERATURE REVIEW

In the first part of the chapter, we provide background information about Social Media, Multimedia Digital EcoSystem (MDES), Multimedia data, Multimedia event, and the 5W1H approach. The second part of the chapter covers the most common techniques for event-related tasks (*e.g., representing dimensions of multimedia objects, detecting events and their relationships, etc.*) and the definitions of concepts, methods, and algorithms needed to understand our models that will be introduced in chapters 5, 6, and 7. The chapter also explains how event-related tasks, graph database, graph model, and graph query language can be used to create and complete the Event Knowledge Graph (EKG).

2.1 Social Media

The term social media refers to a variety of elements, such as websites, applications, devices (*e.g., desktops, smartphones, tablets, etc.*), network (*e.g., Wi-Fi*), content (*e.g., text, voice, video, picture, and the combination of them*) and users that enable the creation, sharing, and/or exchanging of information. Most of the information shared on social media platforms (*particularly from Twitter*[9], *YouTube*[10] *and Flicker*[11]) comprises multimedia data [**26**] that is heterogeneous in content and format, high-dimensional, partially incomplete in dimensions, massive in volume, poorly structured, and created by different users on different digital platforms [**9, 27**]. As a result, several challenges have been exhibited [**26, 5, 27**], such as: *1)* the lack of a common description technique to create, share, collect, and represent multimedia data; and *2)* detecting and analyzing multimedia events remains a difficult task. To overcome these challenges, researchers have proposed a novel framework, namely MDES [**26, 27**]. Detailed information about this framework is provided in the following section.

[9] https://www.twitter.com
[10] https://www.youtube.com
[11] http://www.flickr.com

2.2 Multimedia Digital EcoSystem (MDES)

In modern society, several forms of Digital EcoSystems have emerged due to the advancements in digital technologies and innovations. These include Bank Digital EcoSystems [**1**], Healthcare Digital EcoSystems [**2**], Industry Digital EcoSystems [**3**], and Multimedia Digital EcoSystems [**4**]. Here we explain how multimedia digital ecosystem is used in the analysis of social events, taking the event detection problem as an example. Multimedia Digital EcoSystem is an extension of the Digital EcoSystem (DES), where every interested participant can use social media platforms to generate, define, publish, collect, represent, and analyze multimedia contents [**26, 28**]. In general, MDES can be defined as an open, customer-centric, adaptable, and distributed system with features, such as maintainability, compatibility, self-organization, self-management, scalability, and many more others [**29**]. In MDES, there are massive user-defined multi-features social media data. These data are easily accessible as they can be collected through APIs and contains important event-related data. Therefore, social media data has become a good resource for event detection task. In the following sections, we provide basic concepts, such as multimedia data, metadata, multimedia event, and 5W1H dimensions, before describing the most common tasks related to events in MDES.

2.3 Multimedia Data

2.3.1 Definition

The term "Multimedia" comes from two Latin words: "Multi" means "numerous" and "media" means "middle." It is concerned with the representation of multi-media and multi-feature data. Multimedia data generally refers to data composed of different types of media (*e.g., numbers, texts, images*, *videos*, *animations and others*) to captures information about objects (or events) [**30**]. It also consists of various content/file formats (*including text*, *audio, video*, *and image file formats*) used to store certain types of information.

2.3.2 The Basic Components of Multimedia Data

Multimedia data, mainly from social media ecosystem, is made up of both content and metadata. The content refers to the actual media elements such as images, videos or audio, while metadata provides information about the media such as title, tags, date of creation, location, and participant(s) who created the multimedia object [31]. Metadata also consists of information related event categories (*e.g., entertainment, political, natural disaster, etc.*) and even relationship information (*e.g., temporal relation, spatial relation, and semantic relation*). Each of these dimensions can be modeled as the 5W1H elements. Then after, we can use them to *(i)* define each multimedia object (or event), *(ii)* detect multimedia events, and *(iii)* to identify different types of event relationships.

2.4 Multimedia Event

2.4.1 Definition

There has been a notable emphasis on defining the term "multimedia event" from the social media ecosystem perspective. In most case, multimedia events are defined on the basis of the dimensions associated with each multimedia object. Following this, we examined current definitions of multimedia events in the literatures. In the context of Multimedia Event Detection (MED-11) [**32**], an event (Ɛ) was defined as "*a complex activity occurring at a specific time and place involving people interacting with other people and/or objects; consisting of a number of human actions, processes, and activities that are loosely and tightly organized and that have significant temporal and semantic relationships to the overarching activity and is directly observable*". However, Multimedia Event Detection (MED-11) does not take into consideration other significant semantic dimensions (*such as What, Why, and How*) present in multimedia documents. Any definition of an event that solely focuses on the textual, spatial, temporal, and participant features of multimedia objects disregards the importance of semantically related information associated with multimedia objects. To address this issue, a new definition was introduced in [**9**], which defines an event Ɛ as "*an occurrence of a social or/and natural phenomenon happening within a certain time and location, having a certain textual description, and can be described/identified by the set of social media objects*". It is symbolized as:

$$\mathcal{E} = (\boldsymbol{eid}, \boldsymbol{T}, \boldsymbol{L}, \boldsymbol{S}) \tag{2.1}$$

where, ***eid:*** is a key-value used to identify distinct event and $\boldsymbol{T}, \boldsymbol{L},$ & $\boldsymbol{S}$**:** denote dimensions that pertain to the temporal (When), spatial (Where), and semantic (What) aspects of every multimedia object. Similar to the previous definition, this definition of an event may not be applicable to situations where there are participants (Who) involved and various relationships between events. To address this limitation, the researchers in [**33**] proposed a more comprehensive definition of an event that takes into account the participants and the relationships between events. This is represented as follows:

$$\mathcal{E} = (\boldsymbol{eid}, \boldsymbol{T}, \boldsymbol{L}, \boldsymbol{S}, \boldsymbol{P}, \boldsymbol{R}) \tag{2.2}$$

where,

- $\boldsymbol{P}$: denotes participant (Who) dimension,
- $\boldsymbol{T}$: represents temporal (When) dimension,
- ***L*:** represents geospatial (Where) dimension,
- ***S*:** represents textual features (What), and
- ***R*:** represents the relationships between events $(\mathcal{E}_1, \mathcal{E}_2)$, which indicate their temporal, spatial, and semantic connections. These relationships can be expressed as follows:

$$R(\mathcal{E}_1, \mathcal{E}_2)\ \epsilon (\boldsymbol{Tem}, \boldsymbol{Spa}, \boldsymbol{Cau}) \tag{2.3}$$

where,

- ***Tem:*** denotes temporal relationships, such as starts, before, equal, after, during, overlaps, and ends;
- ***Spa:*** represents group of spatial event relationships include distance-based, topological, and directional relationships, and;
- ***Cau:*** represents semantic event relationship, particularly the cause-and-effect relationship where an event $(\mathcal{E}_1)$ is identified as the cause and $(\mathcal{E}_2)$ is identified as the effect.

Furthermore, in [**34**], an event is defined as *"an occurrence causing change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic, time, and often associated with entities such as people and location."* Table 2.1 outlines the conclusions of event definitions discovered in the chosen papers. The symbol × implies that the approach disregards the corresponding dimension, whereas √ signifies the converse.

Table 2.1: Dimension types used in event definitions.

Definition	Participant (Who)	Temporal (When)	Spatial (Where)	Textual Features (What)	Semantic (What)	Other semantics (How & Why)
NIST [**32**]	√	√	√	√	×	×
Abebe, et al. [**9**]	×	√	√	√	√	×
Raad, et al. [**33**]	√	√	√	×	×	×
Mansour, et al. [35]	√	√	√	√	×	×
Dou, et al. [**34]**	√	√	√	√	×	×

Becker, et al. [36]	√	√	×	√	×	×
Dong, et al. [11]	×	√	√	√	×	×
Zhou, et al. [37]	√	√	√	√	×	√(How)
Choudhury, et al. [38]	√	√	√	×	×	×
Percentage	**77.7**	**100**	**88.8**	**77.7**	**11.1**	**11.1**

In Table 2.1, the primary findings are as follows: 1) all event definition tasks employed five dimensions, but not all researchers used all five dimensions. Each study mentioned above utilized only three or four dimensions to define an event; 2) The temporal dimension was utilized in all studies (100%); 3) 77.7% of the studies utilized both the participant dimension and the textual dimension; 4) 88.8% of the studies utilized the spatial dimension; and 5) 11.1% of the studies utilized semantic dimensions (such as What and How).

2.5 Multimedia Event Related Tasks

So far, we have understood the concept of multimedia event and metadata attached to each multimedia event. In this section, we describe the most common multimedia tasks related to events tasks like detecting multimedia event, identifying relationships among events, representing event and their relationships in the form of nodes and edge, and retrieving events from knowledge graph.

2.5.1 Multimedia Event Detection

This section explains the general sense of event detection and its main stages. More formally, we first analyzed what is multimedia event detection from the literatures. According to the findings of our analysis, multimedia event detection is a machine learning problem that can identify significant events from social media documents. This machine learning problem can be approached using supervised, unsupervised, or hybrid techniques ((cf. Section 2.6.1.3). Before describing these machine learning approaches, it is important to understand the general architecture of current event detection methods, which holds three basic phases*: 1)* collecting and prepossessing social media data; *2*) representing social media data; and *3)* detecting multimedia events, as illustrated in Figure 2.1. Each stage consists of a series of subtasks, which are discussed in the following sub-sections.

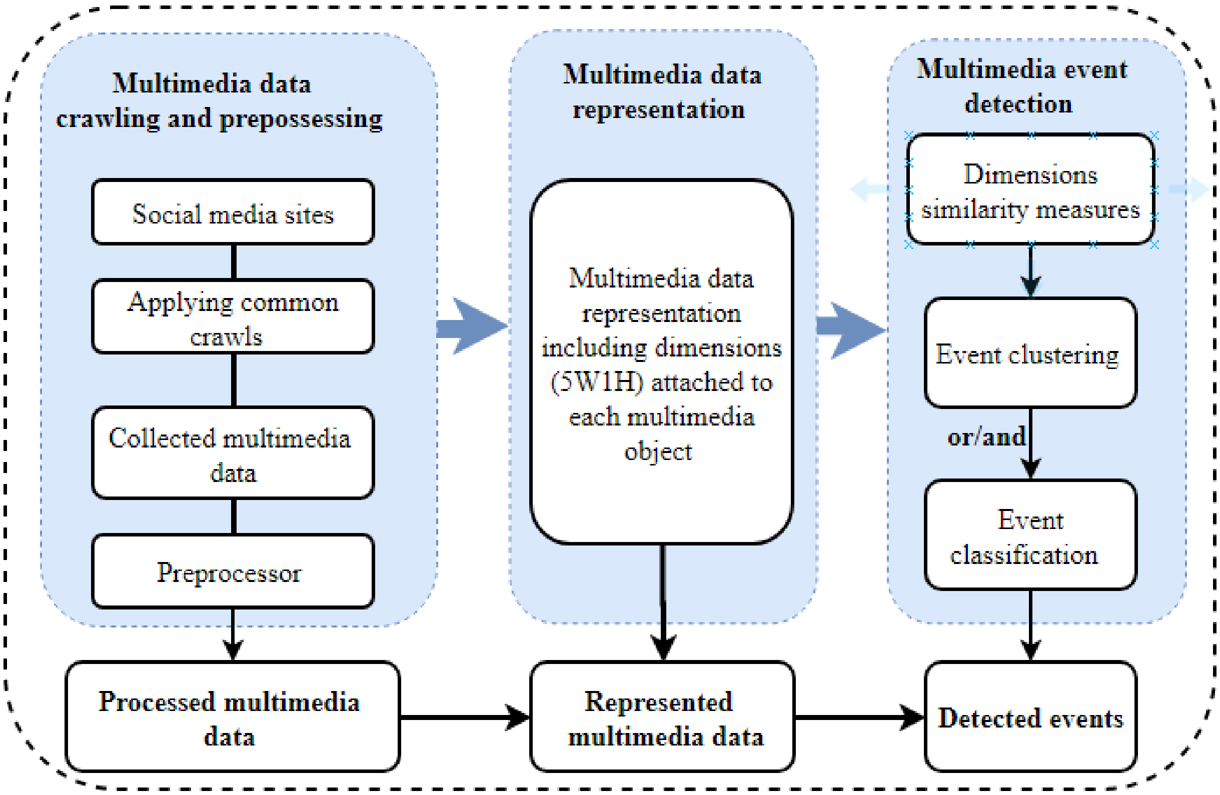


Figure 2.1: General Architecture of multimedia event detection

2.5.1.1 Multimedia Data Collecting and Preprocessing

Multimedia data can be obtained through Twitter, YouTube, and Flicker APIs. The outcome is a set of multimedia data with their respective dimensions. These data can present difficulties concerning heterogeneity in their structure and layout. To tackle this issue, scholars transformed all collected multimedia data, particularly metadata, into a consistent data format without changing their original forms. Subsequently, the metadata was divided into 5W1H concepts. Lastly, the scholars executed data pre-processing activities, mainly focusing on textual dimensions (*including titles, tags, and content descriptions*): removal of non-alphabetic characters (*e.g., symbols, mentions (@), URLs, HTML tags, and hashtags (#)*), stop word removal, tokenization, and translation of non-English textual descriptions (or words) into English. The outcome of the pre-processing task is processed multimedia data, which is utilized as input for multimedia data representation.

2.5.1.2 Multimedia Data Representation

In this part, we provide details background information about multimedia data representation. Specifically, we discuss the data models employed to represent metadata. To effectively represent multimedia data sourced from social media ecosystem, it is crucial to initially define each dimension accompanying to multimedia objects. The most widely used dimensions of multimedia object in the literatures are listed and defined as follows:

- **Temporal (When) Dimension:** It refers to the specific date/time of an event (*or object*). A single multimedia object published on social media sites may have several Timestamps; these could pertain to the past, present, or future. Such timestamps may include content capture time, content uploaded time, content modification time, and streaming time.
- **Spatial (Where) Dimension:** It refers to content-based locations, meaning the location within the multimedia content itself.
- **Textual/Semantic (What) Dimension:** This dimension focuses on three fundamental features produced by users, which are tags, titles, and descriptions of multimedia objects. In reality, researchers frequently utilized these three features as textual/semantic dimensions to apprehend the context within the multimedia object.
- **Participant (Who) Dimension:** It refers to an actor engaging in an event. This actor differs according to the nature/type of the event and may be one actor or group of actors.

After defining the commonly used dimensions attached to every multimedia object (or event), the subsequence task is to represent each dimension. To accomplish this, critical representation techniques have been introduced based on distinct dimension stamp, dimension coverage, and dimension coverage representative point. These entire representation techniques are presented in Table 2.2.

Table 2.2: Illustrates different dimension labels.

Terminologies	**Definition**
Distinct dimension stamp	It is a single value representing a discrete point within a specific dimension including temporal (T), spatial (L), participant (P), or semantic (S). It is useful to represent dimension values, such as temporal stamps that express as a single point, spatial stamps that represent a single point in space, and semantic stamps that capture a single concept in textual feature.
Dimension coverage	It refers to *(i)* the entire individuals involved in the occurrence, *(ii)* surface coverage where the event took place, *(iii)* the duration of time required to record the video/event, and *(iv)* the group of concepts in the event. These types representation is useful to represent polygon geospatial stamps, line geospatial stamps, duration-based time stamp, and so on.
Dimension coverage representative point	It is a single dimension stamp that represents the middle timestamp of temporal coverage, the center of geospatial coverage, the concept with the highest similarity score with respect to all other concepts in the semantic coverage. These types of representations are valuable in calculating the difference (or similarity) in dimensions between two multimedia objects in an efficient manner.

2.5.1.3 Multimedia Event detection Methods

This section presents a literature review on different machine learning methods employed in multimedia event detection. The current methods for multimedia event detection can be categorized into three primary machine learning strategies: *1)* supervised (*or classification, e.g., [**38, 15, 39**]*); *2)* unsupervised (*or clustering, e.g., [**9, 35, 40**]*); and *3)* hybrid (*or supervised + unsupervised, e.g., [**36, 41, 42**]*). Further elaboration on these machine learning approaches is provided below.

Clustering: It is a frequently used technique for identifying and clustering real-world events using unlabeled social media data. To do this, researchers use various unsupervised machine learning algorithms. These algorithms on social media platforms use multi-feature similarity techniques to group similar events into clusters based on their feature similarities. Generally, unsupervised machine learning techniques concentrate on clustering methods that can be categorized into four main groups. These consist of the following:

- **K-Mean Clustering:** It uses an iterative approach for clustering unlabelled data into distinct groups. This algorithm can be broken down into five steps [**30**]: 1) specify the number of clusters (***K***), in which we will group the event. 2) Set up the centroid (***X***), which represents the center of cluster (*cf. Figure 2.2 below*). Initially, the center of point is unknown, so we select random point from the dataset and use it as centroid for cluster. 3) Assign the data points (events) to the nearest cluster centroid by computing the distance/similarity between the data point and the centroid. 4) Re-compute the centroids of the newly produced clusters and 5) repeat steps 3 and 4. The same fashion has been utilized to detect social events on social media platforms. To illustrate, the authors of [**43**] present an approach that employs the K-means clustering algorithm to detect and cluster events by analyzing the Twitter text stream. In other words, their method processes untagged tweets and generates event clusters that represent diverse communities.

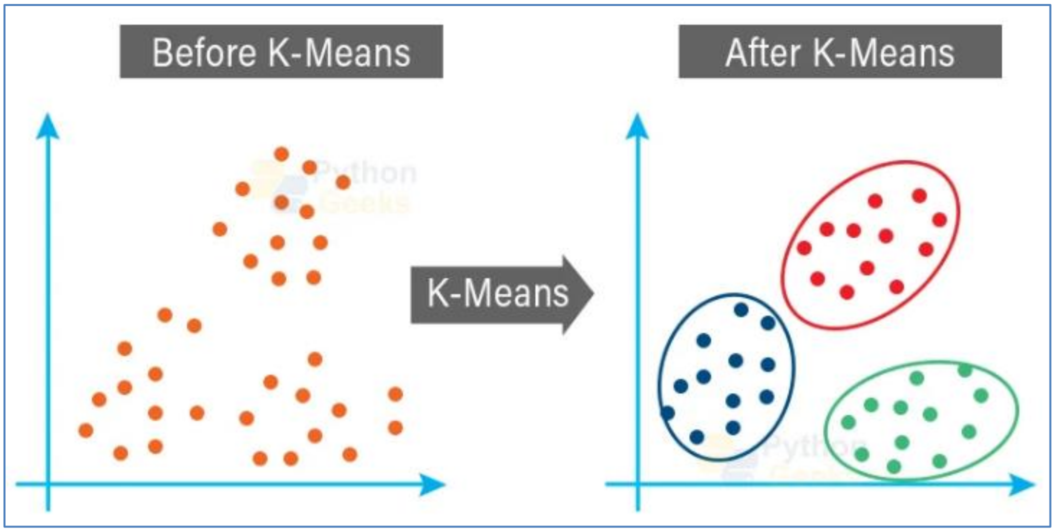


Figure 2.2: K-mean clustering algorithm, adopted from [**44**].

- **Hierarchical Clustering:** As previously stated, the K-means algorithm has a distinct drawback as it necessitates a predetermined number of clusters. This could lead to an inaccurate cluster results. To overcome this issue, a hierarchical clustering algorithm can be used to form a hierarchy of clusters in the form of a tree, also known as dendrogram (*see*

Figure. 2.3). Hierarchical clustering addresses two main types of problems: agglomerative and Divisive. The agglomerative approach follows a bottom-up methodology and involves three fundamental stages: 1) initially, the algorithm considers each data point as an individual cluster; 2) at each iteration, the algorithm computes the distance between each pair of clusters and creates a new cluster by merging the clusters that are close to each other; and 3) the process repeats until one cluster remains which represents the root of the tree. On the other hand, divisive hierarchical clustering is the exact reverse operation of the agglomerative algorithm as it is a top-down approach. These two types of hierarchical clustering algorithms are similar in two ways: (1) they construct a dendrogram; and (2) both of these techniques use similar linkage methods [**45**], such as complete linkage, .single linkage, and average linkage. Figure 2.4 shows these two hierarchical clustering algorithms.

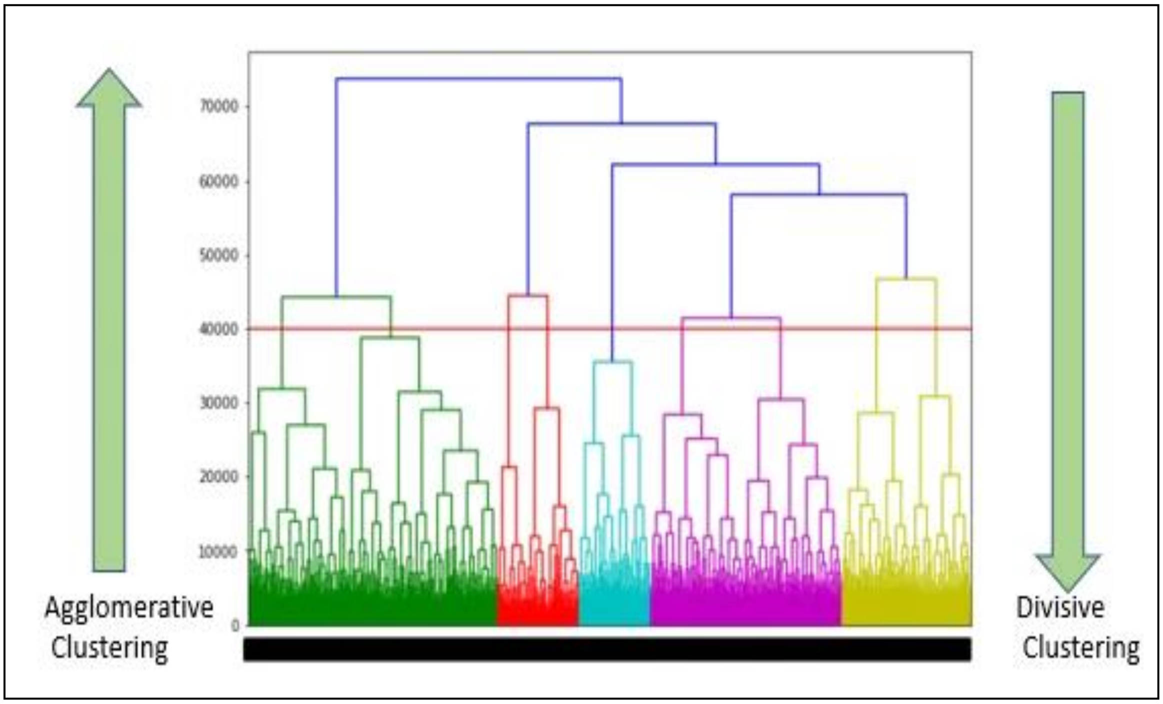


Figure 2.3: Agglomerative clustering vs. divisive clustering, adopted from [**46**].

- **Incremental Clustering:** Because of the vast and continuous flow of data on social media, researcher (*e.g. [36]*) has been forced to use the incremental clustering method. It works in a streaming fashion (or one event (or object) at a time principles). Following this fashion, we can address the clustering problem with the dynamically growing data [**47**]. The basic goal of this algorithm is to serially assign events to their respective clusters. This study adopts an incremental clustering algorithm for detecting and clustering multimedia events. The detail aspect of this algorithm is described in Chapter 5.

Classification: It uses labeled training dataset containing a set of examples to accurately categorize events into specific groups (*such as "event" or "non-event"*). In classification problem, there is model (f), an input variable (A) and an output variable (B). The model learns a function that maps input to output, which can be expressed as follows:

$$B = f(A) \tag{2.4}$$

Figure 2.4 below shows how the classification (supervised) method works.

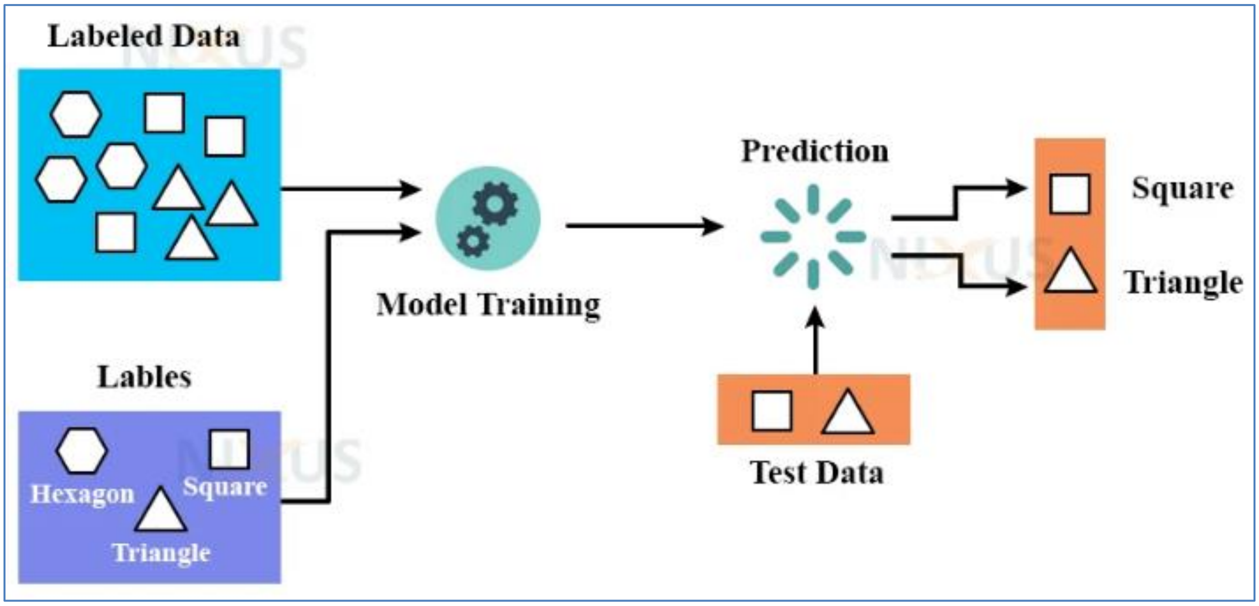


Figure 2.4: Classification method, adopted from[12].

Some common examples of classification algorithms include Naïve Bayes (NB), Support Vector Machine (SVM), Multinomial Naïve Bayes (MNB), and Decision Tree (DT). All these types of classification approaches have been utilized in the process referred to as social event identification. For example, the authors of [**38**] utilized classification-based event detection algorithm to identify events on shared Twitter. They utilized three classification-based event detection techniques: 1*)* the first technique was design based on unigram model (UNI), which employs four distinct classifiers (*such as Multinomial Naïve Bayes, Support Vector Machine, Naïve Bayes, and Decision Tree*). All classifiers were trained using 10-fold-cross-validation strategies and their experimental

[12] https://nixustechnologies.com/supervised-machine-learning/

results were reported in terms of precision, recall, and F-score; 2) UNI+META, which is a lexical features-based event detection method that uses only an SVM classifier, considering contextual lexical features of tweets (*i.e., temporal terms, social relation terms, and co-occurring words*) to detect events and classify them; and *3)* The third model of this research can detect events based on interaction features in addition to the lexical features.

Hybrid Method: This method falls between supervised machine learning and unsupervised machine learning, where the model accepts a mix of labeled and unlabeled datasets. This approach is particularly useful when there is limited labeled dataset, but a significant amount of unlabeled data available [**48**]. Figure 2.5 summaries the machine learning techniques used for event detection.

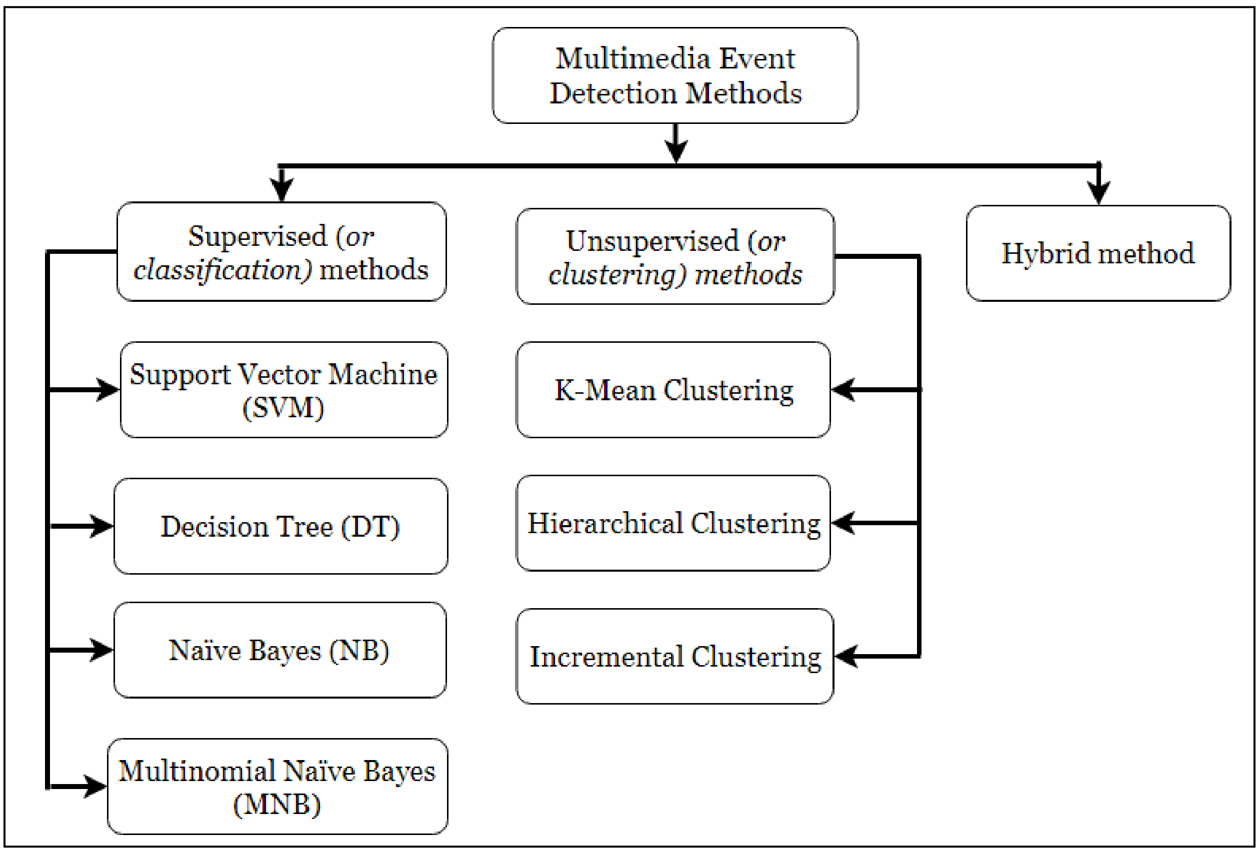


Figure 2.5: Machine learning methods used for event detection

2.5.1.4 Similarity Measures Used in Multimedia Event Detection

Similarity measures can be used in order to classify similar multimedia objects into clusters. In the course of our literature review, we determined the most often used similarity measures and organized them into three groups: *1)* syntactic similarity measures. Some examples of syntactic similarity measures include Haversine distance, Jaccard similarity, Euclidean distance, and Cosine similarity; *2)* semantic similarity measures (*including gloss-based, Lin similarity, node-based, edge-based, and hybrid-based*); and *3)* event similarity measures (*including minimum, maximum, average, and weighted sum functions*). In the next section, we demonstrate how researchers use these distinct similarity measures.

1. **Syntactic Similarity Measures:** are a type of similarity measure used to determine how closely related two multimedia objects by analyzing their syntactic structure, such as word order, length, and similarity in terms of textual, spatial, and temporal features. In the context of event detection, straightforward syntactic similarities have been extensively employed to assess the similarity between two events/objects. Each syntactic similarity is described formally as follows:

- **Cosine Similarity:** The cosine similarity method is often used by many event detection approaches for measuring similarity between textual features. In order to utilize cosine similarity, the approaches described in [**36**, **11**, **49**] use TF-IDF weighting technique to represent each textual dimension (*e.g., content descriptions, tags, and titles*) as vectors. Afterward, the cosine of the angle between two weighted vectors was applied to determine the similarity between two multimedia objects toward detecting and clustering events.
- **Jaccard Similarity:** To quantify how similar two textual features are, [**42**] utilized the Jaccard similarity metric. Similarly, [**49**] adopted the Jaccard similarity metric to measure the similarity of social media documents based on various feature types extracted from URLs including parsed URL, the Whole URL, and parsed query part of URL. Furthermore, [**37**] leveraged the Jaccard similarity metric to determine the extent of co-occurrence associated with a particular participant (Who).
- **Euclidean Distance:** The study presented in [**9**] utilizes the Euclidean distance as a similarity metric to determine the similarity between two multimedia objects (or events) based on the representative points of their temporal coverage. This is can be computed as follows:

$$Sim_T(e_{T1}, e_{T2}) = \frac{1}{1+Dis_T(e_1,e_2)} \in [0,1] \tag{2.5}$$

On the other hand, the work in [**11**] utilized Euclidean distance in order to calculate the likeness between two multimedia objects based on their geospatial coordinates.

- **Haversine distance:** Works in [**9**, **49**, **41**, **50**] utilized Haversine distance [**51**] to calculate the distance between two multimedia objects represented by their latitude and longitude coordinates. The Haversine defined as follows:

$$Dist_L(e_{L1}, e_{L2}) = 2 * r * \arcsin\left(\sqrt{sin^2\left(\frac{Lat2-Lat1}{2}\right) + \cos(Lat1) * \cos(Lat2) * sin^2\left(\frac{Lon2-Lon1}{2}\right)}\right) \quad (2.6)$$

where,

- **r** is the radius of Earth, typically measured in kilometers;
- $\mathbf{lat_1}$ and $\mathbf{lon_1}$ are the latitude and longitude of the first multimedia object;
- $\mathbf{lat_2}$ and $\mathbf{lon_2}$ are the latitude and longitude of the second multimedia object; and
- **arcsin** represents the invers of the sine function.

2. **Semantic Similarity Measure:** Semantic similarity is often more accurate than syntactic similarity measure, as it takes into account the meaning of the textual features. In other words, syntactic similarity fails to capture semantic (meaning) between two objects. To address this issue, works in [**9**, **52**] introduced a method that relies on WordNet [**53**] that computes the semantic similarity between two concepts. This method involves a number of steps, such as extracting textual dimensions from each multimedia object, determining semantic coverage, constructing a semantic network, and identifying a semantic coverage representative concept for each semantic coverage. The semantic similarity between the two concepts can be computed using only the representative points of semantic coverage in the semantic network. Four types of concept similarity measures in a semantic network are available: node-based, edge-based, gloss-based, and hybrid semantic similarities. They are described as follows:

- ***Node-based semantic similarity:*** is a technique that measures the similarity between two unique concepts based on their semantic meaning. Nevertheless, it concentrates only on concepts that are represented as nodes and does not consider a brief description given to a specific concept, which is called gloss. A gloss-based semantic similarity can overcome this limitation by incorporating descriptions as an additional factor in determining similarity.
- ***Gloss-based semantic similarity***: A gloss is a brief description given to a particular concept and can be used for three purposes: 1) to provide additional information about the meaning of a distinct concept; 2) to distinguish one word from other similar words; 3) to measure semantic similarity between two concepts.

- ***Edge-based semantic similarity:*** Unlike node-based and gloss-based techniques, edge-based semantic similarity measures the semantic similarity between two concepts by analyzing the links or edges connecting them.
- ***Hybrid semantic similarity:*** It integrates semantic similarity techniques based on node, edge, and gloss. In this semantic similarity metric, the semantic similarity between two concepts can be computed as follows:

$$Sim_s(C_1, C_2) = w_{Node} \times Sim_{Node}(c_1, c_2, KB) + w_{edge} \times Sim_{Edge}(c_1, c_2, KB) + w_{Gloss} \times Sim_{Gloss}(c_1, c_2, KB) \, \epsilon [0, 1] \quad (2.7)$$

where,

- $\boldsymbol{C_1}$ & $\boldsymbol{C_2}$ represent the concepts being compared,
- KB symbolizes the lexical knowledge base,

Furthermore, the study presented [**52**] utilized Lin similarity metric as a means to calculate the semantic similarity between two concepts (C_1, C_2). This Lin semantic similarity can be computed as follows:

$$Sim_{Lin}(C_1, C_2) = \frac{2 * Log_p(l_{so}(C_1, C_2))}{Log_p(C_1) + Log_p(C_2)} \quad (2.8)$$

where, C_1 & C_2 are the two concepts being compared. Lin similarity values range from 0 to 1, where 0 indicates no similarity between two concepts and 1 indicates complete similarity between them. In general, Lin similarity values closer to 1 indicate higher semantic similarity between two concepts.

3. **Event Similarity:** The above similarity measures demonstrate that there is no a single semantic similarity measure that encompasses all dimensions of multimedia events. Therefore, to determine the similarity between two multimedia events ($Sim(e_1, e_2)$), it is necessary to combine similarity measures corresponding to individual dimensions. To accomplish this, we must select a suitable ensemble function like average, weighted sum, maximum, or minimum functions. For instance, the research in [**9**] utilizes the weighted sum function to assess the similarity between two events. In this research, the formula for event similarity can be expressed as follows:

$$Sim(e_1, e_2) = w_T \times Sim_T(e_1, e_2) + w_L \times Sim_L(e_1, e_2) + w_S \times Sim_S(e_1, e_2) \, \epsilon [0, 1] \quad (2.9)$$

where,

- e_1 & e_2 are two multimedia events,
- $Sim_T, Sim_L,$ & Sim_S are temporal, spatial, and semantic measures, respectively, and
- $w_T, w_L,$ & w_S are weight values for each dimension provided by the researchers.

To summarize, the similarity measures used in the event detection methods are presented in Table 2.3.

Table 2.3: Summary of similarity measures used for detecting and clustering events.

Category	Approach
Syntactic similarity	Haversine distance, Euclidean distance, Jaccard similarity, and Cosine similarity.
Semantic similarity	Lin semantic similarity, node-based semantic similarity, edge-based semantic similarity, gloss-based semantic similarity, and hybrid-based semantic similarity.
Social event similarity	Maximum, minimum, weighted sum, and average functions.

2.5.1.5. Datasets Used in Multimedia Event Detection

In this section, we present datasets used for measuring the performance of multimedia event detection techniques in social media platforms. Some widely-used real-world datasets are listed in table 2.4, which provide a standardized benchmark for evaluating and comparing multimedia event detection algorithms. Table 2.4 also presents statistical information for each dataset.

Table 2.4: Description of the datasets used to evaluate methods for detecting multimedia events.

Papers	Dataset	#Social event	#Photo/tweet	Metadata
[**49**, **41**]	Upcoming dataset	9, 613	273, 842 photos	Unique ID, event title, event description, event tags, start/end time, and geo-spatial
[**41**, **15**]	Last.fm dataset	24, 958	594, 946 photos	Unique ID, start/end time, event title, URL to venue, venue name, participants, and geo-spatial

[**9**, **42**, **52**, **54**]	MediaEval 2013 dataset[13]	21,169	437, 370 photos	Geospatial, uploader information, textual features (*or tags*, *title*, *and description*), and temporal information.
[**9**]	MediaEval 2014 dataset[14]	17, 834	362, 578 photos	Temporal, geo-spatial, uploader, textual features.
[**35**]	ReSSED dataset[15]	21, 000	430, 000 photo	Uploader, temporal, geospatial, textual features.
[**55**]	CrisisLexT26 dataset[16]	26 crisis event types (about 1,000 in each type)	28, 000 tweets	keywords used to select tweets

2.5.1.6 Evaluation Metrics Used in Multimedia Event Detection

Selecting the appropriate evaluation metric is crucial in event detection. The selected evaluation metric must be appropriate to the problem at hand and should align with the research objective. The objective of this section is to present metrics applied to evaluate the effectiveness of event detection methods. Eight primary evaluation metrics were identified and illustrated in Table 2.5. Typically, there are two frequently used metrics to assess cluster quality, namely F-score and NMI. Among these two, Normalized Mutual Information (NMI) is deemed the most suitable evaluation metric for cluster-based social event detection as it enables the quantification of information shared between two clusters. Evaluating large-scale clustering approaches using entropy is uncommon since it works best in small clusters.

Table 2.5: Description of the evaluation measures used to assess the effectiveness of multimedia event detection methods.

Evaluation metric	**Definition**
Precision	It is computed as the ratio of the true positives (events correctly assigned to a cluster/class) over the sum of true positives and false positives (events incorrectly assigned to a cluster/class).

[13] http://www.multimediaeval.org/mediaeval2013/sed2013/
[14] http://www.multimediaeval.org/mediaeval2014/sed2014/
[15] https://dl.acm.org/doi/pdf/10.1145/2557642.2563674
[16] https://crisislex.org/data-collections.html#CrisisLexT26

Recall	It measures the percentage of correctly clustered (or classified) events out of all the events that should have been assigned to the cluster or class.
F-score	It combines precision and recall into a single value.
NMI	It measures the mutual information between the clustering results and the true class assignments of the events.
B-cubed	It estimates the precision and recall for every multimedia object (or social event) in the clustering results.
Entropy	It measures the degree of randomness in the clustering results.
Accuracy	It measures the proportion of correct event detections made by the model.

2.5.2 Event Relationship Identification

This section presents explanations for each semantic relationship between events, the nature and categories of event relationship, and the machine learning procedures employed in extracting event relationship. This section also introduces the use of event relationships in knowledge graphs.

2.5.2.1 What Is Event Relationship

Social media platforms have become a rich source of information about events. Understanding the dimensions associated with each event is a fundamental requirement for deploying automated techniques to identify semantic connections between events. In social media data, concepts and their relationships are often expressed in dimensions. Dimensions are used not only for expressing concepts and relationships, but also for detecting, representing, and reasoning multimedia events. Previous research in social media platforms has mainly focused on identifying events, but lately, attention has been shifting towards identifying relationships between events. For example, works in [**9, 37**, **23**] have introduced methods to define and identify relationships between events. In these methods, event relations were defined as meaningful connections between events that can be either temporal, spatial, or semantic. On the other hand, works in [**23, 56, 57**] defined event relationship as a relationship between two events $(\mathcal{E}_1, \mathcal{E}_2)$, where the first event $(\mathcal{E}_1)$ triggers the second event $(\mathcal{E}_2)$. Such kind of relationships can occur at dimension expressions- between phrases, clauses, and sentences, as well as between spatial and temporal features. The analysis of event relationships can be done at the text level (*e.g.*, *a textual dimension that expresses the meaning*) or at the spatial level focusing on point, line, or polygon information, or at the temporal level focusing on time information [**58**]. In the next section, we present the event relationship types.

2.5.2.2 Basic Types of Event Relationships

There are many different event relationships, but here we only present the three most important event relationships that fall into one of the following categories: *Temporal, spatial, and semantic.* These different forms of event relationships can have different natures and within each of these relationships there are different subtypes of relationships. The following sections describe the three main event relationships and their subtypes of relationships.

Temporal Relationship: is a type of event relationship, in which one or more events have temporal relationship with other events. Such relationship is often identified based on temporal dimension. Temporal dimension comes in different forms: *point-based, duration-based, and temporal coverage representative point-based.* Two recent studies [**9, 37**] used such timestamps to identify temporal relationships between events. In the first study, the system developed by the authors detected two distinct types of temporal relationships: *before and after.* In contrast, the second study utilized Allen's interval algebra to identify a range of different temporal relationships, which are represented as $R\left(\mathcal{E}_i, \mathcal{E}_j\right) = (T)$, where T represents 13 temporal relationships ($T = \{b,\ bi,\ s,\ si,\ m,\ mi,\ o,\ oi,\ f,\ fi,\ eq,\ d, \&\ di\ \}$), as shown in Table 2.6. Identifying these types of temporal relationships can play an essential role in providing knowledge graph, which is essential for retrieving multimedia events.

Table 2.6: Types of temporal relationships and their symbol

T-relationship	**Notation**
Before	*B*
After	*Bi*
Starts	*S*
Started-by	*Si*
Meets	*M*
Met-by	*Mi*
Overlaps	*O*
Overlapped-by	*Oi*

Finishes	*F*
Finished-by	*Fi*
Equals	*Eq*
During	*D*
Contains	*Di*

Spatial Relationship: We turn now to spatial relationships between multimedia events. Within multimedia events, particularly from their metadata, there are features that spatially relate to each other. These features can be represented as a point, line, or polygon. These representations can then be utilized to establish three types of spatial relationships (e.g., topological, directional, and distance) between events as described in [**37**]. Each of these spatial relationships also has several subtypes, which are detailed in Table 2.7. Spatial information is useful in many applications [**60**] including geographical information systems, spatial databases, spatial arrangement, planning, and others. It can also be used to prepare spatial queries.

Table 2.7: Basic types of spatial relationships

Types of Spatial relationships	**Subtypes of relationships**	**Description**
Topological	Equal	An equal geospatial relationship between events ($\mathcal{E}_i$, $\mathcal{E}_j$) be true, when their boundary and interior are identical.
	Disjoint	A disjoint geospatial relationship between events ($\mathcal{E}_i$, $\mathcal{E}_j$) be true, when $\mathcal{E}_1$ shares nothing with $\mathcal{E}_2$.
	Meet	Two events ($\mathcal{E}_1$, $\mathcal{E}_2$) can be connected by meet geospatial relationship, if they meet in space at some point/points.
	Within	Within relationship occurs when $\mathcal{E}_2$ spatial point completely encloses by $\mathcal{E}_1$.
	Overlaps	Overlaps relationship between two events ($\mathcal{E}_1$, $\mathcal{E}_2$) occurs when $\mathcal{E}_1$ and $\mathcal{E}_2$ have multiple points in common.
Directional	North	$\mathcal{E}_i$ is north of $\mathcal{E}_j$ or $\mathcal{E}_j$ is south of $\mathcal{E}_i$
	South	$\mathcal{E}_i$ is south of $\mathcal{E}_j$ or $\mathcal{E}_j$ is north of $\mathcal{E}_i$
	East	$\mathcal{E}_i$ is east of $\mathcal{E}_j$ or $\mathcal{E}_j$ is west of $\mathcal{E}_i$
	West	$\mathcal{E}_i$ is west of $\mathcal{E}_j$ or $\mathcal{E}_j$ is east of $\mathcal{E}_i$
Distance-based	Far Near	Any pairs of events($\mathcal{E}_i$, $\mathcal{E}_j$) whose computed distance values greater than the threshold value is considered to be "far." Otherwise, considered to be "near".

Semantic Relationship: In addition to the temporal and spatial relationships, scholars have explored different types of semantic connections between two events. As an example we can take the work in [**37**], which introduced and categorized semantic event relationships into memberships (*e.g., same-friends, same-club, and others*), kinships (*e.g., mother-of*), and causality (*e.g., due to, because of, related to, etc.*). Furthermore, [**23**] examined social events in tweets that are linked by causality. This is illustrated as follows:

$$C_r(\mathcal{E}_i,\ \mathcal{E}_j) = \begin{cases} Direct\ Causal & if\ \mathcal{E}_i \text{ causes } \mathcal{E}_j \\ Non-causal & Otherwise \end{cases} \tag{2.10}$$

where, $C_r(\mathcal{E}_i,\ \mathcal{E}_j)$ is a function that takes events $(\mathcal{E}_i,\ \mathcal{E}_j)$ as input and produces either *"Direct causal "or "noncausal."* The dimensions and machine learning approaches used for identifying the aforementioned types of relationships are presented in the next section.

2.5.2.3 Event Relationship Identification Approaches

State-of-the-art approaches for identifying event relationships typically rely on two machine learning techniques: 1) supervised and 2) unsupervised. A supervised algorithm utilizes labeled data that includes both events and their relationships to create a model that can accurately recognize connections between events. On the other hand, an unsupervised algorithm determines relationships between events by analyzing similarities in dimensions such as temporal, spatial, and semantic and by using patterns and rules. For instance, [**9**] presents an unsupervised machine learning approach to identify various directional, metric, and topological relationships.

A comparable approach is introduced in [**37**], where the authors identify different types of temporal, spatial, and semantic event relationships. Additionally, supervised deep learning methods have also been used to identify event relationships. For instance, the authors of [**56**] introduced a deep learning method to detect a causal relationship between events. Moreover, [**23**, **57**] introduce deep learning methods that use event context word extension method and an external causal-related knowledge base extracted from rich text sources (*e.g., news articles*) to train neural network model to identify causal relationships in tweets. Similar to the work in [**56**], these two models did not take into account spatial and temporal relationships.

2.6 Knowledge Graph

This section provides the contextual details regarding knowledge graph definition, knowledge graph model, knowledge graph database, and knowledge graph query language as well as approaches and applications widely used in knowledge graph.

2.6.1 Definition

Knowledge graph is a repository of real-world entities/events/concepts coming from several sources and their relationships [**63**]. All of these interlinked entities, events, or concepts can be described using graph topology [**61**], which refers to the arrangement of nodes and edges within the knowledge graph. In order to gain a better understanding of the knowledge graph, it is important to define some key terms in the context of this graph. Presented below are a few fundamental terminologies we frequently used:

- **Knowledge:** is a collection of facts that can be used to answer queries, to make predication, or to support decision-making. In other words, knowledge consists of a collection of information (data + meaning) that can be used to solve problems.
- **Graph:** In computer science, it is powerful data structure comprising a group of vertices (or nodes and edges. Graphs can be viewed in three basic forms: *undirected, directed, and weighted graph*. This study adopts the directed graph, in which each edge will be uni-directional.
- **Node:** is a distinct point within the knowledge graph that symbolizes actual entity/event.
- **Edge:** is a link used to connect nodes within the knowledge graph. Edges in the graph sometimes contain weights that indicate the strength of each link between the nodes.

2.6.2 Knowledge Graph Construction Approaches

There are two approaches for constructing knowledge graph [**25**]: bottom-up and top-down. A top-down approach typically starts by creating a knowledge graph and then populating it with data. This approach may be more suitable if the intended knowledge graph is properly defined. On the other hand, a bottom-up approach starts by collecting data and uses it to build a knowledge graph. If there is no predefined structure for the knowledge representation (or if there is a large amount of heterogeneous information that needs to be integrated in the knowledge graph), this approach may be more suitable. An example of a bottom-up approach is building a knowledge graph for real-word occurrences in social media platforms. In this instance, the initial step involves gathering and analyzing social networking records to recognize occurrences and their associations. Subsequently, the event knowledge graph can be established utilizing these findings. This study follows a bottom-up approach to build an event knowledge graph, which encompasses the collection of social media data, detection of events, and identification of event relationships.

2.6.3 Knowledge Graph Database

Knowledge graph database is a software package that uses graph structure to store and query data that is interconnected by pre-defined relationships. For the representation of data, it uses nodes, edges and properties. This helps to retrieve data easily. There are several knowledge graph database, including RDF, Neo4j, AllegroGraph, DGraph, Nebula Graph, TigerGraph, and others. This study adopts Neo4j as described in Chapter 7.

2.6.4 Graph Query Language

After storing all necessarily data into the graph database, Graph Query Language (GQL) is required to query and manipulate graph data that have been modeled as knowledge graph [**62**]. In other words, GQL is used to retrieve information from the knowledge graph. A summary of graph databases and their corresponding query languages are provided in Table 2.8.

Table 2.8: Knowledge graph databases and their corresponding query languages.

Graph database	Graph query languages
Neo4j	Cypher
Microsoft Azure Cosmos DB	Gremlin
JanusGraph	Gremlin and SPARQL
Titan	Gremlin
OrientDB	Orient DB SQL
ArangoDB	AQL
RDF	SPARQL
Apache TinkerPop	Gremlin
Amazon Neptune	Gremlin and SPARQL

2.6.5 The Integration of Knowledge Graph and Information Retrieval

This section provides an overview on the integration between knowledge graph and information retrieval. Over the past few years, there has been a great interest in merging knowledge graphs (KG) and information retrieval (IR). The primary reason behind this is the advantages offered by knowledge graphs. For example, one of the main advantages of KG is that it provides a structured representation of concepts (or entities). This helps users to find the summary information they need quickly. The second benefit is that KG has ability to understand the semantic aspects of the user query so that search results are accurate and useful. In the event domain, by storing knowledge about events and their relationships in knowledge graph, we can provide more contextually relevant results. For example, searching for "*music festivals in Addis Ababa*" may return irrelevant results when only the keywords are considered. However, performing a search against a knowledge graph that includes information like semantic, spatial, temporal, and participant as well as semantic relationships between events, will provide more accurate and relevant results. Overall, using a knowledge graph for information retrieval helps in comprehending user queries, documents, and facilitates semantic search by offering explanations of entities (events) and their interconnections [**63**].

2.8 Summary

In this chapter, we presented a summary of various topics, such as social media, social media ecosystem, multimedia data, and multimedia event. We also discussed how event-related tasks can be utilized for construction and completion of EKG. Specifically, we have covered social media documents crawling, social media document representation, multimedia event detection, and identification of semantic relationships between events. Additionally, we introduced an overview on the topic of knowledge graph, which includes its model, database, and query language, as well as commonly used techniques and applications. Finally, we closed the chapter by summarizing the benefits of KG in the context of information retrieval system.

CHAPTER 3
RELATED WORK

This Chapter briefly describes the most relevant related works dedicated to detecting multimedia events, identifying event relationships, and constructing an event knowledge graph for event retrieval within the context of Social Media Ecosystem. There has been a few focuses on constructing event-centric knowledge graphs, with efforts made to detect, link, and represent social events.

3.1 Multimedia Event Detection Methods

Multimedia event detection pertains to the process of detecting significant occurrences from a large volume of social media data. It involves automatic extraction of multimedia events that are interest from various sources and then categorized them into relevant groups based on their dimensions (or features) similarities. The primary goal of multimedia event detection is to provide users with a detail summary of what is happening around them. There are various approaches to multimedia event detection, which may be classified into three types: ***1)*** supervised (*classification*); ***2)*** unsupervised (*clustering*); and ***3)*** hybrid (*supervised +unsupervised*) as shown in Figure 3.1.

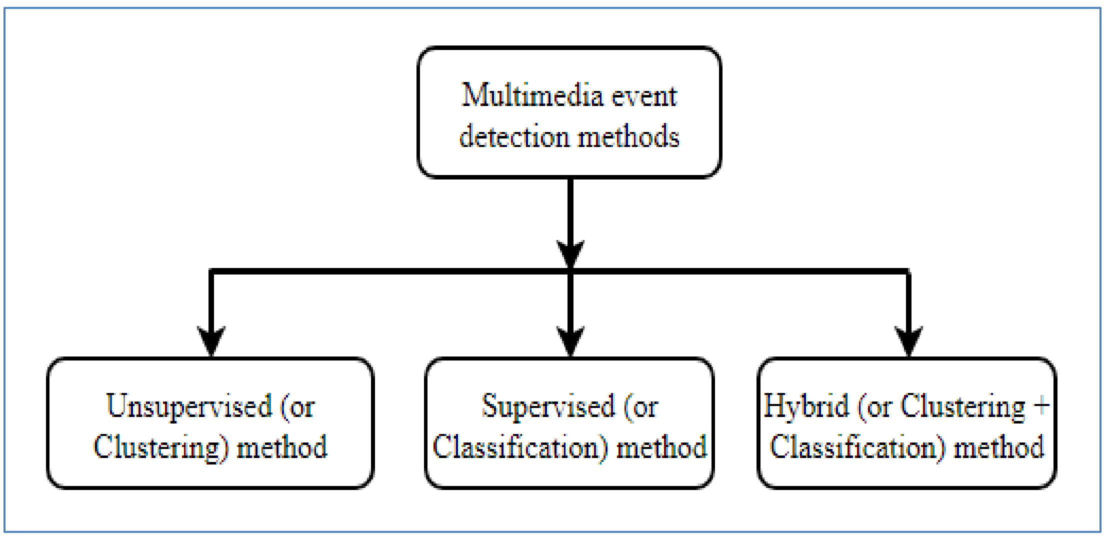


Figure 3.1: Machine learning methods used in multimedia event detection.

Machine learning techniques can be employed to identify multimedia events via two means: supervised machine learning and unsupervised machine learning. In supervised technique, the model is trained using labeled data to detect and classify social events into predetermined categories (*e.g., "sport event" and "non-sport event"*). On the other hand, an unsupervised technique uses unlabeled data to cluster similar events based on their dimensional similarities. Hybrid methods use a combination of these two techniques. For each of them, more details descriptions are presented below.

3.2.1 Unsupervised Methods

Various unsupervised or cluster-based solutions (*e.g.,* ***10, 40, 9, 11, 35***) have been introduced to detect multimedia events from social media data. These methods use different kinds of dimensions (*e.g., textual features, spatial, and temporal*) and different kinds of similarity functions (*e.g., see Section, 2.6.1.4*) to detect and group similar events into clusters. For instance, Li et al. [**10**] proposed a semantic class-based event detection algorithm to detect and cluster events from Twitter. The semantic classes include the Who, Where, When, and What aspects of an event. The authors utilized word embedding technique to learn the semantic features of tweets, instead of adopting an event detection approach that relies on lexical database (*e.g., WordNet [**53**]).* The authors concluded that the approach with a lexical database is not suited to detect events for two reasons: *(i)* tweets include many abbreviations and terms that are not in the lexicon; and *(ii)* a lexical database is usually not updated hence it doesn't reflect the dynamic features of social media documents. Finally, class-wise similarity measures are used for comparing and clustering events.

The authors of [**40**] proposed a 4W-aware method called BEHIND to detect, cluster, and summarize events from Twitter. Extracted bursty elements include 4W dimensions of events (*When, Where, Who, and What*) were first used to filter tweets related to the event. Then, they clustered detected events using a heterogeneous information network (HIN). The authors of [**11**] proposed a wavelet-base algorithm for *(i)* detecting multi-scale events from Twitter, and *(ii)* clustering events after computing similarities between events using spatial (Where), temporal (*When*), and textual description (*What*). Similar to the works in [**10, 40**], this approach uses tweets to detect events. To detect events in the photo and video collections on social media sites, the authors of [**35**] proposed a feature-centric approach to social event detection. They clustered photos/videos concerning events using temporal, spatial, social (*e.g., photo creator name*), and topical features. A similar method is proposed in [**33**], focusing on a clustering algorithm for detecting personal events from photos shared in social media ecosystem. In addition to dimension similarity measure, they used a set of rules defined based on event features (*Who, When, and Where*) to refine cluster results.

Considering social media platforms jointly (*e.g., Flickr, YouTube, and Twitter*), Abebe et al. [**9**] proposed a framework for detecting, describing, and linking social events. They used a graph-based agglomerative group average-link clustering algorithm to detect events from social media documents. Their model operates on three dimensions of events (*such as Temporal (When), Spatial (Where), and Semantic (What)*). This method utilized aggregation similarity measure to compute the similarity between two multimedia objects. In the context of event detection from image collection on Flicker, the authors of [**64**] detected events using metadata (*or user-generated time and location attributes*) associated with Flickr photos. In this study, photos are clustered based on the time and location dimensions. [**54**] presents another unsupervised approach for clustering social events from Flickr. The authors used three different clustering approaches: *(i)* temporal-based clustering; *(ii)* spatial-based clustering; and *(iii)* text-based clustering. They also used the temporal and spatial properties jointly to cluster social events.

Similarly, a study in [**65**] presented an unsupervised approach to event detection. This model used location, time, and participant dimensions to detect events, but ignored other semantic dimensions, such as What, Why, and How. Considering known and unknown events on social media platforms, Psallidas et al. [**49**] proposed an approach that identifies both known and unknown events. Initially, the authors used two-stage query formulation method using the rich features of social media documents to retrieve only documents related to known events from social media sites. To identify unknown events, features of unknown events, such as "*bursty vocabularies*" are used. Once both known and unknown events are identified, the authors utilize online clustering method in which cosine similarity for textual feature and the Haversine distance [**51**] for spatial feature were used for computing and clustering events. Finally, the approach is evaluated based on bursty vocabulary per time window and obtained high-quality clustering results. The abovementioned unsupervised event detection methods are summarized in Table 3.1.

Table 3.1: Unsupervised event detection methods.

Paper	Method	The technique used	Target data type	Features used in event detection method
Chen et al. [**64**]	Clustering	Retrospective detection (*feature-pivot technique*)	Photo	Temporal, & spatial
Li et al. [**54**]	Clustering	Semantic class-based clustering + class-wise similarity technique	Tweet	Participant, temporal, spatial, & textual features
Zeng [**40**]	Clustering	Heterogeneous information network (HIN) + similarity measure	Tweet	Temporal, spatial, participant, & textual features
Dong et al.[**11**]	Clustering	Wavelet-base algorithm	Tweet	Temporal, spatial, & textual features
Mansour et al. [**35**]	Clustering	Formal concept analysis (FCA)	Photo & video	Temporal, spatial, social (*e.g., photo creator name*), & topic features
Raad et al. [**33**]	Clustering	Rule refinement technique	Photo	Temporal, spatial, & participant
Abebe et al.[**9**]	Clustering	Agglomerative average-link clustering algorithm + Aggregate similarity measure	Photo, video, & tweet	Temporal, spatial, & semantic
Zaharieva et al. [**54**]	Clustering	Adaptive clustering technique	Photo	Temporal, spatial, & textual features
Liu et al. [**65**]	Clustering	Threshold & similarly measure techniques	Photo	Temporal, spatial, & participant
Psallidas et al. [**49**]	Clustering	Threshold, weight, & similarly measure techniques	Photo, video, & tweet	Textual features, temporal, & spatial

3.2.2 Supervised Methods

Supervised (or classification) method is a type of machine learning algorithm where the machine is trained on labeled dataset. In labeled dataset, each data point has a known label or category, such as *"event" or "non-event"*. The algorithm learns the pattern or the measurable characteristics of the data point. Once trained, the algorithm can then be used to detect and classify unlabeled multimedia data by predicting their label based on the patterns it has learned from the labeled data. Following this procedure, several classification methods have been used for detecting and classifying multimedia events. For example, the authors of [**15**] propose an approach to classify social media documents into a growing and evolving set of events. The approach consists of four stages: *(i)* retrieving candidate events using 6 queries related to event capture /upload time, location, tags, and description; *(ii)* training the SVM model using positive and negative examples; *(iii)* measuring the similarity between documents over different combinations of features associated with each document; and *(iv)* detecting event-related documents to classify them into pre-classified groups using pre-trained SVM[17] model.

The authors of [**38**] introduced supervised learning method for detecting common personal life events shared on Twitter. They used three classification-based event detection methods: *(i)* unigram model (UNI)-based event detection, which uses four distinct classifiers (*such as Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Multinomial Naïve Bayes (MNB)*) to detect personal event tweets and classify them into pre-classified groups using word frequencies as features. All classifiers are trained using 10-fold-cross-validation strategies and their results are reported in terms of precision, recall, and F-score; *(ii)* lexical features-based event detection (*UNI+META*), in which the authors use only an SVM classifier, considering contextual lexical features of tweets (*e.g., co-occurrence words, social relation terms, and temporal terms*) to detect events and classify them; and *(iii)* the third model of this work can detect events based on interactions features in addition to the lexical features.

The authors of [**39**] proposed a framework to detect real-time traffic events from social media. First, Twitter API is used to collect real-time tweets and then traffic-related tweets were filtered. Using this filtered data, events have been classified into positive, negative, and neutral classes using a trained SVM classifier. Similar to the work in [**15**], this task does not consider the semantic meaning of textual features and other important event dimensions (*Cause (Why) and method (How)*) in the event detection process. Moreover, the authors of [**66**] proposed a novel topic model to identify and classify social events from social media documents using boosted multi-modal supervised Latent Dirichlet Allocation (BMM-SLDA). This method consists of three steps: (1) representing social media documents (including images and texts) using bag-of-words representation; (2) training a classifier using the training data to iteratively obtain a set of social event classifier models; and (3) Choosing and applying a more accurate social event classifier only to predict labels for new event documents. But, the authors do not consider the semantic aspects

[17] SVM is a supervised learning method used for classification

of the textual features associated with each multimedia object. Each of these supervised methods is summarized in Table 3.2.

Table 3.2: Supervised event detection methods.

Paper	Method	The technique used	Target data type	Features used in event detection method
Reuter et al. [**15**]	Classification	SVM	Photo	Temporal, spatial, , & textual features
Choudhury et al. [**38**]	Classification	Naïve Bayes (NB), SVM, Decision Tree (DT), & Multinomial Naïve Bayes (MNB))	Tweet	Co-occurrence words, social relation terms, temporal terms, & interaction features
Salas et al. [**39**]	Classification	SVM	Tweet	Temporal terms
Qian et al. [**66**]	Classification	Boosted multi-modal supervised Latent Dirichlet Allocation (BMM-SLDA)	Photo & text	Textual features(e.g., topic)

3.2.3 Hybrid Methods

A hybrid event detection method is a way to combine the above two methods to achieve better results that are not possible with a single machine learning approach. Such kinds of solutions have been used to detect social events from social media ecosystem. For instance, Becker et al. [**36**] presented a hybrid method that combines an incremental clustering method with a Support Vector Machine (SVM) classifier. Their goal is to introduce a general framework that can detect events in social media documents. Initially, the authors used incremental clustering method to group candidate events. After the clusters were formed, they used an SVM classifier to separate *"event" and "non-event"*. Finally, the proposed method was evaluated using a dataset consisting of over 2.6 million tweets. But, this method doesn't take into consideration the semantic aspect of the textual feature.

An improved hybrid-based event detection algorithm supported multi-feature similarity learning technique is presented in [**41**]. This approach is most similar to the work of [**36**]. However, the method proposed from [**41**] has some fundamental differences from [**36**]. For instance, the ensemble similarity learning strategy is one of them. More specifically, the proposed method consists of two steps: *(i)* representing metadata information attached to each object, such as textual, spatial, and temporal features; and *(ii)* using both ensemble and classification-based similarity learning strategies in conjunction with an incremental clustering algorithm to identify and cluster social events. Moreover, two large-scale datasets (*i.e., Upcoming and Last.fm*) are used to train and test the proposed method. Similar to the work in [**8**], this method does not take into account the semantic (What) aspect of the textual features.

Another hybrid method is presented in [**42**], aiming to address the two challenges identified in SED 2013: *clustering and classification methods*. For event clustering, a novel data structure technique, called UT-image (or user-time image) is first used to store the metadata associated with images. In this data structure, every row of an image contains all data related to one user. Once the whole metadata turned into a UT-image, the clustering method was performed based on the similarity of the temporal, spatial, and textual features. Then, the classifier, named Support Vector Machine (SVM) was trained with "*event*" and "*non-event*" training sample sets and used for classifying events.

In [**67**], the authors proposed an integrated framework for identifying events on Twitter. The framework consists of five steps: data collection, preprocessing, classification, clustering, and summarization. After data collecting and preprocessing, the authors use three pre-existing classification methods (*i.e., Naïve Bayes (NB), Support Vector Machine (SVM), and Logistic regression (LR)*) to distinguish events from noisy tweets. The output of the classifiers is then utilized as the input to the clustering method. Targeted events are clustered based on spatial, temporal, and textual features. A hybrid method to detect emergency events is presented in [**68**]. This approach mainly performed three basic tasks: (i) classifying posts into "emergency" vs. "non-emergency" using two-step classification; (ii) extracting time and location information from emergency-related posts; and (iii) clustering emergency events according to their 3W dimensions similarity (*What, When, and Where*). Additionally, the authors of [**69**] proposed a hybrid learning method to detect spatio-temporal social events. They introduced a scalable architecture, namely Deep-Eware, which consists of two models: (i) supervised deep learning model used for features extraction and topic classification; and (ii) unsupervised spatio-temporal clustering model used for grouping events in terms of time and location. The target events are classified using CNN and bidirectional LSTM. The abovementioned hybrid social event detection methods are summarized in Table 3.3.

Table 3.3: Hybrid event detection methods.

Paper	Method	The techniques used	Target data type	Features used in event detection method
Becker et al. [**36**]	Hybrid	Incremental clustering + SVM classifier.	Tweet	Temporal, social (*such as retweets, replies, and mentions*), topical, & twitter-centric features
Becker et al. [**41**]	Hybrid	Incremental clustering + SVM classifier +.ensemble similarity learning technique	Photo	Temporal, spatial, & textual features
Nguyen et al.[**42**]	Hybrid	Watershed-based clustering + SVM classifier	Photo & video	Participant(author), temporal, spatial, & textual features
Alsaedi et al. [**67**]	Hybrid	Classifier (*i.e., Naïve Bayes (NB), Support Vector Machine (SVM), and Logistic regression (LR)*) + Online clustering	Tweet	Spatial, temporal, & textual features
Huang et al. [**68**]	Hybrid	Classifier (*i.e., KNN, Naïve Bayes (NB), Linear SVM (RBF), Decision Tree (DT), & Text-CNN*) + Dynamic clustering algorithm	Weibo posts	Temporal, spatial, & textual features

3.2.4 Discussion on Multimedia Event Detection

In conclusion, there are several approaches for detecting multimedia events. Among them, cluster-based multimedia event detection approach that considers temporal or spatial feature alone may not be sufficient to accurately detect and cluster events. The reason is that semantically different types of events (*e.g., political events, sport events, entertainment events, etc*.) can occur at the same time or location. Thus, utilizing clustering approaches that rely on individual feature (*e.g., temporal or spatial feature only*) to detect and cluster events may not be effective since the semantic feature is ignored. While combining spatial and temporal features in event detection may provide helpful information about when and where an event occurred, but it may not capture all semantically related information associated with multimedia contents. For example, if an algorithm primarily focuses on the spatial and temporal dimensions, it could overlook semantically relevant information, leading to the grouping of events with different meanings in the same cluster based on temporal or spatial similarities. To address these challenges, Abebe et al. [**9**] proposed an innovative approach that aggregates three event expressive dimensions: spatial (Where), temporal (When), and semantic (What). This technique effectively utilizes a semantic feature from social media data when detecting events. However, this study did not consider scalability issues, participant information (Who), and additional semantic dimensions (such as Why and Wow).

Many prior approaches to multimedia event detection have focused on a limited set of 5W1H aspects of event descriptors, such as spatial, temporal, and textual features. All of these approaches are limited in their ability to detect events that contain multiple features and heterogonous events. Additionally, many current approaches are not scalable and fail to handle vast amounts of multimedia data. These limitations highlight the need for new approach that can address these challenges and provide more accurate event detection results. For this, we developed a scalable event detection approach that builds up on [**9**] by including participant-related information (*Who*) and addressing scalability for detecting sufficient multimedia events from social media data.

3.3 Event Relationship Identification

An event relationship refers to the link that exists between two or more events, which can vary in type and nature. For example, two events may be related to each other in terms of time or location. To identify this relationship, a relationship extraction method is used, which is a subtask of information extraction that leverages machine learning methods like supervised, unsupervised, and semi-supervised learning [**70**]. In supervised machine learning, the algorithm for event relation identification is trained on a labeled dataset consisting of pairs of events and their corresponding relationship labels (e.g. temporal, spatial, and causal). During the training process, the algorithm learns from the provided labeled data to identify patterns and relationships between events. Once the model is trained, it can then be used to predict the relationship label for the given pair of events that have not been previously labeled. The accuracy of the model is dependent on the quality and size of the training data.

In contrast, an unsupervised event relationship identification algorithm compares dimensions over events, such as their temporal, spatial, and semantic characteristics, to identify relationships between events. This approach does not require a labeled dataset and instead relies on patterns within the data to identify different types of relationships between multimedia events. The machine learning techniques used for event relation identification are further discussed below.

3.3.1 Event Relationship Identification Methods

Very few event identification solutions have been developed on social media platforms. For example, the work in [**9**] proposed an unsupervised machine learning technique for detecting directional, metric, and topological relationships among events. The authors of [**33**] used the same technique to identify temporal, spatial, and semantic event relationships. Moreover, deep learning approaches have also been used for event identification in social media platforms. For instance, the authors of [**56, 23, 57**] developed a deep learning method to detect causal relationships between events. However, these three studies did not take into account any spatial or temporal relationships that may exist between events.

Table 3.4: Event relationship identification methods

Paper	Method	Relationship types	Dimensions used in event relationship identification methods
[**9**]	Unsupervised	Directional, metric, and topological	Temporal, spatial, and semantic features
[**33**]	Unsupervised	Temporal, spatial, and semantic	Temporal, spatial, and textual features
[**56**]	Deep learning	Causal	Textual feature
[**23**]	Deep learning	Causal	Textual feature
[**57**]	Deep learning	Causal	Textual feature

3.3.3 Discussion on Event Relationship Identification

In the literature, most current methods (e.g., **56, 23, 57**) for identifying event relationships from social media rely on supervised machine learning algorithms that extract causal relationship from tweets. However, event relationship methods that rely on grammatical structure patterns at the sentence level are often ineffective in social media documents because of their noisy, short, incomplete, and poorly structured nature. As far as we know, existing methods do not take into account relationships that involve more than two nodes (events) and do not consider events (nodes) that may be linked by different types of relationships (e.g., [**33**, **9**). Additionally, most of the current approaches for identifying relationships do not encompass the extraction of the three primary relationship categories (e.g., semantic, spatial, and temporal), as well as each type of relation within these categories (*e.g., distance, topological, and directional relations under spatial category*). They focus on extracting one or two event relationships. To overcome these, we propose an event relationship extraction model based on hypergraph notation that is simple and useful for identifying the three major event relationships as well as each relation types under the three major categories.

3.4 Event Knowledge Graph

In this section, we start our discussion with popular examples of knowledge graphs. After that, we focus on how it develops and state-of-the-art in knowledge graph particularly in the context of social media data. Before define what an event knowledge graph is; first, let's take a look at some existing knowledge graphs. One of the most known examples of knowledge graphs is the Google Knowledge Graph [**71**], which was first released in May 2012 and is used to improve the efficiency of knowledge representation and its search engine results with information such as title, short summary of the topic (or definition), images of the person, place, or thing and snippets of additional key facts. It gets these facts from databases shared across the web like Wikipedia, Wikidata, Crunchbase, LinkedIn, and Freebase. Another example of a knowledge graph is DBpedia [**72**], which is an effort to structure knowledge from Wikipedia articles. It represents knowledge about millions of entities and their relationships in graph structure. In addition to the Google Knowledge Graph and DBpedia, there are many other examples of knowledge graphs, including YAGO [**73**], Wikidata [**74**], OpenCyc [**75**], Freebase [**76**], and Linked Open Data (LOD) [**77**]. All of these knowledge graphs focus on entities and their relationships, rather than events along with their various relationships. To overcome this, in recent years, EventKG [**78**] and Open Event Knowledge Graph (OEKG) [**79**] have been published as event-oriented knowledge graphs that aim to represent events in a structured manner. On the bases of these, Event Knowledge Graph (EKG) can be defined as a type of knowledge graph that uses a graph data model to represent, link, and retrieve knowledge about events [**80**]. Such knowledge graph consists of events and their relationships. In order to create an event knowledge graph, it is necessary to extract events and their relationships. Then after, we can represent them in the form nodes and edges in the knowledge graph.

In social media ecosystem, very few event-centric knowledge graphs have been created. For example, the work in [**81**] proposed a method for representing events and their connections as a knowledge graph. Their method consists of three stages: 1) collecting social media data using API; 2) discovering events and their relationships using ICA (or unsupervised learning algorithm); and 3) construction event-centric knowledge graph. But, the graph was not design to represent temporal, spatial and semantic relationships between events.

The work in [**82**] proposed a method for constructing earthquake knowledge graph. The method consists of 4 steps: *1)* identifying required entities using BiLSTM-CRF from earthquake text data; *2)* sorting the identified entities in pairs; *3)* extracting relationships between entities; and *4)* storing knowledge about earthquake using Neo4j database. The authors also use RDF for expressing the given knowledge as triples. But, the proposed solution does not consider different event relationships between events. In additions, the method does not show the impact of hypergraph formulation (includes hypernodes and hyperedges) in the process of constructing knowledge graph.

Moreover, in [**83**], the authors propose an approach to build event-oriented knowledge graph with temporal attribute to get structured representation of events. The process typically involves the following steps: 1) collecting news documents from Sina platform; 2) extracting events from collected new documents using supervised machine learning technique; 3) extracting temporal relationship between events using BERT pre-trained machine learning model; 4) constructing event ontology; and 5) finally, they build event-centric knowledge graph. Overall, this approach is useful for generating structured and coherent storyline events. However, the proposed solution did not incorporates others important event relationships, such as semantic and spatial (or directional, topological, and distance relations) in their graph. The authors also fail to take into account hypergraph notation, which enables us to represent events (nodes) that may be connected by two or more relationship types.

3.4.3 Discussion on Event Knowledge Graph

Findings in event-centric knowledge graphs highlight the following main observations:

- All previous approaches considered a pairwise graph representation approach in the graph construction process, which means that hypergraph formulation is not used in prior studies.
- All event-centric knowledge graph did not take in account semantic (or causal) and spatial (*or distance, topological, and directional*) relationships between nodes in their graphs.
- We also observe that researchers are primarily focused on event detection. Very little attention has been paid to create Event Knowledge Graph (EKG). Building a comprehensive EKG could be useful to get structured and relevant events.

3.5 Knowledge Graph as input for Event Retrieval

There is a strong connection between information retrieval (IR) and knowledge graph. Knowledge Graph (KG) can be thought of as a structured database that captures the relationships between entities. IR, on the other hand, is the process of obtaining relevant information based on user queries. KG assists information retrieval system to better understand complex queries and take into account the relevant relationships between entities. Generally, the use of KG allows IR to provide more accurate and relevant results without clicking on blue links provided by the search engine. Similar fashion has been used to facilitate event retrieval by representing events and their relationships in a structured way. For example, the work in [**13**] presents a knowledge retrieval system (or information retrieval interface) that presents information about events from event-centric knowledge graph. However, the authors do not account for the three main event relations categories (such as, spatial, temporal, and sematic) and many relation types under these categories. In addition, the authors failed to account for multiple types of relationships between two events and as a single relationship that may exist between two or more independent events, which cannot be represented by using a paired graph.

3.5.1 Discussion on Knowledge Graph as Input for Event Retrieval

Pairwise graph representation has been utilized for representing entities of events and their relationships in the form of nodes and edges, but it can't present relationship that can link more than two nodes at once. Additionally, two or more two relationship types may exist between only two events, or a single relationship may involve multiple independent events, which are challenging to illustrate using Pairwise graph model. It is critical to have a general graphical representation that can capture all aspects of an event. Hence, the objective of this research is to construct a comprehensive event knowledge graph that is 5W1H-aware and capable of representing all aspects of an event (Cf. Chapter 7).

3.6 Summary

In this chapter, we have presented the most related prior tasks and approaches used in each of these tasks. We also presented discussions for each tasks. Finally, we have noticed that there are still requirements for techniques that can effectively address the following concerns:

- **Scalability in Event Detection:** Machine learning methods have achieved impressive performance in event detection, but their scalability remains restricted. To overcome this obstacle, we employed an incremental clustering strategy in our study.
- **Error Correction in Event Detection Results**: Most of studies can detect candidate events, but not real-word events. In order to deal with this issue, we utilized a supervised machine learning method to distinguish events from non-events in every cluster. It is important to conduct a serious analysis of errors, particularly in clustering outcomes to establish a high-quality event knowledge graph.
- **Identification of Naturally Different Multiple Event Relationships**: Multimedia events are multi-dimensional in nature, resulting in a complex structure for every type of event relationship. Due to this complexity, current approaches for identifying event relationships concentrate on extracting only one or two types of relationships. The study was able to identify 7 different types of temporal relationships, 11 different types of spatial relationships, which have been categorized into three groups: distance, directional, and topological, and 1 semantic relationship.
- **Representation of Multi-Dimensional/Domain Event as a Knowledge Graph:** It is not possible to represent events that have multiple dimensions in a pairwise graph structure. Furthermore, two or more two relationship types may exist between only two events, or a single relationship may involve multiple nodes (or events), which are difficult to depict with a paired graph. To address this, we used hypergraph and property graph notations in our Event

Knowledge Graph, which enable the representation of more complex relationships and multiple dimensions.

- **Event Retrieval from Event Knowledge Graph:** Although there are many search engines that retrieve information related to events, they are still lacking in providing information about the essential details of an event. Additionally, the end results of the current search engines are ranked lists of pages, in which the user is expected to browse through some of them and filter out the relevant information of the target event. These can be solved by using an event-based knowledge graph together with information retrieval system. EKG allows information retrieval system to query and retrieve full view real-world events.

To summarize, so far, very few event knowledge graphs have been introduced from social media ecosystem. However, most solutions are 1) domain dependent; 2) do not consider different event relationships between nodes (*temporal, spatial, and semantic*); and 3) do not consider hyper graph and property graph notations. This study addresses these issues by providing a novel framework. The framework consists of six stages: 1) social media data crawling; 2) social media data representation; 3) social event detection; 4) semantic relationship identification; 5) Event KG construction; and 6) event search API. Chapter 4 presents the detail aspects of our framework.

CHAPTER 4
PROBLEM DEFINITIONS AND PROPOSED FRAMEWORK

In the first part of this chapter, we present basic definitions (*including multimedia object, event, event relationship, and event knowledge graph*) that are used throughout this dissertation. In the second part, we present the proposed framework, which demonstrates its major modules and functions in each module.

4.1 Preliminaries and Problem definitions

In this dissertation, our central goal is to investigate 5W1H approach to construct EKG in order to enhance event search results with semantic information gathered from various social media sources. EKG allows our event search API to understand the meaning behind search queries in order to provide relevant and more accurate results. To achieve this objective, we have outlined the explanations of technical terms employed in this manuscript in the subsequent sections.

4.1.1 Definition of Multimedia Object

Definition 4.1 (Multimedia Object (O)). refers to any kind of digital content that combines different media types (*e.g., text, video, image, etc*.) to provide an interactive/engaging experience for the user. This may include additional information (or metadata) about the object attaching to it. This metadata includes a variety of information. A multimedia object and its metadata can be represented as [**9**]:

$$O = (Oid, L, S, T, P) \tag{4.1}$$

where:

- **Oid**: unique id of the object;
- **L:** location of creation/publication of the object;
- **S**: title/name/description/tags of the object;
- **T**: date and time of creation/publication of the object and;
- **P**: publisher/creator of the object.

All of these features can be used to describe the content, but, they cannot be used to distinguish event-related multimedia objects from non-event multimedia objects. For instance, information such as the time and location of publication or the publisher of a multimedia object, as well as its textual description, are usually inadequate to determine if the object is related to an event or not. This is because the event time or location may vary from the event uploader time or location, as well as event creator differs from event uploader. Thus, there is a need for a new approach that can (1) identify common features (or dimensions) for both event and non-event multimedia objects (*such as textual features like titles, tags, and content descriptions*); (2) identify unique features for events only (*such as event capture time, event capture location, participates involved in event*); and (3) understand the semantic of textual feature (What) of multimedia object, which is the first requirements for identifying events from social media documents. Additionally, it needs clear event definition. Following this, we define an event as follows.

4.1.2 Definition of Event

An event can be defined as follows from social media perspective

Definition 4.2 (Event ($\mathcal{E}$)). is a concept that refers to an occurrence of social, natural, political, or other phenomena that take place at a specific time $(T_{\mathcal{E}})$ and location$(L_{\mathcal{E}})$ involving one or more participants$(P_{\mathcal{E}})$, and having textual descriptions that discusses the specific action$(S_{\mathcal{E}})$, reason$(R_{\mathcal{E}})$, method $(M_{\mathcal{E}})$ of the event. This definition can be represented as follows:

$$\mathcal{E} = (T_{\mathcal{E}}, L_{\mathcal{E}}, P_{\mathcal{E}}, S_{\mathcal{E}}, R_{\mathcal{E}}, M_{\mathcal{E}}) \tag{4.2}$$

where:

- $\boldsymbol{T_{\mathcal{E}}}$**:** represents temporal dimension describing ***When*** the event occurs;
- $\boldsymbol{L_{\mathcal{E}}}$**:** represents geospatial dimension describing ***Where*** the event occurred;
- $\boldsymbol{S_{\mathcal{E}}}$**:** symbolizes semantic feature describing **What** type of event occurred;
- $\boldsymbol{P_{\mathcal{E}}}$: represents participants ***Who*** involved in the event;
- $\boldsymbol{R_{\mathcal{E}}}$**:** represents the reason behind **Why** the event occurred
- $\boldsymbol{M_{\mathcal{E}}}$**:** represents the method (or **How**) through which the event occurred.

Note that this study focuses on four dimensions of the event, such as temporal, spatial, sematic, and participant.

4.1.3 Dimensions Definitions

In social media ecosystem, multimedia data generally refers to the combination of different forms of contents [**65**], such as image, voice, video, and text, with in a single post. Besides the actual content of the multimedia data, there is also metadata (or dimension) associated with multimedia object. This metadata (or dimension) includes information, such as date/time of creating/posting, location where the content was posted/created, participant information (*e.g., the user who posted/shared the content or individuals/organization who was part of the event*), and contextual textual features (*e.g., tags, title, and descriptions*) that can provide additional context to the multimedia content. By analyzing these dimensions, we can identify relevant multimedia events. However, metadata (or dimensions) often requires dimensions definitions and dimension representation techniques in order to effectively detect social events. This is due to the fact that dimensions typically take many different forms and can be highly complex (*e.g., line and polygon forms*), making it difficult to process and interpret the data. To address this, we first present four definitions to define spatial, temporal, participant, and textual dimensions. After that, we proposed a dimension representation technique that can transform each dimension into a format that is easily to analyze and interpret. Each dimension associated within a multimedia object is defined as follows:

Definition 5.1 (Participant (Who) Dimension ($\boldsymbol{P}_{\mathcal{E}}$)). is a term used to refers an actor who is involved in a particular event. In the context of social media data, participant dimension can refer to details about the user who posted or shared content. In other words, this dimension refers to the person who involved during the event took place or the person who interacted with the content through shares, likes, and comments. However, all of these types of participant dimensions cannot be used directly as features of events or to identify events. Among all these participant information, only information about participants who were involved during an event can be used as an event descriptive feature for identifying events. Therefore, in this study, participant information regarding individuals/organizations involved in a particular event was used. Such information is typically stated in textual descriptions and can be identified using Named-Entity-Recognition (NER) techniques.

Definition 5.2 (Temporal Dimension ($\boldsymbol{T}_{\mathcal{E}}$)). It refers to time/date data associated with the creation, posting, and sharing of an event (or object) on social media sites. A single multimedia object published on social media sites may contain multiple Timestamps, which could refer to the past, present, or future. Each timestamp can be represented as content created, content uploaded, and content shared timestamp. This study uses only content creation timestamp because it accurately indicates when the event took place. Moreover, in social media data, there are two types of temporal dimensions that are commonly used: 1) single point time, which represents a specific point in time (*such as timestamp associated with an image post*; and 2) duration based time, which refers to the length of time to capture an event/object (*or a timestamp associated with a video post*). Effectively analysis of the temporal dimension often requires dimension representation technique, which is discussed in Section 5.2.2.

Definition 5.3 (Spatial (Where) Dimension ($\boldsymbol{S}_{\mathcal{E}}$)). It refers to the geographical location associated with multimedia object (or event). Social media sites often use geotagging and location-based (or GPS) services to add location data (*e.g., latitude and longitude coordinates)* to a distinct objects (or event) at the time of creation. Formally, this can be represented as follows:

$$\boldsymbol{S}_{\mathcal{E}} = (\emptyset, \boldsymbol{\lambda}) \tag{5.1}$$

where,

- $\emptyset$: represents latitude, and
- $\boldsymbol{\lambda}$: represents longitude coordinates.

Definition 5.4 (Semantic (What) Dimension ($\boldsymbol{S}_{\mathcal{E}}$)). It refers to the meaning (or context) surrounding textual features, which consists of tags, titles, and descriptions of multimedia objects or events provided by users. To understand the semantic aspect of these features (or the meaning of events), a semantic network was built based on WordNet [53]. This was accomplished in four stages: 1) indexing words in textual features; 2) extracting concepts; 3) identifying relationships between concepts (*e.g., synonym, type-of, part-of, etc.*); and 4) representing concepts and their relationships in a network structure. More specifically, our semantic network contains semantic classes called synsets, each of which refers to groups of words that are synonymous in meaning. For example, the synset "*concert*" is synonymous with words like performance, show, presentation, live music, and others. These words are all related to "*concert*" and can be used interchangeably in certain contexts. By interlinking each concept through semantic relationships and providing a brief description (or "gloss"), a more comprehensive understanding of the event can be gained.

4.1.4 Definition of Event Relationship

Definition 4.3 (Event Relationship (R)). It refers to the connection between different events. These connections can take various forms: *temporal, spatial, and semantic*. In other words, two events (e_1, e_2) can be linked using these three event relationship types. For this, we adopt definition in [**33**] and can be represented as follows:

$$R(e_1, e_2) = (Tem, Spa, Cau) \tag{4.3}$$

where,

- ***Tem:*** represents a group of temporal relationships
- ***Spa:*** symbolizes various types of spatial connections that exist between events. In our case, an event can be spatially related in three ways: *topological, directional, and distance-based (see Chapter 6 for more detail).*
- ***Cau:*** represents causality relationship between two events $(e_2,\ e_1)$.

4.1.5 Definition of Event Knowledge Graph

Definition 4.4 (Event Knowledge Graph ($\boldsymbol{EKG}$)). It defined as $EKG = (V, E)$, where V is a set of vertices $(u \subseteq V)$ and E is a set of direct edges $(e \subseteq E)$. This graph model effectively represents all aspects of events using four types of nodes (base node, master hypernode, hypernode, and simple node) and two types of links (hyperedge and simple edge). Table 4.1 presents descriptions regarding the components of EKG.

Table 4.1: Knowledge Graph terms and their definitions as used throughout this dissertation.

Term	Definition
Hypergraph	Hypergraph is a graph model that contains a set of nodes and hyperedges/edges, where a hyperedge can link any number of nodes.
Base Node	It is the highest node in the graph that corresponds to the entire event and serves as a root node that manages all the master hypernodes.
Master Hypernode	It is one of nodes in the graph that represent different event categories or domains, such as cultural, political, social, natural, and sport events.
Hypernode	The third layer in the graph and used to represent an event along with multimedia objects associated to it (such as photos/videos). For instance,

	photo/video of processions, photo/video of Ketera before Timkat, photo/video of dancing during Timkat, and photo/video of singing, are all related to the Timkat *(or Epiphany* ($\mathcal{E}_3$)*)* festival event, as shown in Figure 4.1 (. The inner nodes of the hypernode reflect individual events (e.g., Ketera*)* with their associated dimensions.
Simple Node	It was used for representing an event with a single image/video.
Edge	It is a connection between exactly two events.
Hyperedge	It is a connection between any numbers of events.
Graph database	It is a database model designed to store and manages graph data.
Graph query	It is a retrieval operation on a graph database using query languages, such as Cypher. Graph query enables users to (1) express their needs, (2) traverse through the graph data, and (3) retrieve a relevant event that matches specific criteria.

After describing each term, we construct the event graph, following definition 4.4 presented in Section 4.1.4. Our event graph was constructed based on six basic concepts: objects (all events), event types (domains), specific group of related events, simple event, features of events, and relationships between events. These concepts are sufficient to describe any real-world event and fit into a hierarchy of events, as the first layer starts at all events, all events into event types, event types into specific group of events, and specific group of events into individual events. Figure 4.1 demonstrates the figurative representation of our graph.

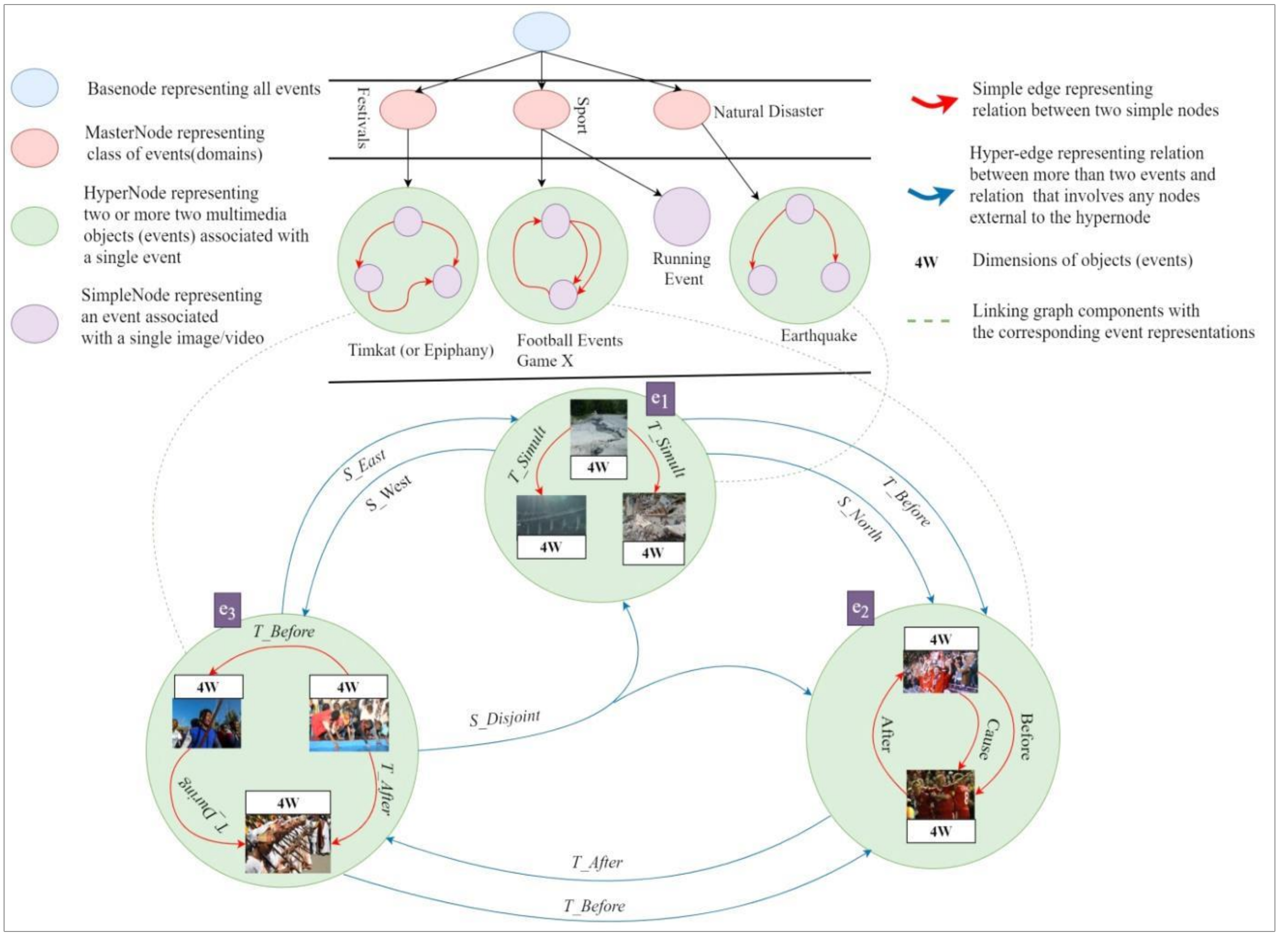


Figure 4.1: Conceptual visualization of event graph

To summarize, Figure 4.1 indicates three main observations: *(i)* the presented event hypergraph is a directed hypergraph, for example, $\mathcal{E}_1 \xrightarrow{Before} \mathcal{E}_2$ or $\mathcal{E}_1 \xleftarrow{After} \mathcal{E}_2$, *(ii)* in this graph, there are two types of links: *(1)* an ordinary edge (linking only two vertices) and *(2)* hyper-edge (linking more than two hypernodes or any nodes external to the hypergraph), and *(iii)* there are dissimilar types of hyper-edges between hypernodes (events): spatial, temporal, and causal.

4.2 Proposed Framework

In this work, we propose a new framework that incorporates the 4W elements of an event extracted from digital ecosystem. As illustrated in Figure 4.2, the framework contains six interacting components: 1) social media data crawling and preprocessing; 2) social media data representation; 3) social event detection; 4) semantic relationship identification; 5) Event KG construction; and 6) event search API. We here provide brief description about each component. Details on each component are provided in Chapter 5, 6, and 7.

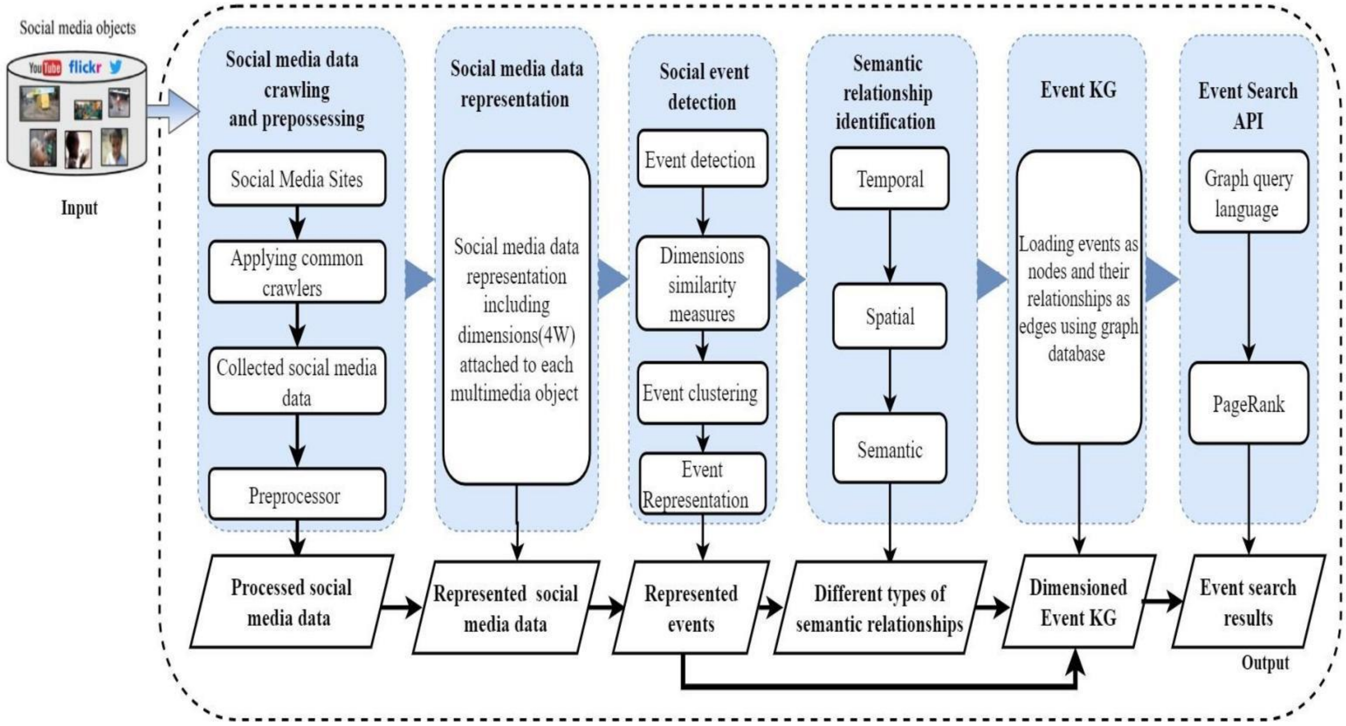


Figure 4.2: Proposed framework.

4.2.1 Social Media Data Crawling and Pre-processing

In this module, multimedia data can be obtained through Twitter, YouTube, and Flicker APIs. The outcome is a set of multimedia data with their respective dimensions. These data can present difficulties concerning heterogeneity in their structure and layout. To tackle this issue, researchers transformed all collected multimedia data, particularly metadata, into a consistent data format without changing their original form [**9**]. Furthermore, multimedia objects are typically accompanied by textual descriptions that may contain noise, ungrammatical text, irregular expressions, spatial symbols (*e.g., hashtags, Emojis, etc*.), abbreviations, and often consist of short sentences [**9**, **41**]. As a result, it is essential to perform text preprocessing tasks to improve the quality of the dataset. Text preprocessing phase involves activities like, tokenization, stop word removal, smoothing noise data, POS tagging, Named Entity Recognition (NER), and stemming. Smoothing noise data is an important step for removing irrelevant information (*e.g., HTML tags, emojis, punctuation marks, and hashtags*), while tokenization can help to breakdown textual descriptions into words (tokens). Stop word removal can eliminate meaningless words (*e.g., "the", "a", "an", and so on*) from textual descriptions, while part-of-speech (POS) tagger can help us to identify which words are pronouns, nouns, conjunction, verbs, adjectives, and so on. Named entity recognition is also important for identifying specific entities (*e.g., organizations, people, or places*) in the textual description. Furthermore, the process of stemming is carried out to minimize the words to their stem form, which can enhance the precision of event detection. Finally, dimensions selections (e.g., *spatial, temporal, semantic, and participant*) are performed.

4.2.2 Social Media Data Representation

This component of the framework mainly handles social media data representation in a way that is suitable for the event detection process. This is a challenging task as social media data is often unstructured and noisy as well as there are naturally different dimensions that require different data representation models. Before proceeding to the event detection task, dimensions for each multimedia object must be represented. For this purpose, the dimensions representation sub-component is responsible for identifying distinct-based dimension values (*e.g., image timestamp*), dimensions coverage values (*e.g., video timestamp, polygon or line-based spatial values*), and representative values for each dimension coverage. Afterward, only distinct and dimensions coverage representative values were used to measure the similarity between events efficiently during the event detection process. More information on individual dimension representation details can be found in Chapter 5.

4.2.3 Social Event Detection

It uses represented social media data to detect real-world events in a scalable manner (or the output of social media data representation component can be used as input for this component). In order to detect and cluster events, the event detection algorithm uses four different similarity measures (*e.g., Haversine distance, Jaccard similarity, Euclidean distance, and Semantic similarities*) and dimensions weighted strategies. Specifically, this component uses a detection method that considers the dimensions of the event. The first step of this method is to identify the dimensions associated with each multimedia object (or event), which includes temporal, spatial, semantic, and participant dimensions. Once the dimensions are identified, the authors assign weights to them according to their significance. These weighted dimensions are utilized to detect social events from social media documents. Our experimental results indicate that the weighted event detection method outperforms the unweighted event detection method as described in Chapter 5.

4.2.4 Semantic Relationship Identification

It is responsible for identifying semantic relationships between events by comparing dimensions over events. The results from detecting events can serve as input for this process. As each detected event possesses multiple characteristics, they can be interconnected based on factors such as cause, place, and time. To establish such interconnections, this component of the framework mainly used unsupervised machine learning algorithm to identify three main types of event relationships (*spatial, temporal and causal*) and their sub-relationships (cf. Chapter 6).

4.2.5 Event KG

This component of the framework uses graph database to store event detection outputs as nodes and relationship identification outputs as edges to construct an Event Knowledge Graph (EKG). In other words, it is responsible for constructing EKG by storing events and their relations in the graph database in the form of nodes and edges (cf. Chapter 7 for more details)

4.2.6 Event Search API

This component is responsible for providing an API for searching semantically rich multidimensional events from Event Knowledge Graph (EKG). The API allows users to search events based on textual, temporal, or spatial queries. More specifically, this component of the framework works by utilizing the event knowledge graph DB and search queries as inputs and produces pertinent events with a short overview that encompasses information related to spatial, temporal, and others.

4.3 Summary

In social media platforms, the majority of existing approaches concentrate on event detection, but there are still some unresolved issues, such as identifying semantic relationships among events from the perspective of social media. This is a crucial step in constructing a comprehensive Event Knowledge Graph (EKG) that can provide semantically rich and dimensioned events. To tackle these issues, we first introduce formal definitions used in this work and then proceed to explain key terms related to our event knowledge graph, including hypergraphs, nodes, edges, Knowledge graph databases, and Knowledge graph query languages. We also explore the role of knowledge graphs in event retrieval. Finally, we present our framework, which consists of functions from multimedia data collection to event retrieval using our event search engine.

CHAPTER 5

STAGE-II & III: SOCIAL MEDIA DOCUMENT REPRESENTATION AND EVENT DETECTION

As outlined in Chapter 4, our proposed framework consists of six essential components (or stages). Of these stages, social media document representation (Stage-II) and event detection (Stage-III) are the second and third components within the framework. Here, we explain how these stages are employed in event detection process. The Chapter also presents event detection algorithm, similarity measures, cluster comparison strategies, experimentation settings, evaluation dataset, and the results obtained in Stage-III. Finally, we illustrate how evaluation metrics presented in 5.4.1 are utilized to assess the effectiveness of our multimedia event detection.

5.1 Introduction

Today, social media sites are generating a vast amount of multimedia data, often in an unstructured form which contains various types of content (such as texts, pictures, videos, etc.) and having diverse standard formats that are produced by countless users on several digital platforms. Consequently, a number of difficulties are presented [**26**, **85**], which include: *1)* absence of standardization in data model to enable the representation of metadata (or dimensions) for each multimedia object; *2)* the difficulty of identifying multimedia events within social media data as it requires an understanding of the semantic of features attached to a multimedia object; and *3)* the challenge of achieving scalability while identifying events to manage the continuously growing nature of multimedia data. To address these difficulties, we propose an incremental clustering technique to detect multimedia events. The approach takes represented social media documents from various sources as input and generates clustered multimedia events as output. This process is achieved through a series of sub-steps within stage-II and stage-III, including metadata (dimensions) representation, similarity measures determination, cluster comparison strategy determination, event detection, and detected event representation. Each of these sub-steps involved in stage-II & III are described in the subsequent sections.

5.2 Social Media Document Representation

5.2.1 Dimensions Representation

In the previous Chapter, we discussed the definitions of the four dimensions of multimedia objects. To represent these dimensions, it is necessary to comprehend the essential components of social media documents. These documents have two parts: *1)* multimedia contents (audio, *text*, *image*, *video, and others*); and *2)* metadata (*i.e., 5W1H dimensions*). To identify multimedia events, accurately representing these dimensions is essential. To achieve this, we have proposed an approach that can extract and represent dimensions involving spatial (Where), temporal (When), semantic (What), and participants (Who). The following are details of these dimension representations.

Participant Dimension Representation: To accomplish this task, we first extract only the actors involved in the event. We did not take into account other categories of participants (*such as uploaders, users who posted/shared contents*) in this research since they do not aid in identifying actual events. Moreover, participant dimensions may differ according to event types and may consist of one or more actors. For participant(s) dimension, we represent values as persons or organizations.

Temporal Dimension Representation: For this, we represent temporal data associated with each objects as Unix timestamp. The Unix timestamp[18] is a way to represent a specific time as a number of seconds elapsed starts at the Unix Epoch on January 1st, 1970@00:00:00 UTC [**26**]. For example, the Unix timestamp for "*2010-04-24 12:49:47*" is "*1272102587*." It is a widely accepted representation technique for comparing time data. Our temporal representation was done in three steps. In the first step, single point temporal values (*e.g., date/time associated with an image post*) were identified. The second step was to identify duration-based times, which refer to particular time periods, e.g., a timestamp associated with a video post. These timestamp can be defined as temporal coverage. However, using temporal coverage values to compute the time difference between two events is, on the other hand, computationally expensive. To address this, we used the

---

[18] The Unix time stamp is a way of representing a specific point in time as the number of seconds that have elapsed since January 1, 1970 at midnight GMT (Greenwich Mean Time). This number is often used in computing systems for tracking and recording time, since it represents a universal, standardized reference point that is independent of time zones and other factors. The Unix time stamp is also referred to as the Epoch time or Unix Epoch time

method described in the paper [**9**], which enables a single-point timestamp for every temporal coverage**.** The representative point for temporal coverage can be derived by computing:

$$t_c(\boldsymbol{T}) = \frac{t_s + t_e}{2} \tag{5.2}$$

where,

- $\boldsymbol{t_c}(\boldsymbol{T})$: symbolizes temporal coverage representative point
- $\boldsymbol{t_s}$: symbolizes start timestamp in T
- $\boldsymbol{t_e}$: symbolizes end Timestamp in T

The final step was to convert all time data to Unix timestamps.

Spatial Dimension Representation: Social media dataset used in our study contains multimedia objects (or events) that took place in three distinct spatial dimensions: points, lines, and polygons. Each of these dimensions can be used for representing and analyzing geographic data related to the event. For instance, points can represent the location of a specific event, like a fire in a building. Lines can represent linear dimensions, such as an event that occurred over a road like a marathon. Polygons can represent areas of events, such as a football match. For representing these dimensions, our study followed a three-step process. Initially, we represented point-based spatial values as single coordinates to indicate the specific event location. Next, we determined spatial coverage for polygonal or linear multimedia objects/events. Finally, we identified the midpoint coordinates as representative point of spatial coverage. All identified representative points can be used to efficiently compare different multimedia events.

Semantic Dimension Representation: For this, we represent textual features as a semantic network based on WordNet [**53**] for measuring the semantic similarity between two concepts. The task of semantic dimension representation is done in three steps. In the first step, we identify concepts and their semantic coverages reflecting group of synonyms words/phrases expressing the same concept. In the second step, we create semantic network consisting of two main components (G= (V, E)), where V representing concepts nodes and E representing semantic relationship between nodes as edges [**87**]. In the second step, we determine semantic coverage representative point (or concept) from synsets in the sub-graph. By comparing concepts with each other in semantic coverage, a representative concept (point) can be selected. It is a single concept representing the concept having the maximum similarity score among the concepts in the same group. This enables us to effectively compare two multimedia objects using their semantic coverage representative concept (point), rather than comparing two sub-graphs of concepts.

To summarize, the detection of events requires an appropriate data model of dimension representation and often entails the utilization of multiple dimensions in combination to obtain precise results. In the succeeding subsection, we describe the event detection algorithm, which takes social media data that has been represented and generates events as its output.

5.3 Event Detection

Once social media data represented, the next task is event detection. The event detection task takes represented social media data as input and produces clustered real-world events as output. Often, the event detection process involves machine learning approaches, such as unsupervised (or clustering) and supervised (or classification) [**88**]. This study adopts clustering approach that can be used to identify and split events into several groups based on their dimensions. In the following paragraph, we provide details about our event detection method.

To identify events, we started by choosing a suitable and scalable approach from the sate-of-the-art in unsupervised (or clustering) algorithms. To accomplish this, we established some criteria based on our objectives and the type of data we have. The chosen clustering algorithm needed to fulfill the following criteria: 1) scalability to handle the dynamically growing multimedia data; (*2*) capability to handle noise data and dimension diversity, such as spatial, temporal, etc.; and *(3*) it should not require previous knowledge about the number of clusters [**89**]. Thus, clustering algorithms like K-mean Clustering [**90**], Self-Organizing Maps (SOM) [**91**], and Fuzzy C-means Clustering (FCM) [**92**] that require prior knowledge about the number of clusters are not well suited for the problem. Similarly, the Graph Partitioning Algorithm (SGPA) [**93**] is unsuitable because it becomes difficult, cost-ineffective, and memory-intensive when the graph/data size increases too much. Additionally, algorithm like Agglomerative Hierarchical Clustering [**94**] is unsuitable as it does not perform adequately on large heterogeneous data.

Based on the above criteria, we used a scalable clustering approach, named "*Incremental Clustering Algorithm*". The algorithm is simple, scalable and effective as it has ability to efficiently process/handle large amount of data by incrementally adding new objects or clusters to an existing data. This method is useful for scenarios where clustering does not require prior knowledge about the number of clusters (or needs to be updated dynamically). Here are the steps involved in our incremental clustering algorithm (Algorithm 1): Given, represented social media objects to cluster $(O_1, \dots, O_n)$, feature weighting (W), threshold (Δ), and similarity functions (σ). The algorithm considers each social media object (O_i) in sequence. Initially, it starts by producing a new cluster (C_1) for the first object (O_1), $C_1 = \{ O_1 \}$. Then, the algorithm gets the next object (O_i) and computes its similarity against with an object in the existing cluster (C_1) using their features (or dimensions). Then after, the relevance of a cluster to an object is determined based on the similarity measures and decision score, cluster decision score is either the average score across all objects in the cluster (average-link strategy), maximum decision score in each cluster (single-link strategy) or the highest minimum score in each cluster (complete-link strategy) [**41, 67**]. If there is a cluster whose similarity to O_i is greater than or equal to Δ, then O_i is assigned to the most relevant cluster. Otherwise, it creates a new cluster containing object (O_i). The iteration step repeats the above steps for each new data pint. Please note that the similarity between entering documents and documents in existing clusters can be determined by comparing their features (or dimensions). To summarize, our clustering algorithm (see Figure 5.1) generates a set of clusters containing real-world events.

```
Algorithm 1: Event Detection based on 5W1H aspect of events
Input:
1       Multimedia Objects: Collection        // Represented Multimedia Objects. these representations
  includes Semantic (e.g. title, description and tag), Spatial(geographical mid-point values), Temporal
  (time/date) and participants (person/organizations).

Variables:
2       Sim(): Decimal //similarity function
3       Threshold Values(Δ): Decimal //a pre-defined threshold Value
4       Clusters: Collection        // Cluster of Objects

Output:
5       Cluster of Events        //contain Detected Events

Begin:
6       Initialize empty set Ω of cluster
7       Initialize empty set decision score value of cluster Z
8           If O_1 be the first multimedia object Then
9               Putting object O_1 into the first cluster C_i = {O_1}
10          Else
11              for each object o ∈ O , do: // one by one
12                  Lookup the decision score value Z_i of each cluster C_i, ..., C_n // Decision score
   value computing using single-link, group average-link and complete-link techniques
13                  Sim(o, z_i) //Computing the similarity between the current object and
   decision score value for each cluster
14                      If Sim(o, z_i)≥ Δ, do :
15                          assigned a given object o to C_i
16                          Recomputed/Update the decision score value of z_i
17                          Go to Step 8 and Continue processing the input objects.
18                      End If
19              Next
20          End If
21          Next
22          If a given object o not assigned to any existing clusters Then
23              Create a new cluster C_new = {o} put C_new into Ω
24              Set its decision score value z_new = o and put z_new into Z
25              Return Events // Cluster of Detected Events
26      End
```

Figure 5.1: Pseudo code of multimedia event detection.

5.3.1 Similarity Measured Used in the Proposed Event Detection

Cluster-based event detection algorithm requires choosing two or more suitable similarity measures to compare and calculate similarity scores between multimedia objects based on their dimensions. Therefore, similarity metrics are an essential element utilized by the cluster-based event detection algorithm to cluster similar multimedia objects together [**95**]. The subsequent section explains the similarity metrics employed in event detection algorithm.

Participant (Who) Similarity: To identify groups of people (or individual) who are participating a specific event, we represent each participant (Who) as a set of dimensions. Then after, we can compute the Jaccard Coefficient similarity between pairs of participants (P_i, P_j). This can be defined as follows:

$$Sim_P(\, P_i \,, P_j) = \frac{|p_i \cap p_j|}{|p_i \cup p_j|} \tag{5.3}$$

The similarity score value ranges from 0 to 1, where a score close to 1 indicates that the two participants (p_i, p_j) almost identical. If the score near 0 indicates that p_i and p_j almost dissimilar.

Temporal (When) Similarity: For this task, we used time data represented as the number of seconds (*cf. Section 5.3.1*). Then after, we used Euclidian distance to measure the similarity between two multimedia events [**9**, **41**]. This is computed as [**9**]:

$$Sim_T(e_{T1}, e_{T2}) = \frac{1}{1+Dis_T(e_1,e_2)} \in [0,1] \tag{5.4}$$

where, $Dis_T(e_{T1}, e_{T2}) = |t_1 - t_2|$, t_1 is temporal value of e_1, and t_2 is temporal value of e_2. Note that we apply normalization function to scale each Euclidian distance to the [0, 1] range, meaning the similarity score value ranges from 0 to 1, where a score close to 1 indicates that the two objects $(O_i \,, O_j\,)$ are perfect identical, and 0 indicates no similarity at all between the objects (events).

Spatial (Where) Similarity: To accomplish spatial similarity, we used represented spatial data as noted in Section 5.2.2 and Haversine distance [**51**] to compute the distance between two locations of multimedia objects ($(Sim_L\,(\mathcal{L}_{1,}, \mathcal{L}_2))$ on Earth. The computation is as follows:

$$Dist_L(\, e_{L1}, e_{L2}) = 2 * r * \arcsin(\sqrt{sin^2\left(\frac{Lat2-Lat1}{2}\right) + \cos(Lat1) * \cos(Lat2) * sin^2\left(\frac{Lon2-Lon1}{2}\right)}\,) \tag{5.5}$$

where,

- **r** is the radius of Earth, typically measured in kilometers;

- **lat$_1$** and **lon$_1$** are the latitude and longitude of the first multimedia object;
- **lat$_2$** and **lon$_2$** are the latitude and longitude of the second multimedia object; and
- **arcsin** is the invers of the sine function.

Semantic (What) Similarity: Measuring semantic similarity is crucial in machine learning tasks such as event detection. This measure focuses on the meaning of textual features to establish the similarity between two multimedia objects. Syntactic similarity measures fail to capture the semantic proximity between two objects. To address this concern, the authors of [**9**, **52**, **96**, **97**] have proposed semantic similarity techniques that rely on WordNet [**53**] to compute the semantic similarity between two concepts. This process involves various steps, as outlined in Section 5.2.2, such as indexing, determining semantic/concept coverage, constructing a semantic network (or graph), and identifying a semantic coverage representative concept (point). Using only the semantic coverage representative points, we can compute the semantic similarity between two concepts. In a semantic network, measures of concept similarity can be classified into four groups: node-based, edge-based, gloss-based, and hybrid. They are described as follows:

- ***Node-based semantic similarity:*** is a technique that measures the similarity between two unique concepts based on their semantic meaning. Nevertheless, it concentrates only on concepts that are represented as nodes and does not consider a brief description given to a specific concept, which is called gloss. A gloss-based semantic similarity can overcome this limitation by incorporating descriptions as an additional factor in determining similarity.
- ***Gloss-based semantic similarity***: A gloss is a brief description given to a particular concept and can be used for three purposes: 1) to provide additional information about the meaning of a distinct concept; 2) to distinguish one word from other similar words; 3) to measure semantic similarity between two concepts.
- ***Edge-based semantic similarity:*** Unlike node-based and gloss-based techniques, edge-based semantic similarity measures the semantic similarity between two concepts by analyzing the links or edges connecting them.
- ***Hybrid semantic similarity:*** It integrates semantic similarity techniques based on node, edge, and gloss. In this semantic similarity metric, the semantic similarity between two concepts can be computed as follows:

$$Sim_s(C_1, C_2) = w_{Node} \times Sim_{Node}(c_1, c_2, KB) +$$

$$w_{edge} \times Sim_{Edge}(c_1, c_2, KB) + \quad (5.6)$$
$$w_{Gloss} \times Sim_{Gloss}(c_1, c_2, KB) \; \epsilon [0, 1]$$

where, C_1 *and* C_2 are represent the two concepts associated with multimedia objects (or events) and KB symbolizes the lexical knowledge base.

Events (Ɛ) **Similarity:** As we have seen from the similarity measures presented above, there is no single similarity measure that encompasses all kinds of dimensions in multimedia objects (or events). Consequently, the similarity between two multimedia objects/events ($Sim(e_1, e_2)$) can be computed by combining individual dimensions similarity measures. To accomplish this, there are ensemble functions, such as maximum, minimum, average, and weighted sum similarity measures. This study adopts the weighted sum similarity measure to compute the similarity between two different multimedia objects (or events). Mathematically can be represented as follows:

$$\begin{aligned} Sim(e_1, e_2) = \quad & w_T \times Sim_T(e_1, e_2) + \\ & w_P \times Sim_P(e_1, e_2) + \\ & w_L \times Sim_L(e_1, e_2) + \\ & w_S \times Sim_S(e_1, e_2) \end{aligned} \quad (5.7)$$

where,

- $\boldsymbol{e_1}$ ***and*** $\boldsymbol{e_2}$**:** two multimedia events (objects)
- $\boldsymbol{Sim_T}$, $\boldsymbol{Sim_P}$, $\boldsymbol{Sim_L}$ **&** $\boldsymbol{Sim_S}$**:** represent temporal, participant, geospatial, and semantic similarity measures, respectively.
- $\boldsymbol{w_p}$, $\boldsymbol{w_T}$, $\boldsymbol{w_L}$, **&** $\boldsymbol{w_S}$**:** represent the similarity weight values, which is set by the author of this work.

5.3.2 Cluster Comparison Strategies Used in Event Detection

This section explains our strategies for analyzing cluster outcomes, namely complete-link, single-link, and average-link. These strategies are widely utilized in hierarchical clustering algorithms [98], and we have employed them in our incremental clustering algorithm. To summarize, our strategies are as follows:

- In single-link strategy, when a new object (O_i) arrives, the incremental clustering algorithm calculates its similarity with objects already present in each cluster by taking into account their dimensions. Subsequently, it assigns O_i to the cluster that satisfies the maximum predetermined threshold. However, if the similarity score is below the predetermined threshold, a new cluster containing O_i is created.

- In the average-link strategy, like single-link strategy, the algorithm first computes the similarity between the new object (O_i.) and objects in each existing cluster. After that, it assigns O_i to a cluster with the highest average similarity. Otherwise, it creates a new cluster for O_i and updates cluster average result.
- In the complete-link strategy, upon the arrival of a new object (O_i), the algorithm evaluates its similarity with the objects in each cluster and then the new object will be assigned to the cluster that meets the highest minimum predetermined threshold. If the similarity value is below the predetermined threshold, it initiates a new cluster containing O_i.

5.4 Experimental Settings and Results

In this section, we first present the experimental setup consisting dataset, preprocessing tasks applied to the dataset, evaluation metrics, and implementation details. Next, we present experimental results of our event detection algorithm. Finally, we provide discussion on different cluster comparison and feature weighting strategies.

5.4.1 Experimental Settings

Dataset: For event detection experiments, we used MediaEval 2013 benchmark [**52**], which is often used for evaluating algorithms in the multimedia domain. It includes more than 21,169 events illustrated by 437, 370 images. The images were collected from Flickr and annotated with upload times (100%), capture times (98.3%), geotags (45.9%), uploaders (100%), and textual descriptions (including tags (95.6%), titles (97.9%), and content descriptions (37.9%)). For this work, we used 8, 023 events (35 %, after preprocessing) and their uploaded (Who), geotags (Where), capture times (When), and textual descriptions (What) dimensions.

Preprocessing: Before using the above dataset for evaluation, data preprocessing tasks need to be done to make the dataset more suitable for the algorithm. These tasks include data cleaning (e.g., *HTML tags, special characters, punctuation marks, non-English texts, and hasgtags removal*), XML element merging (*e.g.*, representing three separate XML elements, such tags, title, and description into one XML element), and metadata transformation (*e.g., transforming date/time data into UNIX timestamps and transforming polygon/line spatial value into single point spatial value*) as described in Section 5.2.2.

Cluster Comparison Strategies: For this, we used complete-link, single-link, and average-link strategies (cf. Section 5.3.2 for detail of cluster comparison strategies).

Evaluation Metrics: In order to measure the effectiveness of event detection, various evaluation metrics have been used, such as Rand Index [**102**], Purity [**101**], F-score [**99**], and NMI [**100**]. For this task, we utilized two widely recognized evaluation metrics in event detection, namely Normalized Mutual Information (NMI) and F-score. F-score takes into account both precision (PR) and recall (R) and is defined as follows:

$$F - score = \frac{2 \; x \; PR \; x \; R}{PR + R} \in [0,1] \tag{5.8}$$

In general, F-score result ranges from 0 to 1, where a score near to 1 indicates perfect precision (PR) and recall (R) and close to 0 indicates there is no correct detections.

On the other hand, Normalized Mutual Information (NMI) is an effective and widely used evaluation metric used to evaluate the accuracy of clustering outcomes. It measures the mutual information between the clustering results (K) and the true class assignments (C). To elaborate further, let us consider two distinct event clusters with n objects: 1) Clusters that were generated by our clustering algorithm $\mathrm{K} = \{K_1, K_2, \dots, K_i\}$ and 2) Ground-truth clusters $C = \{C_1, C_2, \dots, C_j\}$. The normalized mutual information between K and C is denoted $\mathrm{NMI}(\mathrm{K}, C)$ and can be defined as follows:

$$\mathrm{NMI}(\mathrm{K}, C) = \frac{2 * I(\mathrm{K}, \; C)}{[H(\mathrm{K}) + H(C)]} \in [0,1] \tag{5.9}$$

where

$$\mathrm{I(K, C)} = \sum_{k_i \in K} \sum_{c_j \in C} p(K_i, C_j) \, log \frac{p(K_i, C_j)}{p(K_i) + p(C_j)} \tag{5.10}$$

$$\mathrm{H(K)} = -\sum_{k_i \in K} p(i) \log \, p(i) \, , and \tag{5.11}$$

$$\mathrm{H(C)} = -\sum_{c_i \in C} p'(j) \log \, p' \, (j) \tag{5.12}$$

Where,

- **K**: represents ground-truth clusters;
- **C**: represents generated Clusters;
- $\boldsymbol{I}(\mathbf{K}, \boldsymbol{C})$: represents mutual information between K and C.
- $\boldsymbol{H}(\mathbf{K})$: symbolizes entropy of cluster K
- $\boldsymbol{H}(\boldsymbol{C})$: represents entropy of cluster C.
- $\boldsymbol{p}(\boldsymbol{K_i}, \boldsymbol{C_j})$: represents the probability that an event picked at random from cluster K_i in K and cluster C_j in C and it can be computed as, $p(K_i, C_j) = |K_i \cap C_j|/\mathrm{N}$ [**100**].

The NMI resulting score ranges between 0 and 1, with a score closer to 1 indicating perfect agreement between the cluster pairs, and a score closer to 0 indicating significant differences between the two clusters.

5.4.2 Experimental Results and Discussion

The previous section introduced experimental setup and showed how to use them in the event detection process. This section presents a series of experiments. The goal of this experiment is to identify significant events which can be used as input for Stage-IV (see Chapter 6). Moreover, for each experiment, results were measured and compared as follows:

5.4.2.1 Experiment #1: Non-Feature Weighted Event Detection Method

In experiment#1, we treated all dimensions equally (or they do not have any specific importance), $i.e., P_\varepsilon = T_\varepsilon = L_\varepsilon = S_\varepsilon$ and applied our clustering comparison strategies, as we discussed in Section 5.4.2. Then after, we achieved the highest F-score (**0.88**) and NMI (**0.91**) in unweighted dimensions configuration. Nonetheless, this experiment failed to exhibit the influence of multimedia event dimensions such as participant, temporal, spatial, and semantic in the event detection process. To overcome this, we used weighted features similarity technique to enhance cluster quality (*cf. Experiment#2*). Feature weighting is central, especially for cluster-based event detection, because not all features of a multimedia object are equally important. Feature weighting is a technique used to adjust the importance of different features in the event detection process. Feature weighting is required when some features are more important than others to determine the similarity between two objects. Below are the experimental details of Experiment#2.

5.4.2.2 Experiment #2: Feature Weighted Event Detection Method

In experiment 2, our clustering method was evaluated using feature weighting technique that assigns importance (or weights) to features, such as spatial, temporal, participant, and semantic. This process gradually adjusts the weights of individual features. In this study, the central goal of stepwise increment in feature weighting technique is to 1) iteratively adjust the feature weights and measure their performance; 2) identify specific features that have greater impact than others; and 3) enhance the performance of the presented event detection method.

For each individual feature, we consider weights in the range [0, 1] and apply a stepwise increment of 0.1: the first is 0.0, the second is 0.1, the third is 0.2, and so on, until the upper limit (1.0) is reached. For instance, we used different values of w_L from 0.0 to 1.0, with 0.1 increments, while the weight values of w_p, w_T, and w_S were set equal based on the weight variation strategy described in research [**9**], i.e., $w_p = w_T = w_S =$ (1-$w_L/2$), resulting weights range from 0.0 to 0.50 with 0.05 stepwise increments. Experimental results show that the feature-weighted clustering approach not only improves the NMI and F-score scores, but also outperforms the non-feature weighted clustering approach (see Experiment 1). Additionally, the performance of our clustering algorithm further improved when we varied all feature weights in the range from 0.30 to 0.45 with 0.05 increments, shown in Table 5.1.

Overall, like spatial dimension presented above, we performed similar feature-weighting experiments for participatory, temporal, and semantic dimensions. For each experiment, experimental results were measured and compared using a clustering comparison strategy (see results in Figure 5.2 a, b and c). In all feature-weighting experiments, we noticed that clustering results can be over-influenced (unbalanced) if one dimension is weighted higher (*e.g.,*$w_L = 0.8$) than others (*e.g.,* $w_p = w_T = w_S = 0.15$). For instance, when spatial dimension/feature weighed higher than others, semantically distinct events and spatially identical events may be grouped together. To minimize this imbalance, we normalize feature weights to balance the contribution of each feature using inclusion and exclusion criteria. For this purpose, we consider the parameters (*or weights ranging from 0.30 to 0.45*) that produce the best results in the experiments above. Weight values were increased from 0.30 (lower limit) to 0.45 (upper limit), with 0.05 increments (e.g., $w_P = 0.30,\ w_T = 0.35, w_L = 0.40,\ \&\ w_s = 0.45$), as shown in Table 5.1. Please note that the result of Table 5.1 generated from 4 different tables.

Table 5.1: The top five results obtaining by varying the weight values in the range [0.30, 0.45].

Weights				Strategies					
				Average-link		Single-link		Complete-link	
W_P	W_T	W_L	W_S	NMI	f-score	NMI	f-score	NMI	f-score
0.30	0.35	0.40	0.45	0.9925	0.9801	0.9823	0.9798	0.9711	0.9661
0.45	0.40	0.35	0.30	0.9821	0.9789	0.9723	0.9790	0.9698	0.9483
0.40	0.45	0.35	0.30	0.9743	0.9745	0.9537	0.9722	0.9467	0.9534
0.35	0.30	0.45	0.35	0.9816	0.9722	0.9816	0.9815	0.9822	0.9833
0.40	0.30	0.35	0.45	0.9928	0.9914	0.9904	0.9900	0.9892	0.9879

Table 5.1 shows the top five results from a set of various parametric settings for average-link, single-link, and complete-link strategies. As can be seen in Table 5.1, $w_P = 0.40$, $w_T = 0.30,\ w_L = 0.35,\ and\ w_S = 0.45$ gave the highest F-score value (*0.9914*) and NMI value (*0.9928*).

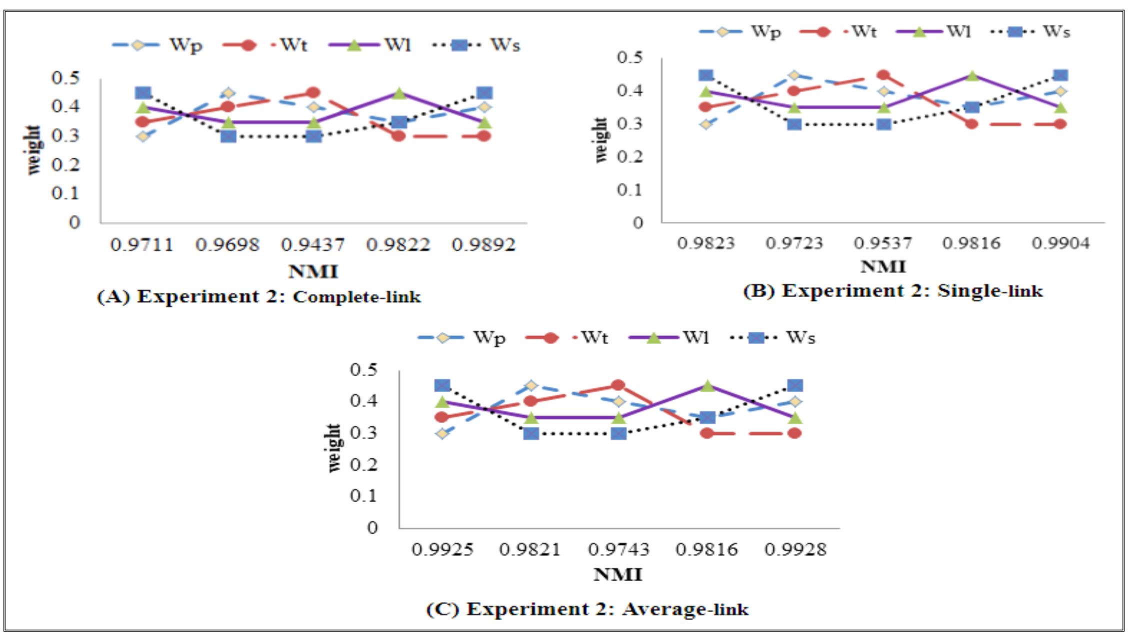


Figure 5.2: NMI results highlight the impact of participant, temporal, spatial and semantic dimensions on the event detection algorithm (*a similar line charts can be obtained for F-score results*).

5.4.2.1 Discussion

We carried out two experiments to assess the efficacy of the event detection method, as described in the previous sections. The results of these experiments demonstrate three key observations: **1**) treating all features equally important, $i.e., P_{\mathcal{E}} = T_{\mathcal{E}} = L_{\mathcal{E}} = S_{\mathcal{E}}$ (or ignoring feature-weighted technique), reduces the quality of clusters; **2**) assigning upper feature weight to spatial only or temporal only (*e.g.*,$w_L = 0.8$) than others (*e.g.*, $w_p = w_T = w_S = 0.15$), produces in most case imbalance clustering results; **3**) when detecting events, it is essential to take into account small feature weights (between 0.30 and 0.45) and close weight values for w_P, w_T, w_L *and* w_S. Furthermore, while all four dimensions have a significant impact on event identification, it is important to give more emphasis on the semantic (What) and participant (Who) dimensions of multimedia objects. For example, if the participant is a singer, it will not be so difficult to guess that the event is a concert. Similarly, understanding the semantic feature of multimedia object is essential to distinguish events from non-events in social media documents. To summarize, we observed that increasing the weights of w_P *and* w_S improves the quality of the clusters.

5.5 Summary

One of the specific objectives of this dissertation was to identify real-world events that were used as input for Stage-IV and Stage-V. To achieve this objective, we developed a scalable event detection model consisting of two main sub-components: similarity measures and cluster comparison strategies. The model was evaluated using MediaEval 2013, commonly used dataset to evaluate algorithms in multimedia event detection. Lastly, we assessed the effectiveness of the event detection algorithm by utilizing evaluation metrics like NMI and F-score. The experimental findings indicate that the proposed model is efficient

CHAPTER 6
STAGE-IV: EVENT RELATIONSHIPS IDENTIFICATION

The previous chapter introduced event detection model that produces clustered real-world events, which can be used as input to this chapter (or Stage IV implementation). The detected events can be linked to each other, through cause, location, time, and others. Therefore, after event detection, the next task is to identify semantic relationships between events. To accomplish this, we propose an approach that uses an unsupervised machine learning algorithm to identify three main types of event relationships (*spatial, temporal, and causal*) as well as their sub-relationships. This chapter also explains a formal concept definition, a set of rules, and experimental setup. Finally, experimental results are presented in this chapter.

6.1 Introduction

Events in social media sites are multi-media, multi-features, and multi-sources. Such events may be connected to one another in various ways. Identifying these connections/relationships can help us to better understand the concepts surrounding the events and will enable us to create an Event Knowledge Graph and retrieve events from it. Before describing event relationship categories and machine learning technique used in identifying them, let us initially consider the overall meaning of an event relationship. According to studies on event relationship identification (e.g., [**9, 33**, **23**]), event relationship can be defined as connection between events. It's important to note that the type and manner of this connection can differ. For instance, events may be connected in a temporal, semantic, or spatial way. With this in mind, the authors of [**9**, **33**] define an event relationship as the temporal, spatial, and semantic links between two events, taking into account their temporal, spatial, and textual aspects. In contrast, the research presented in [**23**, **56**, **57**] defines an event relationship as cause-and-effect relationship between events in tweets, in which one event (the cause) is the reason for the other event (the effect).

In social media data, a few event relationship identification techniques have been developed (e.g., [**33, 57**), but they face the following limitations: *(1)* are domain dependent and only take into account specific type of document (*e.g., Twitter only*), and *(2)* focus on classification problems to identify one relation only (*e.g., causal*) and ignoring the identification of different types of event relationships as well as their sub-relationships between events (*see Section 6.2*). To overcome these, we proposed a novel approach that leverages a set of predefined rules to identify three main relationships and their sub-relationships.

Our model consists of four major steps: 1) representing events along with their temporal, spatial, and textual expressions; 2) developing a set of rules based on identified patters and relationships in the event data; and 3) validating and refining the rules to determine their accuracy; and 4) identifying different types of relationships between events based on our rules in unsupervised setting. All of these steps and each connection category between events are elaborated in the subsequent sections.

6.2 Event Relationship Type Definitions

Defining event relationship is a key foundation for understanding and identifying relationships between events. The next sections provide definitions for each type of event relationship used in this study.

6.2.1 Temporal Relationship

To define temporal relationships, we examined three forms of temporal dimension representations: 1*)* point-based; *2)* duration-based; and *3)* temporal coverage representative point-based timestamps. By utilizing these timestamps, we define temporal relationship as follows.

Definition 6.1 Temporal Relationship ($\boldsymbol{T_r}(\mathcal{E}_1,\ \mathcal{E}_2)$). It refers to the order or interval of occurrence overtime in event document (or it refers to the relationship between events that are related to time). Here, the 13 relations of the Allen's interval [**59**] can be used to infer the temporal relationship between objects (events) utilizing their temporal dimensions and formalized as $R\left(\mathcal{E}_i,\ \mathcal{E}_j\right) = (Tem)$, where $Tem \in \{b,\ bi,\ s,\ si,\ m,\ mi,\ o,\ oi,\ f,\ fi,\ eq,\ d,\ \&\ di\}$. Details about these temporal relationships are presented in Table 6.1.

Table 6.1: Allen's temporal relations and their definition

Notation	$T_r(\mathcal{E}_1, \mathcal{E}_2)$	Definition
B	Before	$\mathcal{E}_i\ b\ \mathcal{E}_j$ = *if* $T\,(\mathcal{E}_i)$ *precedes* $T(\mathcal{E}_j)$
Bi	After	$\mathcal{E}_i\ bi\ \mathcal{E}_j$ = *if* $T\,(\mathcal{E}_j)$ *is preceded by* $T(\mathcal{E}_i)$ *(or the inverse of before)*
S	Starts	$\mathcal{E}_i\ s\ \mathcal{E}_j$ = *if* $T\,(\mathcal{E}_i)$ & $T(\mathcal{E}_j)$ *share their start points and* $T\,(\mathcal{E}_i)$ *ends before* $T\,(\mathcal{E}_j)$
Si	Started-by	$\mathcal{E}_i\ si\ \mathcal{E}_j$ = *if* $T\,(\mathcal{E}_i)$ & $T(\mathcal{E}_j)$ *share their start points and* $T\,(\mathcal{E}_i)$ *ends after* $T\,(\mathcal{E}_j)$ *(or the inverse of starts)*
M	Meets	$\mathcal{E}_i\ m\ \mathcal{E}_j$ = *if* $T(\mathcal{E}_i)$ *stops exatly when* $T(\mathcal{E}_j)$ *starts*
Mi	Met-by	$\mathcal{E}_i\ mi\ \mathcal{E}_j$ = *if* $T(\mathcal{E}_i)$ *starts exatly when* $T(\mathcal{E}_j)$ *ends (or the inverse of meets)*
O	Overlaps	$\mathcal{E}_i\ o\ \mathcal{E}_j$ = *if* $T(\mathcal{E}_i)$ *overlaps wiht* $T(\mathcal{E}_j)$
Oi	Overlapped-by	$\mathcal{E}_i\ oi\ \mathcal{E}_j$ = *if* $T(\mathcal{E}_j)$ *is overlapped by* $T(\mathcal{E}_i)$ *(or the inverse of overlaps)*
F	Finishes	$\mathcal{E}_i\ f\ \mathcal{E}_j$ = *if (*$T(\mathcal{E}_i)$ & $T(\mathcal{E}_j)$ *stop at the same time and* $T\,(\mathcal{E}_i)$ *starts after* $T\,(\mathcal{E}_j)$
Fi	Finished-by	$\mathcal{E}_i\ fi\ \mathcal{E}_j$ = *if (*$T(\mathcal{E}_i)$ & $T(\mathcal{E}_j)$ *stop at the same time and* $T\,(\mathcal{E}_i)$ *starts before* $T\,(\mathcal{E}_j)$*(or the inverse of finishes)*
Eq	Equals	$\mathcal{E}_i\ eq\ \mathcal{E}_j$ = *if* $T\,(\mathcal{E}_i)$ & $T(\mathcal{E}_j)$ *have the same temporal coverage*
D	During	$\mathcal{E}_i\ d\ \mathcal{E}_j$ = *if* $T\,(\mathcal{E}_i)$ *statrs & ends while* $T(\mathcal{E}_j)$ *is occurring*
Di	Contains	$\mathcal{E}_i\ di\ \mathcal{E}_j$ = *if* $T\,(\mathcal{E}_j)$ *contains* $T(\mathcal{E}_i)$

Note that the study adopts only 7 event temporal relationships that are commonly used in the literature, i.e. starts, before, finishes, after, during, overlaps, and equal. Figure 6.1 presents graphically these temporal relationships.

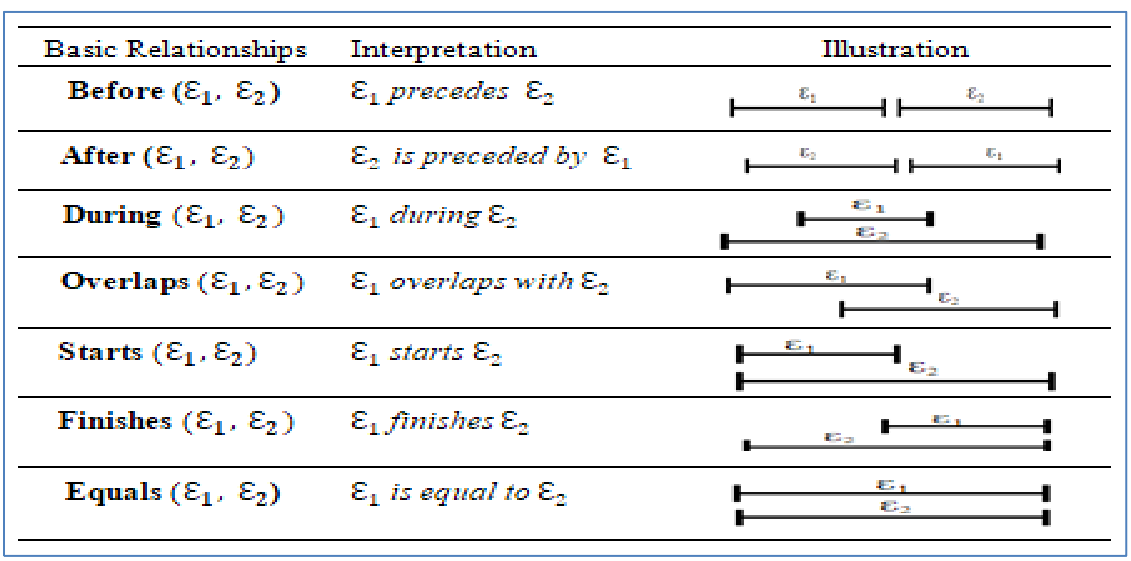


Figure 6.1: Types of temporal relationships between events and their interpretation

6.2.2 Spatial Relationship

Similar to the temporal relationship, the spatial dimension associated with each multimedia event can be used to establish the connection between events. In our case, different events can be linked through any of the following three spatial relationships: 1) topological (e.g., *overlap, equal, meet, within, and disjoint*); 2) directional (e.g., *south, north, west, and east)*; and 3) distance (*e.g., far and near*). Below are detailed definitions for each category of spatial relationship.

Definition 6.2 Topological relationships ($\boldsymbol{S_r^{Topo}}(\mathcal{E}_i, \mathcal{E}_j)$). It refers to the spatial relationships between events based on topological spaces. The concepts of adjacency, interior, enclosure, and boundary of objects are utilized to identify relationships between events. It is important to note that these relationships were identified based on either simple spatial representation (e.g., *spatial data represented as single points*) or complex spatial representation (*e.g., spatial data represented as lines and polygons).* In this context, we adopt the 5 fundamental topological relationships that are commonly used in the literature [**9, 33**]: *meet, disjoint, overlaps, within, and equal*. These relationships are defined as follows:

$$\boldsymbol{S_r^{Topo}}(\mathcal{E}_i, \mathcal{E}_j) = \begin{cases} \text{Meet} & if\ \mathcal{E}_i\ meets\ \mathcal{E}_j \\ Disj & if\ \mathcal{E}_i\ disjoint\ \mathcal{E}_j \\ Overlaps & if\ \mathcal{E}_i\ overlaps\ with\ \mathcal{E}_j \\ Within & if\ \mathcal{E}_i\ within\ \mathcal{E}_j \\ Equal & if\ \mathcal{E}_i\ is\ equals\ to\ \mathcal{E}_j \end{cases} \tag{6.1}$$

where, $\mathcal{E}_i$ and $\mathcal{E}_j$ are two events along with spatial values.

- Two events ($\mathcal{E}_i\ and\ \mathcal{E}_j$) topologically meet if they have at list one point in common at their boundary.
- Two events ($\mathcal{E}_i\ and\ \mathcal{E}_j$) topologically disjoint if they do not share any points or they do not touch each other.
- The topologically overlap relationship between events ($\mathcal{E}_i\ and\ \mathcal{E}_j$) be true if they share at least one/some geospatial points, but not all points in common.
- An event ($\mathcal{E}_i$) topologically within another ($\mathcal{E}_j$) if the spatial value of $\mathcal{E}_i$ is completely contained inside $\mathcal{E}_j$.

- Two events ($\mathcal{E}_i$ *and* $\mathcal{E}_j$) are topologically equal if they have the same spatial dimension representation (*e.g., two line representations have the same start and end points*).

Definition 6.3 Directional relationships ($S_r^D(\mathcal{E}_i, \mathcal{E}_j)$). They refer to the spatial relationships among events that are established on the basis of their direction with respect to one another. These can be used for describing the spatial relationships between events. For example, if we have two events ($\mathcal{E}_i$ *and* $\mathcal{E}_j$), we can say $\mathcal{E}_i$ north of $\mathcal{E}_j$. In this context, we used the four frequently used directional relationships, namely south, north, west, and east. The work in [**9**] defined directional relationships as:

$$S_r^D(\mathcal{E}_i, \mathcal{E}_j) = \begin{cases} \mathcal{E}_i \xrightarrow{D_{North}} \mathcal{E}_j & if\ \mathcal{E}_i\ is\ north\ of\ \mathcal{E}_j(\ \mathcal{E}_i\ N\ \mathcal{E}_j) \\ \mathcal{E}_i \xrightarrow{D_{South}} \mathcal{E}_j & if\ \mathcal{E}_i\ is\ south\ of\ \mathcal{E}_j(\ \mathcal{E}_i\ S\ \mathcal{E}_j) \\ \mathcal{E}_i \xrightarrow{D_{East}} \mathcal{E}_j & if\ \mathcal{E}_i\ is\ east\ of\ \mathcal{E}_j\left(\ \mathcal{E}_i\ E\ \mathcal{E}_j\right) \\ \mathcal{E}_i \xrightarrow{D_{West}} \mathcal{E}_j & if\ \mathcal{E}_i\ is\ west\ of\ \mathcal{E}_j(\ \mathcal{E}_i\ W\ \mathcal{E}_j) \end{cases} \tag{6.2}$$

where, $\mathcal{E}_i$ and $\mathcal{E}_j$ are two multimedia events with spatial points. To identify directional relationships between these two events, a three-step process are followed. Firstly, ranges/angles in degree are measured for every representative point. Secondly, we infer directional relationships based on an entry of degrees (or degrees are converted into cardinal direction strings). Finally, we establish the directional relationships between two objects. For each of them, more details descriptions are presented in Table 6.2.

Table 6.2: Rules for directional relationships

Cardinal direction	Range of degree
North	Degree >= 0 & Degree < 90
East	Degree >= 90 & Degree < 180
South	Degree >= 180 & Degree < 270
West	Degree >= 270 & Degree < 360

It is important to note that any change in the position of an event will result in a change in the relationships between the events as well.

Definition 6.4 Distance-based Relationships ($S_r^{Dist}(\mathcal{E}_i,\ \mathcal{E}_j)$). It refers to the spatial relationship between two or more events$(\mathcal{E}_i,\ \mathcal{E}_j)$ based on the distances between them. This relation can be expressed as either near or far. In order to identify these two relationships, our approach utilized pairs of events $(\mathcal{E}_i,\ \mathcal{E}_j)$ with spatial values and a user-defined threshold ($Maxt_d$) as input and outputs near/far relationships. These relationships are defined as follows:

$$S_r^{\mathrm{Dist}}(\mathcal{E}_i,\ \mathcal{E}_j) = \begin{cases} \mathcal{E}_i \xrightarrow{Dist_{near}} \mathcal{E}_j & if\ Dist_S(\mathcal{L}_i(\mathcal{E}_i),\ \mathcal{L}_j(\mathcal{E}_j)) < Maxt_D \\ \mathcal{E}_i \xrightarrow{Dist_{far}} \mathcal{E}_j & if\ Dist_S(\mathcal{L}_i(\mathcal{E}_i),\ \mathcal{L}_j(\mathcal{E}_j)) > Maxt_D \end{cases} \tag{6.3}$$

where,

- $\mathcal{L}_i(\mathcal{E}_i)\ and\ \mathcal{L}_j(\mathcal{E}_j)$ are the spatial representative points
- $Maxt_D$ is user-defined distance threshold, which serves as the foundation for identifying the two distinct types of spatial connections.

In summary, if the distance between two events $(\mathcal{E}_{i,}, \mathcal{E}_j)$ is less than the user-defined distance threshold, they are considered spatially close to each other otherwise declared as far.

6.2.3 Semantic Relationship

Though spatial relationships have been well-defined and studied in [**9**], the causal relationships within multimedia events are ignored. Our study addresses this problem by presenting an approach that can extract causal connections between events showing effect occurring after the cause in linear order. Causal connections can be classified into two categories: *direct and indirect*. This study considers only direct cause-and-effect relationship as defined below:

Definition 6.5 Direct Causal Relationship ($r_C^{Dir}(\mathcal{E}_i \rightarrow\ \mathcal{E}_j)$). It is a relationship between two events, where the cause event $(\mathcal{E}_i)$ leads to the occurrence of the effect event $(\mathcal{E}_j)$. In other words, it is a relationship that always directed from the cause to the effect $(\mathcal{E}_i \rightarrow \mathcal{E}_j)$, thus the effect event cannot happen without the cause [**23**]. This is defined as follows:

$$r_C^{Dir}(\mathcal{E}_i,\ \mathcal{E}_j) = \begin{cases} Direct\ Causal & if\ \mathcal{E}_i \text{ causes } \mathcal{E}_j \\ Non-causal & Otherwise \end{cases} \tag{6.4}$$

where, r_C^{Dir}: represents a causality identifier that accepts textual features as input and produces *"Direct Causal "or "Non-causal"* as output [**23**]. Note that we primary used the textural feature (*What*) to identify causality between events.

6.3 Rule Sets for Event Relationships Identification

Utilizing relationship between events introduced in the previous sections, our objective in this section is to create rule sets that serve as the foundation for identifying event relationships. Table 6.3 illustrated rules used to identify event relationships.

Table 6.3: Rules used to identify diverse types of relationships between events.

Rules	***Rel- Type***	***Rel-Categories***
if ($T(\mathcal{E}_1)$) PRECEDS ($T(\mathcal{E}_2)$) ***then*** *label (Before)*	*Before*	
if ($T(\mathcal{E}_2)$) is PRECEDED by ($T(\mathcal{E}_1)$) ***then*** *label(After)*	*After*	
if ($T(\mathcal{E}_1)$ & $T(\mathcal{E}_2)$) BEGIN and END at THE SAME TIME ***then*** *label (Simultaneous)*	*T_Sim*	*Temporal*
if ($T(\mathcal{E}_2)$) occurs during ($T(\mathcal{E}_1)$) ***then*** *label (During)*	*During*	
if ($T(\mathcal{E}_1)$ STARTS before ($T(\mathcal{E}_2)$) & ENDS during $\mathcal{E}_2$ ***then*** *label (Overlaps)*	*Overlaps*	
if ($T(\mathcal{E}_1)$ & $T(\mathcal{E}_2)$) share their START point and $T(\mathcal{E}_1)$ ENDS before $T(\mathcal{E}_2)$ ***then*** *label (Starts)*	*Starts*	
if ($T(\mathcal{E}_1)$ & $T(\mathcal{E}_2)$) share their END point and $T(\mathcal{E}_1)$ BEGINS later ***then*** *label (Ends)*	*Ends*	
if ($S\ (\mathcal{E}_1)$ shares the boundary of $S\ (\mathcal{E}_2)$) ***then*** *label (Meet)*	*Meet*	
if ($S\ (\mathcal{E}_1)$ & $S\ (\mathcal{E}_2)$) equal ***then*** *label (Equal)*	*Equal*	
if ($S\ (\mathcal{E}_1)$ & $S\ (\mathcal{E}_2)$ spatially disconnected ***then*** *label (Disjoint)*	*Disjoint*	*Spatial: Topological relationship*
if ($S\ (\mathcal{E}_2)$ completely encloses by $S\ (\mathcal{E}_1)$ ***then*** *label (Within)*	*Within*	
if ($S\ (\mathcal{E}_1)$ partly covers by $S(\mathcal{E}_2)$) ***then*** *label (Overlap)*	*Overlap*	
if $S\ (\mathcal{E}_1)$ is NORTH of $S\ (\mathcal{E}_2)$ ***then*** *label (North)*	*North*	
if $S\ (\mathcal{E}_1)$ is SOUTH of $S\ (\mathcal{E}_2)$ ***then*** *label (South)*	*South*	*Spatial: Directional relationship*
if $S\ (\mathcal{E}_1)$ is EAST of $S\ (\mathcal{E}_2)$ ***then*** *label (East)*	*East*	
if $S\ (\mathcal{E}_1)$ is WEST of $S\ (\mathcal{E}_2)$ ***then*** *label (West)*	*West*	
if $Dist_S(\mathcal{L}_1(\mathcal{E}_1), \mathcal{L}_2(\mathcal{E}_2)) < Maxt_D$ ***then*** *label (Near)*	*Near*	*Spatial: Distance-based relationship*
if $Dist_S(\mathcal{L}_1(\mathcal{E}_1), \mathcal{L}_2(\mathcal{E}_2)) > Maxt_D$ ***then*** *label (Far)*	*Far*	
if $\mathcal{E}_1$ causes $\mathcal{E}_2$ & $\mathcal{E}_1$ proceeding $\mathcal{E}_2$ ***then*** *label (Dir_c_rel)*	*Dir_c_rel*	*semantic relationship*

6.4 Event Relationships Identification Algorithm

The algorithm primarily focuses on two tasks: using pairs of events and applying event relationship rule sets as outlined in Table 6.3 to identify causal, temporal, and spatial relationships between events. Figure 6.2 shows pseudo code for identifying event relationships.

Algorithm 6.1: Event Relationships Identification

Input:

Event Pairs (Event$_i(Dimen_k)$, $Event_j(Dimen_n)$)

Output:

R$_m(event_i, event_j)$ ▷ Different kinds of event relationships

Begin

if $Pairs(Temp_k(event_i), Temp_n(event_j)) \in event - data$ **then**

  Temp-Relation$(event_i, event_j) \leftarrow Temporal - Rules(pairs(Temp(event_i), Temp(event_j))$ ▷ *Following temporal definition (6.1) and its inference rules;*

  Return *temporal relation types;*

else if $Pairs(Spatial_k(event_i), Spatial_n(event_j)) \in event - data$ **then**

  Spatial-Relation$(event_i, event_j) \leftarrow Spatial - Rules(pairs(Spatial(event_i), Spatial(event_j))$ ▷ *Following spatial definitions (6.2, 6.3 and 6.4) and their inference rules;*

  Return *spatial relation types;*

else if $Pairs(Cause_k(event_i), Effect_n(event_j)) \in event - data$ **then**

  Causal-Relation$(event_i, event_j) \leftarrow Causal - Rules(pairs(Cause(event_i), Effect(event_j))$ ▷ *Following Causal definition (6.5) and its inference rules;*

  Return *event causality;*

else

  Go back to step 1

Figure 6.2: Pseudo code for identifying event relationships

6.4.1 Identification of Temporal Relationships

In order to identify temporal relationships, previous studies have used different possible pairs of events using their temporal features. These pairs of events ($\mathcal{E}_i$,$\mathcal{E}_j$) are often grouped into four major categories [**103**]: 1) *Timex-Timex, 2) Event-DCT, 3) Event-Timex, and 4) Event-Event.* All of these pairs work well in rich textual narrative texts, such as news articles, where it is not difficult to create all event pairs mentioned above and identify their relationships. Not all pairs mentioned above can be applied to multimedia events published on social networks, due to their short textual descriptions. But, we can create event-event pairs by utilizing content creation timestamps and textual descriptions of multimedia events in order to identify temporal relations between events. This is done in three phases.

- The first phase identifies events and timestamps associated with each event.
- In the second phase, we created event-event pairs ($\mathcal{E}_i(T_i)$-$\mathcal{E}_j(T_j)$) containing temporal expressions.
- Finally, we identified temporal relationships between events by comparing their timestamps.

More specifically, our algorithm takes time-based event pairs and produces temporal relationships between events. The extracted relationships can be expressed as $R(\mathcal{E}_i \xrightarrow{T-relation-type} \mathcal{E}_j)$ and represented as edges in the Event Knowledge Graph.

6.4.2 Identification of Spatial Relationships

Similar to the temporal relationship identification, we begin by creating spatial-based event pairs ($\mathcal{E}_1(\mathcal{L}_1 = (lat_1, lon_1)$, $\mathcal{E}_2(\mathcal{L}_2 = (lat_2, lon_2))$ that include region, line, and point representations. By comparing location-based features associated with each event, spatial relationships between events can be identified. For example, we can compare the geo-spatial points associated with each event to determine if they are close or far. In this study, we consider eleven spatial event relationships that can be classified into three main groups: *distance–based, topological, and directional (see Section 6.2.2 for more details)*. In summary, the algorithm takes spatial-based event pairs and produces the above mentioned spatial relationship types between events.

6.4.3 Identification of Direct Causal Relationship

Algorithm 6.1 also has the ability to detect the causal relationship between events, but this is a challenging task [**104**]. To identify the causal relationship, we primarily used textual descriptions (What). In these textual descriptions, the most difficult task is to *(i)* discover semantic descriptions that contain both cause and effect events and *(ii)* decide whether $\mathcal{E}_1$ causes $\mathcal{E}_2$ or $\mathcal{E}_2$ causes $\mathcal{E}_1$. In order to address this issue, the algorithm relies on causal marker words/phrases (*such as "since", "as a result", "because", "lead to", "following", "because of" & "due to" and others*) to establish causal structure dependence, allowing it to identify cause-and-effect pairs and their relationship. For example, consider the following content description, *["heavy rain ($\mathcal{E}_1$) leads to flash floods ($\mathcal{E}_2$)"]*, within which $\mathcal{E}_1$causes $\mathcal{E}_2$ because the phrase "*leads to*" can connect a cause

NP (e.g., *heavy rain*) to an effect NP (e.g., *flooding*). Notably, the causality is dependent on the temporal dimensions, as the cause event usually occurs before the effect. Finally, both causal and effect events as well as relationship between them are represented as a knowledge graph.

6.5 Evaluations

In this section, we performed three different experiments using the methods described in Chapter 6. We first present the relevant datasets in Section 6.5.1. We then present the results of each experiment in Sections 6.5.2, 6.5.3, and 6.5.4.

6.5.1 Experimental Datasets

In this thesis, we initially conducted an investigation on a publicly available dataset commonly used to evaluate the performance of event relationship identification algorithms. Based on our investigation findings, we decided to create our datasets for evaluating the temporal and spatial relationships identification algorithms. This is because existing datasets do not take into account some temporal data types (*such as duration based Timestamps*) and spatial data types (*such as points, lines, and polygons*). Additionally, most event-related datasets concentrate primarily on text sources. Therefore, we need a dataset containing the basic data types of events extracted from the multimedia digital ecosystem. For this purpose, we utilized 510 event multimedia data identified by our event detection algorithm (see Chapter 5). For each event multimedia data, we record its multimedia object id (*e.g., video or image-id*), textual description (*e.g., content description, title, and tags*), the associated spatial expression (*e.g., Latitude-Longitude coordinates*) and the associated temporal expression (*e.g., time/date*). Notably, we exclude participant dimensions since we do not consider relationships that may be extracted from these dimensions. After completing the preprocessing steps highlighted in Chapter 5, we developed the following two datasets:

- **Temporal Relationship Dataset:** Within this dataset, 310 multimedia events are annotated based on three temporal data types: image Timestamp, video Timespan and sentence-based time expressions from textual descriptions. Additionally, the dataset includes seven temporal relationship types between pairs of events, as detailed in Section 6.4.1.

- **Spatial Relationship Dataset:** Similar to the temporal relationship dataset, we created 200 event representations containing event spatial features, such as points, lines, polygons, and spatial coverage representative points for each line and polygon coverage as well as all spatial relationship types between pairs of events (see, Section 6.4.2).
- **Semantic Relationship Dataset:** For evaluating semantic relationship identification, we utilized the dataset developed by [**23**] containing 913 pairs of causal events that were annotated as either "causal" or "non-causal."
- **Evaluation Metric:** We used F-score evaluation metric for measuring the performance of event relationships identifiers.

6.5.2 Experimental Results on Temporal Relationships Identification

In this part, we carried out two experiments using the aforementioned dataset to assess the effectiveness of our temporal relationship identifier. The first experiment focused on only the spatial data associated with each multimedia event, such as Timestamps for images or videos. The second experiment took into account textual descriptions/dimensions that contained temporal expressions. In summary, experiment 1 yielded superior results compared to experiment 2. Let us see with example why experiment 1 better than experiment 2, consider the following two textual dimensions attached with two multimedia objects (events), extracted from YouTube: (1) "*At least 10 people were killed and 20 others were wounded on Tuesday (June 20) in a car bombing in Somali capital Mogadishu*" and (2) "*Ketera and Timkat Celebration in Addis Ababa.*" For these two textual descriptions, the proposed algorithm failed to identify the temporal relationship between events, particularly when there was no relational indicator expression between two events. For the first textual description, the temporal relationship type between "killed (e_1)" and "wounded (e_2)" events should be "*T_Simultaneous*". For the second textual description, the temporal relationship type should be "*T_before*" as "*Ketera*" occurred before "*Timkat*." Finally, the best results obtained from our experiments are presented in Figure 6.3.

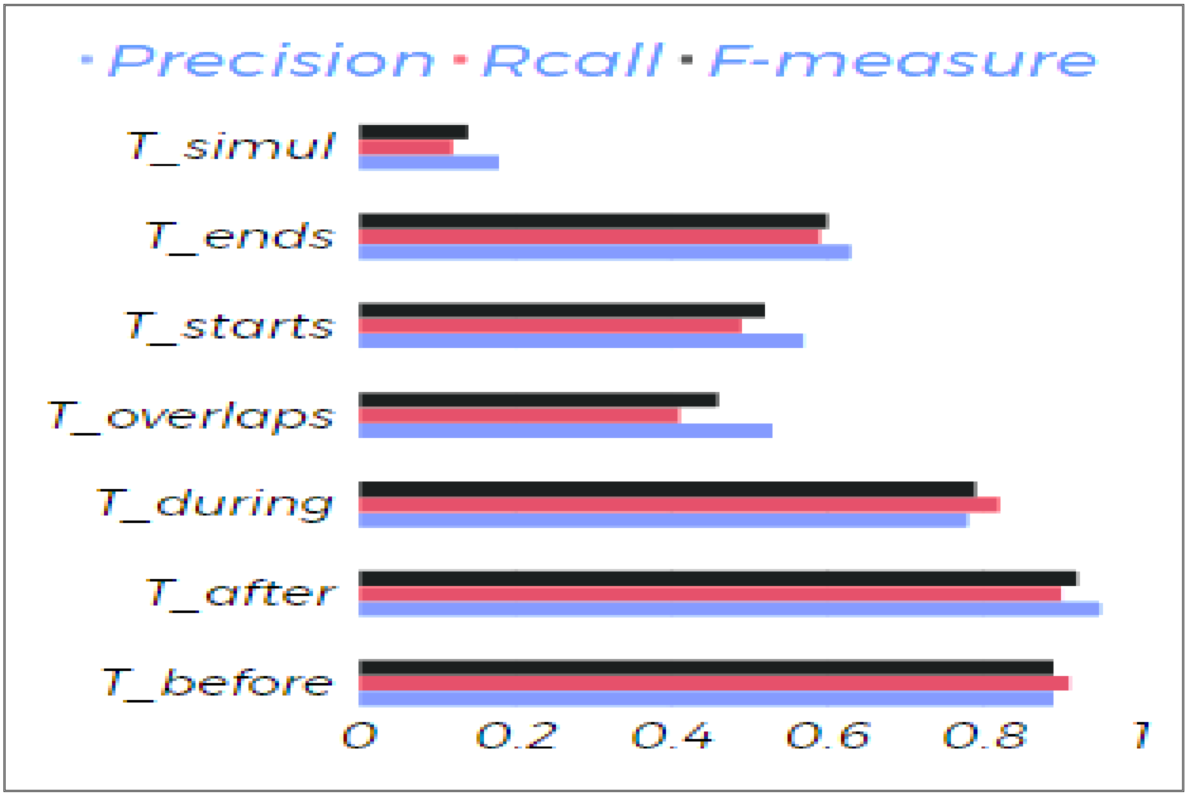


Figure 6.3: Illustrates performance results for temporal relationship identifier.

6.5.3 Experimental Results on Spatial Relationships Identification

This section concerns the evaluation of three identifiers for spatial relationships: distance-based, topological, and directional relationships identifiers. These three identifiers utilize spatial data to generate distinct types of spatial relationships. To assess their efficacy, we utilized the dataset outlined in Section 6.5.1. The following subsections present detailed experiments and obtained findings for each of the spatial relationship identifiers.

Distance-based Spatial Relationship Experiment: To identify distance-based spatial relationships between two events, which can be expressed as *S_near* $(\mathcal{E}_i, \mathcal{E}_j)$ and *S_far* $(\mathcal{E}_i\ \ \mathcal{E}_j)$, measuring the distance between two events using their linear or polygonal representation is computationally intensive. To overcome this, we utilized a spatial coverage representative point strategy that uses the centroid point for polygonal and liner representations (cf. Figure 6.4 (B) and (C)). This approach allowed us to efficiently address the computational challenge.

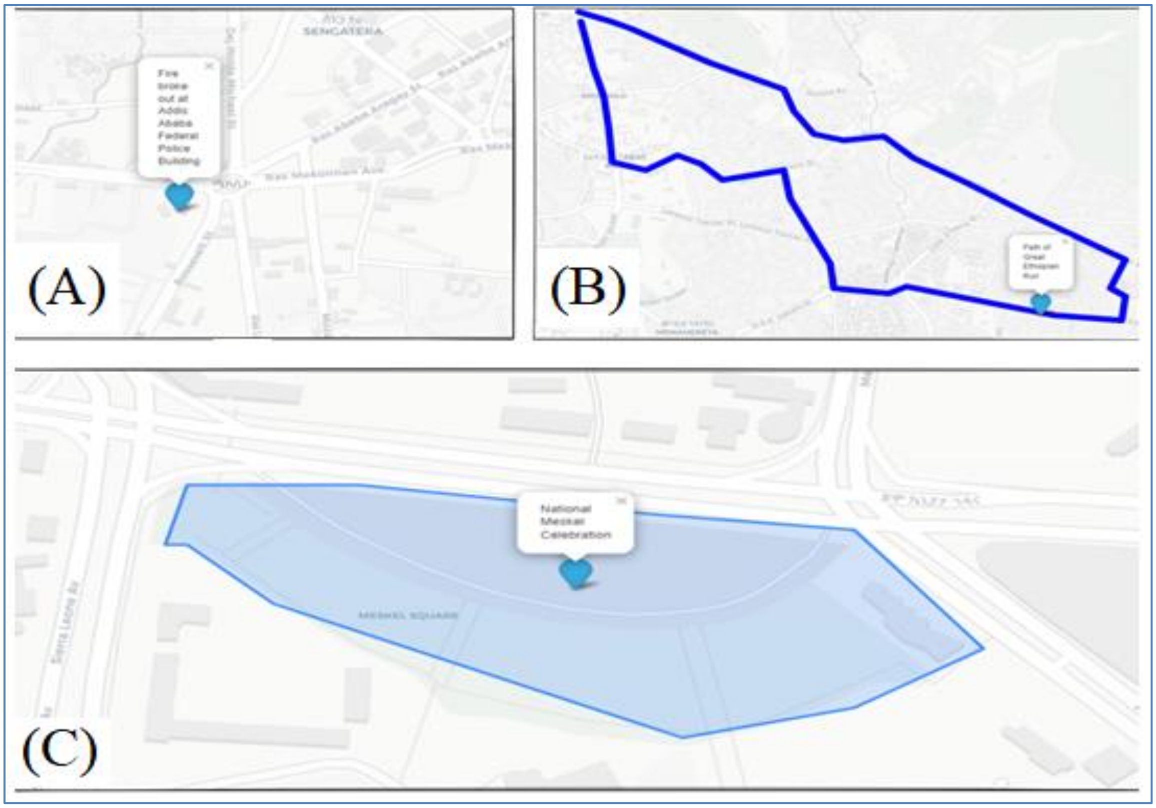


Figure 6.4: Snapshot of multimedia events that occurred at a point (A), line (C), and polygon (B).

In (A), the location marker refers an event that occurred at a single point. In (B), the marker represents a spatial coverage representative point for the national Meskal celebration, while in (C), it represents a spatial coverage representative point for the Great Ethiopian Run in Addis Ababa.

Once representative points for each spatial coverage have been identified, we can use these points to convert from degrees to radians to measure the distance between two events $(Dist_s\ (\mathcal{L}_1 = (lat_1,\ lon_1)\ \&\ (\mathcal{L}_2 = (lat_2,\ lon_2))$ using different distance metrics, such as Geopy great circle, Haversine formula, Haversine with sklearn, and Geopy. The differences between these distance metrics can be seen in Figure 6.5.

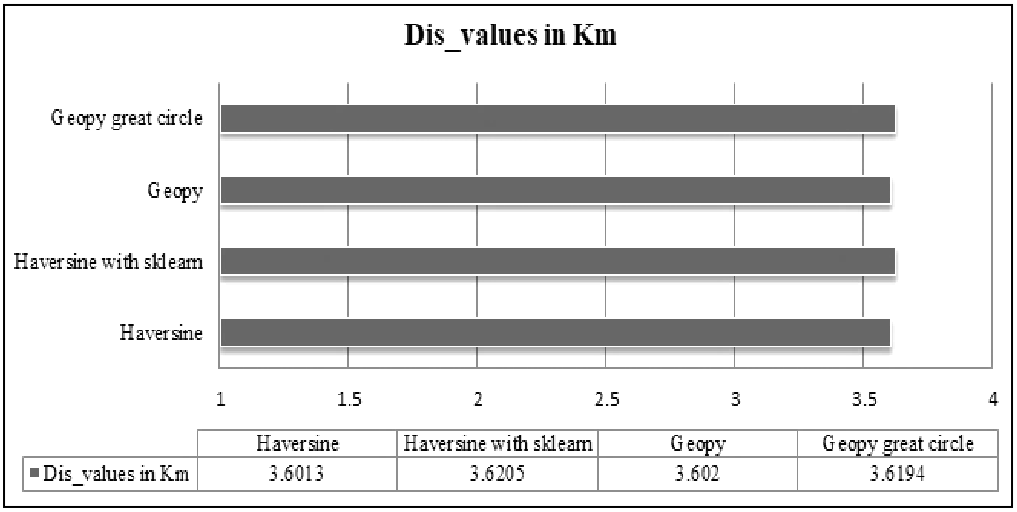
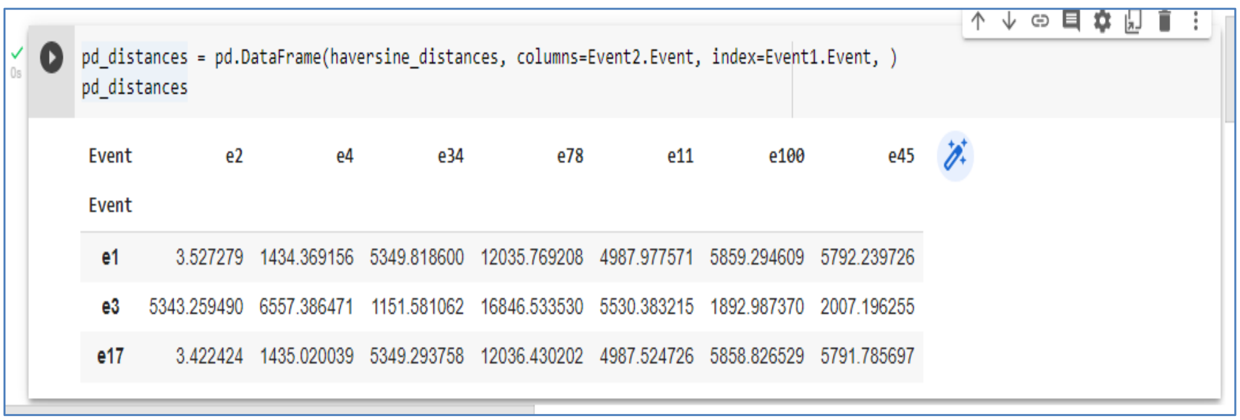


Figure 6.5: Illustrates distance scores between two events using four different distance metrics.

As can be seen in Figure 6.5, all distance metrics returned approximately the same values for the given two event pairs. Such results lead us to use one of the distance metrics, i.e., Haversine. After selecting distance matrix, we generated a pairwise distance matrix as described in Figure 6.6.



```
pd_distances = pd.DataFrame(haversine_distances, columns=Event2.Event, index=Event1.Event, )
pd_distances
```

Event	e2	e4	e34	e78	e11	e100	e45
Event							
e1	3.527279	1434.369156	5349.818600	12035.769208	4987.977571	5859.294609	5792.239726
e3	5343.259490	6557.386471	1151.581062	16846.533530	5530.383215	1892.987370	2007.196255
e17	3.422424	1435.020039	5349.293758	12036.430202	4987.524726	5858.826529	5791.785697

Figure 6.6: Snapshot of distance matrix function.

This pairwise function takes radians and outputs the Haversine distance between two events. (see results in Table 6.3).

Table 6.3: Samples of distance-based spatial relationships between events.

$Event_i$	$Event_j$	Distance(Km)	S_rel: <100km (S_near) or >100km (S_far)
e1	e8	3.52	S_near(e1, e8)
e8	e2	1434.36	S_far(e8, e2)

e1	e10	3.42	S_near(e1, e10)
e10	e7	5791.78	S-far(e10, e7)

In summary, two events $(\mathcal{E}_1, \mathcal{E}_2)$ are considered spatially close if their distance is less than user-defined distance threshold. If their distance is greater than this threshold, they are considered spatially far apart.

Directional and Topological Spatial Relationships Experiments: In order to identify the topological relationships, we create buffers around each event, defining an area around a geographic feature like a point, line or polygon. In our experiment, our method for identifying these relationships consistently achieves an f-score of 1, indicating that the inferred topological relationships precisely match the rule sets described in Section 6.3. If there is no relationship between objects, the method returns 0 or false. Similarly, our directional relationship identifier also achieves a score of 1 or True if there is a directional relationship between objects and 0 or false otherwise. Overall, experimental results highlight some important observations that are summarized in Table 6.5.

Table 6.5: Description on topological and directional spatial relationships experimenters.

Categories	Types	Observations
Topological relationships	Meet	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; \mathcal{E}_1 \; and \; \mathcal{E}_2 \; meet \; at \; boundary \\ 0 \; indicates \; two \; evens \; not \; meet \; at \; the \; boundary \end{cases}$
	Inside	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; that \; the \; \mathcal{E}_2 \; spatial \; value \; lies \; inside \; \mathcal{E}_1 \\ 0 \; indicates \; that \; there \; is \; no \; \mathcal{E}_2 \; spatila \; value \; inside \; \mathcal{E}_1 \end{cases}$
	Overlap	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; that \; the \; \mathcal{E}_1 \; spatial \; value \; partly \; covers \; from \; \mathcal{E}_2 \\ 0 \; indicates \; that \; two \; evens \; are \; not \; partly \; overlay \; each \; other \end{cases}$
	Equal	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; \mathcal{E}_1 \; and \; \mathcal{E}_2 \; have \; identical \; spatial \; values \\ 0 \; indicates \; that \; they \; don't \; have \; identical \; spatial \; values \end{cases}$
	Disjoint	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; that \; there \; is \; no \; intersection \; area \; between \; \mathcal{E}_1 \; and \; \mathcal{E}_2 \\ 0 \; indicates \; that \; they \; intersect \; each \; other \end{cases}$
Directional relationships	North	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; location \; \mathcal{E}_2 \; is \; north \; of \; \mathcal{E}_1 \\ 0 \; indicates \; that \; \mathcal{E}_2 \; is \; not \; north \; of \; \mathcal{E}_1 \end{cases}$
	South	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; location \; \mathcal{E}_2 \; is \; south \; of \; \mathcal{E}_1 \\ 0 \; indicates \; that \; \mathcal{E}_2 \; is \; not \; south \; of \; \mathcal{E}_1 \end{cases}$
	East	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; location \; \mathcal{E}_2 \; is \; east \; of \; \mathcal{E}_1 \\ 0 \; indicates \; that \; \mathcal{E}_2 \; is \; not \; east \; of \; \mathcal{E}_1 \end{cases}$
	West	$\text{pairs}(S(\mathcal{E}_1), S(\mathcal{E}_2)) = \begin{cases} 1 \; indicates \; location \; \mathcal{E}_2 \; is \; west \; of \; \mathcal{E}_1 \\ 0 \; indicates \; that \; \mathcal{E}_2 \; is \; not \; west \; of \; \mathcal{E}_1 \end{cases}$

6.5.4 Experimental Results on Causal Relationship Identification

As discussed in Section 6.4.3, the causality identifier in our study utilizes the textual information within multimedia objects. We extract the textual dimensions from these objects and treat them as a series of sentences. These sentences are then fed into a causal discovery algorithm which generates an event causal structure expressed as $\mathcal{E}_1$ causes $\mathcal{E}_2$ or $\mathcal{E}_2$ causes $\mathcal{E}_1$. This structure can be obtained from a single textual description (*such as* "*heavy rain* ($\mathcal{E}_1$) *causes flash floods* ($\mathcal{E}_2$)") or from two separate textual descriptions of event posts (e.g., "rainfall post ($\mathcal{E}_1$) vs. flood post ($\mathcal{E}_2$)"). To evaluate our causal identifier performance, we used causal corpora developed by [23] and achieved an f-score of 78.37%.

6.6 Summary

This chapter describes an algorithm that identifies various relationships between multimedia events. The algorithm was evaluated using two of our datasets and an additional pre-existing dataset. Finally, we measured the performance of the algorithm using the F-score evaluation parameter. Table 6.6 summarizes the main categories of semantic relations and the types of relations under these main categories.

Table 6.6: Main event relationship categories and their type of relations

Relationship categories	Relationship Type Description
Temporal	Starts, before, finishes, after, during, overlaps, and equal.
Spatial	Spatial can be viewed as follows: • Topological: *meet, disjoint, overlaps, within, and equal*; • Directional: *south, north, west, and east;* • Distance-based: *far and near*
Semantic	Direct causal relationship

CHAPTER 7

STAGE V&VI: EVENT KNOWLEDGE GRAPH CONSTRUCTION AND RETRIEVAL

In the previous two chapters, Stage III and Stage IV produced tangible multimedia events and diverse forms of relationships, respectively. The next challenge is to build an Event Knowledge Graph (EKG) for retrieving events. To achieve this, in **Stage V**, we first explained how we created EKG based on dimensioned graph model. Subsequently, we demonstrated how to employ the results of **Stage III** as nodes and the results of **Stage IV** as links between nodes. This chapter also elaborates on the graph database utilizing for storing nodes and their links. Finally, we present some experiment results on event retrieval using the search API.

7.1 Introduction

Event Knowledge graph is a graph model used to model and store real-world events and their relationships. All of these interlinked events can be described using graph topology [**61**]. Graph topology refers to how nodes and edges are organized in a knowledge graph. To develop a deeper understanding of the Event Knowledge Graph, it is important to define several common terminologies that are frequently used. Here are some of the fundamental terms that we will use throughout this chapter:

- **Knowledge:** is a collection of interlinked facts about events that can be used to answer queries. In other words, knowledge consists of a collection of event information (event + event dimension + relationship) that can be used to answer queries.
- **Graph:** In mathematics, it is a data structure that represents a graph of nodes and edges. Graphs can be viewed in three basic forms: 1) undirected, 2) directed, and 3) weighted graph. This study adopts the directed graph, in which each edge will be uni-directional.
- **Property Graph:** It is one of graph models allowing nodes and edges to have properties associated with them. Please note that this study uses dimensions (or metadata) as properties.

- **Node:** is a distinct point in the knowledge graph that represents a real-world event along with its dimensions (5W1H).
- **Edge:** is a link between nodes/events in the knowledge graph. Edge can be represented in three ways: 1) weighted edge that indicates the strength of each link between events; 2) multi-edge, which refers to situation where two events in a graph are connected by more than one edge; and 3) hyperedge, is an edge that connects any number of nodes.
- **Knowledge Graph Database:** is a type of database designed to represent and store knowledge in a structured way that enables efficient information retrieval system.
- **Knowledge Graph Query:** is a language used to query knowledge graph database.

Researchers in the field of social events have begun to work on building event-based knowledge graphs. While a few researchers have already created event knowledge graphs, as seen in studies [**81**, **83**, **106**], they have focused on a pairwise graph model where trigger words or entities indicating events are represented as nodes, and edges connect only two events. This pairwise graph model is inappropriate for representing our complex events, which contain multimedia contents, multi-dimensions, and divers relationships between events. Additionally, events can be linked by different types of relationships, such as spatial, temporal, and causal relationships, and some event relationships can connect more than two events (nodes). Due to these event properties, the pairwise approach is not appropriate for representing complex events. To overcome this, we used a property graph formulation to construct our Event Knowledge Graph (EKG), in which nodes and edges can have properties to further describe them. Our graph also contains hyperedges that can connect any number of events.

In summary, the procedure of constructing EKG and event retrieval from it consists of six stages as described in Chapter 4. The first four steps have already been covered in the previous Chapters. The rest of the last two steps will be discussed in the next sections of this chapter.

7.2 Event Knowledge Graph

Before describing our graph construction process, it is important to recall the definition of an Event Knowledge Graph (EKG) described in Chapter 4. Following this definition, we here constructed directed dimensioned graph data model (see Figure 7.1) where nodes and edges have associated dimensions offering more details about the events as well as their relationships. Hyperedges, which can link a number of nodes and two or three different relationships that may exist in parallel with two events, are also included. In general, EKG contains nodes, edges, and properties/dimensions, all of which are used to represent/store event data and can be defined as follows:

- **Nodes:** represent different events (*e.g., Concert, Timkat, Meskal celebration, Eid al- Fitr, and great Ethiopian run and others*). Note that nodes can be labeled and can contain multiple labels at once to identify what kind of nodes they are.
- **Properties:** are metadata (dimensions) associated with each multimedia event, such as *location, time, description, and participant*.
- **Relationships:** are used to relate nodes by type (*e.g., spatial, temporal, and semantic*) and direction (*e.g., unidirectional*).

Figure 7.1: An example of dimensioned knowledge graph with two events

7.3 Constructing an Event Knowledge Graph

The purpose of this step is to implement the event knowledge graph. This involves performing various tasks such as, selecting an appropriate graph database, storing events and their relations in the graph database, interacting with the event knowledge graph using Cypher graph query language, and developing a search API for searching relevant events. The subsequent sections describe each of these tasks.

7.3.1 Selecting Graph Database and Graph Model

To create our event knowledge graph, we need a graph database that captures all events and their relationships. Thus, selecting an appropriate graph database is a crucial first step in constructing an event knowledge graph and requires several selection criteria, including but not limited to functionality, query execution speed, scalability, easy-of-use, and others. Furthermore, it is essential to specify the graph data model to represent the event data as a knowledge graph. Graph data models can be divided into two categories [**107**]: *RDF graph model and property graph model*. These two graph data models provide ways to represent and graphically visualize linked data. But, in terms of their representation format and query langue, these two graph data models are different. For example, to represent data, RDF uses triples (subject-predicate-object) format and the property graph *(*PG) model uses nodes, edges, and properties. Another difference between RDF and PG are their querying languages. SPARQL is the language used for querying RDF data, while Cypher is the language used for accessing data in PG.

Moreover, there are popular graph databases that support the property graph model, such as IBM Graph[19], Sparksee[20], Neo4j[21], Titan[22], Amazon Neptune[23], ArangoDB[24], and OrientDB[25]. On the other hand, AllegroGraph[26], GraphDB[27], Stardog[28], Blazegraph[29], and Apache Jena[30] are some commonly used graph database in RDF graph model.

In contrast to RDF, the nodes and edges in the property graph (PG) model have properties that provide more details about the entities and their relationships without the need to create additional nodes. In general, the common limitations of RDF graph model includes: 1) RDF lacks an internal data structure; 2) In RDF, each entity and entity attributes become nodes, which makes the graph more complex; 3) RDF graph model doesn't allow edges to have attributes; 4) Inability to uniquely identify relationship type and represent complex relationships (e.g., more than two different relationships that may exist between two events); and 5) user quires that traverse in RDF graph may take a long time to execute. Considering these limitations, this study adopts dimensioned property graph model and Neo4j graph database. Using Neo4j, one of the world leading graph

[19] https://developer.ibm.com/tv/ibm-graph/
[20] https://dbpedia.org/page/Sparksee_(graph_database)
[21] https://neo4j.com/
[22] https://titan.thinkaurelius.com/
[23] https://aws.amazon.com/neptune/
[24] https://www.arangodb.com/
[25] http://orientdb.org/
[26] https://allegrograph.com/
[27] https://www.ontotext.com/products/graphdb/
[28] https://www.stardog.com/
[29] https://blazegraph.com/
[30] https://jena.apache.org/

databases, we can easily represent and retrieve our complex events. Detailed description on the representation of events is presented below.

7.3.2 Loading Events and their Relationships

In the previous section, we mentioned that we choose Neo4j as our graph database for loading event data. This involved constructing a knowledge graph by storing every individual event as a node along with their properties that captured temporal, spatial, textual, and others dimensions. Additionally, we represented the relationships between events as edges, which were coded with properties that specified the type and nature of each relationship. For instance, if we consider two events ($\mathcal{E}_{10}$, $\mathcal{E}_{122}$), we can represent each of them as a node with attributes such as location (longitude and latitude coordinates), timestamp, and textual feature. To create a relationship between these two events ($\mathcal{E}_{10}$, $\mathcal{E}_{122}$), we used a directed edge between them that encoded as “topological_disjoint” relationship. This was accomplished by employing the Cypher language to create nodes and establish connections between them. Figure 7.2 provides an illustration of the code utilized to generate nodes and edges in our study.

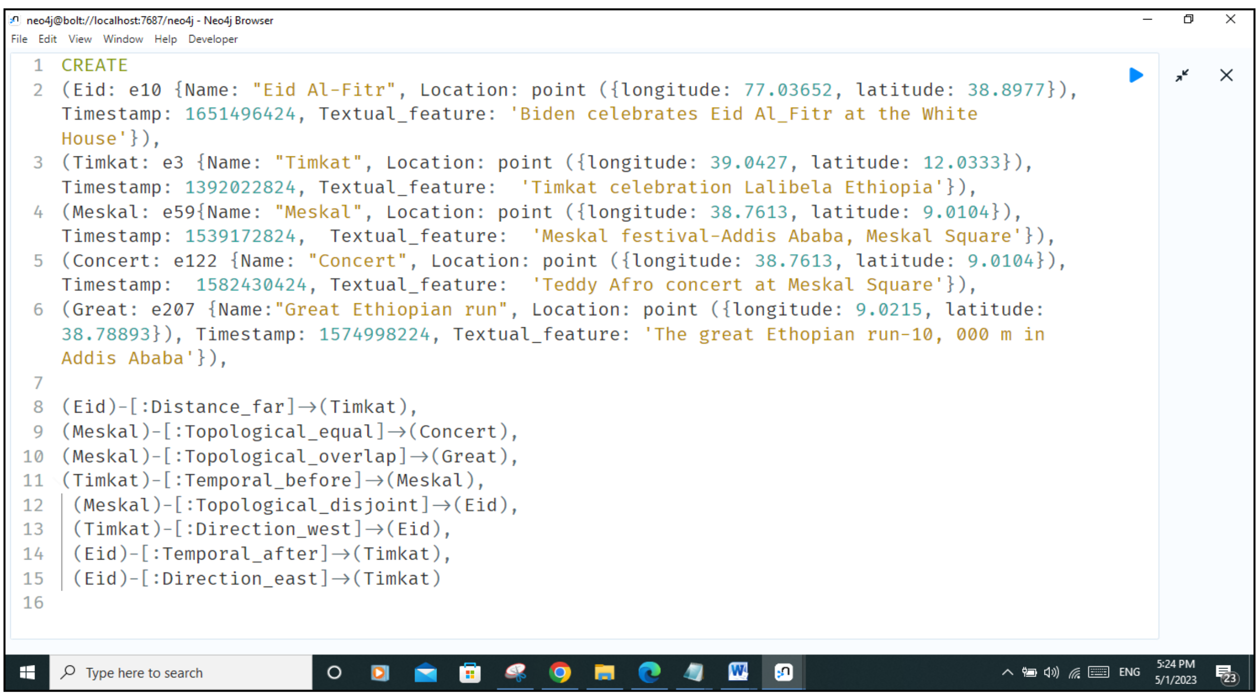


Figure 7.2: Sample snapshot of loading nodes (events) and edges (relationships) into the Neo4j graph database.

In the above Cypher code, five new nodes are created using 'CREATE' clause: '*Timkat*', '*Great Ethiopian run*', '*Eid Al-Fitr', 'Meskal*', and '*Concert*' with their dimensions. The code then creates the relationships between these five nodes, such as Distance_far, Topological_equal, Topological_overlap, Temporal_before, Topological_disjoint, Direction_west, Temporal_after, and Direction_east, as shown in Figure 7.3.

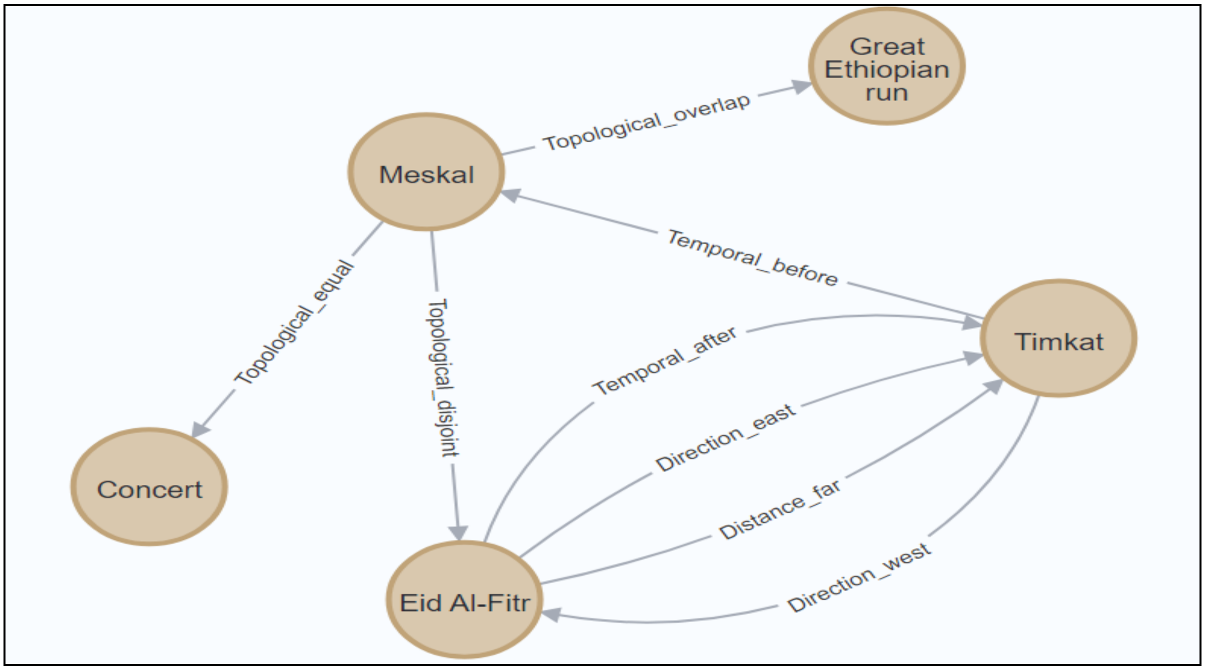


Figure 7.3: Sample snapshot of Event Knowledge Graph.

7.3.3 Using Graph Query Language

After loading all relevant information regarding events into the graph database, Graph DB query languages (*such as GraphQL, Gremlin, Cypher, and SPARQL*) is required to retrieve information related to events that have been modeled as an event knowledge graph. This study uses Cypher, which is the Neo4j query language and widely used and recognized as an open graph query language for property graph structures. An example of a Cypher graph query that retrieves node information is shown in Figure 7.4.

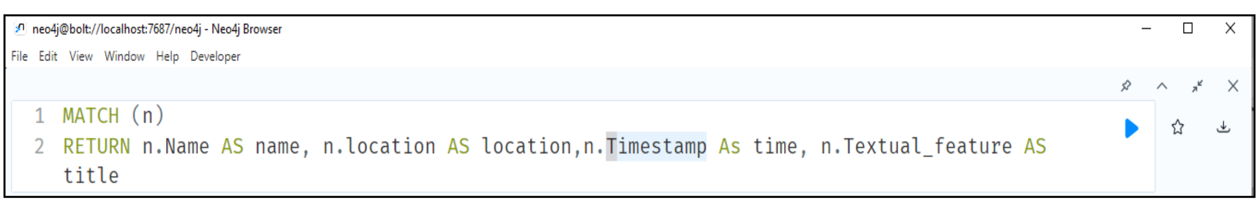
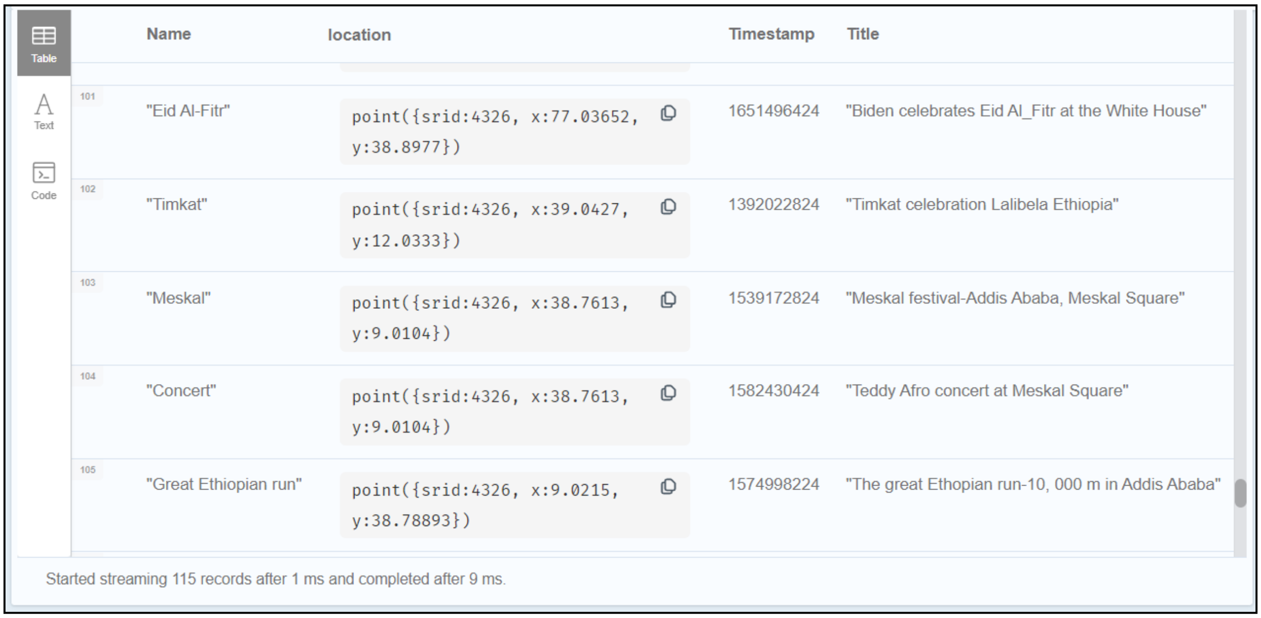


Figure 7.4: Sample snapshot of Cypher query that retrieves nodes details from the knowledge graph.

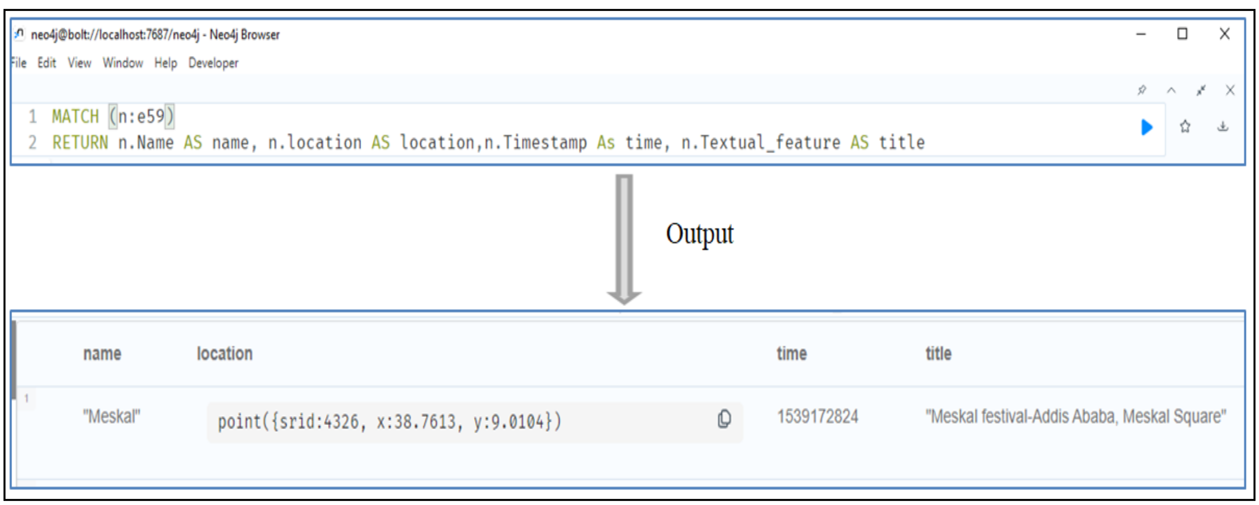
The above code can be used to retrieve all nodes with their details in the graph database, as shown in Figure 7.5.



	Name	location	Timestamp	Title
101	"Eid Al-Fitr"	point({srid:4326, x:77.03652, y:38.8977})	1651496424	"Biden celebrates Eid Al_Fitr at the White House"
102	"Timkat"	point({srid:4326, x:39.0427, y:12.0333})	1392022824	"Timkat celebration Lalibela Ethiopia"
103	"Meskal"	point({srid:4326, x:38.7613, y:9.0104})	1539172824	"Meskal festival-Addis Ababa, Meskal Square"
104	"Concert"	point({srid:4326, x:38.7613, y:9.0104})	1582430424	"Teddy Afro concert at Meskal Square"
105	"Great Ethiopian run"	point({srid:4326, x:9.0215, y:38.78893})	1574998224	"The great Ethopian run-10, 000 m in Addis Ababa"

Figure 7.5: Sample snapshot result of the query used to retrieve all nodes with their details.

Rather than retrieving all the nodes present in the graph database, it is possible to retrieve specific nodes based on node ID or node name. Presented below is an example of the query implemented to retrieve the details of a single node by utilizing its node ID.




```
MATCH (n:e59)
RETURN n.Name AS name, n.location AS location,n.Timestamp As time, n.Textual_feature AS title
```

	name	location	time	title
1	"Meskal"	point({srid:4326, x:38.7613, y:9.0104})	1539172824	"Meskal festival-Addis Ababa, Meskal Square"

Figure 7.6: A sample screenshot of the query used for retrieving details for a single node (event).

In addition, the Cypher language can be used to retrieve a variety of relationships among events in the graph database. For example, the query presented in Figure 7.7 applies the "*MATCH*" clause to find the node named “*Eid Al-Fiter*”. Then, it looks for the relationships like “*Temporal_after*”, “*Distance_far*”, and “*Direction_east*” that are connected with the node named “*Timkat*”.

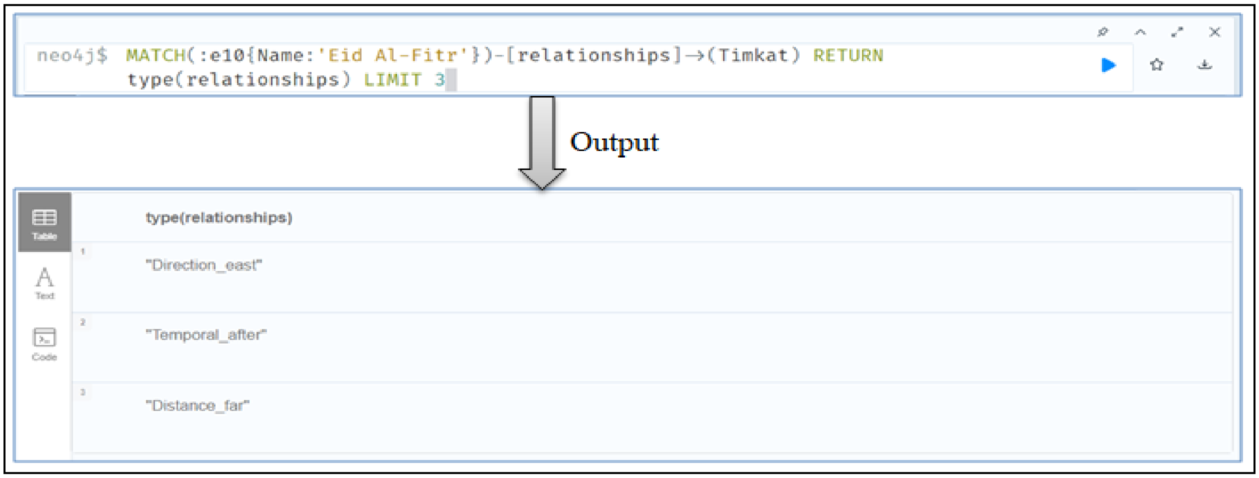


Figure 7.7: An example screenshot of the query used for retrieving different kinds of relationships between two nodes (events).

7.3.4 Building Search API for Retrieving Events

Knowledge graph-based search methods are superior to typical search methods because they are able to produce highly specific and accurate results. Such kinds of search methods are able to capture the semantics (meaning) of the search queries. As a result, they are commonly referred to as "semantic search". In this study, our semantic search API uses the event knowledge graph DB and search queries as inputs. Here are the phases for how the event search API retrieves events from its knowledge graph DB. In the first phase, the search API accepts a search query. The API then sends the search query to the graph DB. Finally, once a relevant event has been identified, it is retrieved along with a brief summary containing details, such as spatial, temporal, and textual information. Moreover, it utilizes indexing and PageRank Algorithms (for PageRank, see Section 7.4.1) to facilitate the retrieval of relevant events for users. We have used the “CREATE INDEX ON” statement to build indexes based on dimensions (or metadata) attached to each event. Once indexed, the event search API can use them to quickly retrieve relevant results for users. Figure 7.8 presents an example of an event search API output for a particular input query such as “Timkat”.

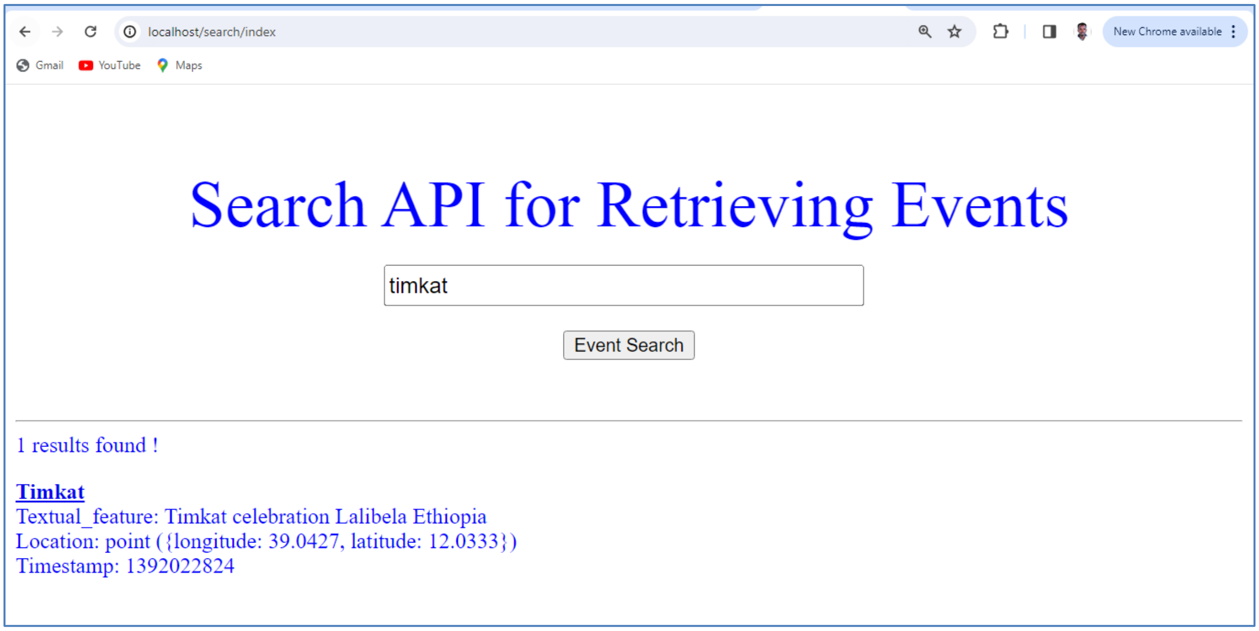


Figure 7.8: Snapshot of an event search using the event search API.

Figure 7.8 shows the process of retrieving events using node names. Additionally, integrating geographic data into the Search API enables it to provide relevant results to users, particularly when users are searching for events associated with a specific geographic location. For example, if the user inputs the search query “Meskal Square”, the system produces two events that occurred at that location. This implies that the retrieved events share a common location, shown in Figure 7.9.

Figure 7.9: Snapshot of an event search using spatial data (e.g., Meskal Square).

As shown in Figure 7.10, the Event Search API might return "*Sorry, no matching results found*" for a particular search query due to various reasons, such as spelling errors, ambiguous search terms, and absence of results corresponding to the search terms.

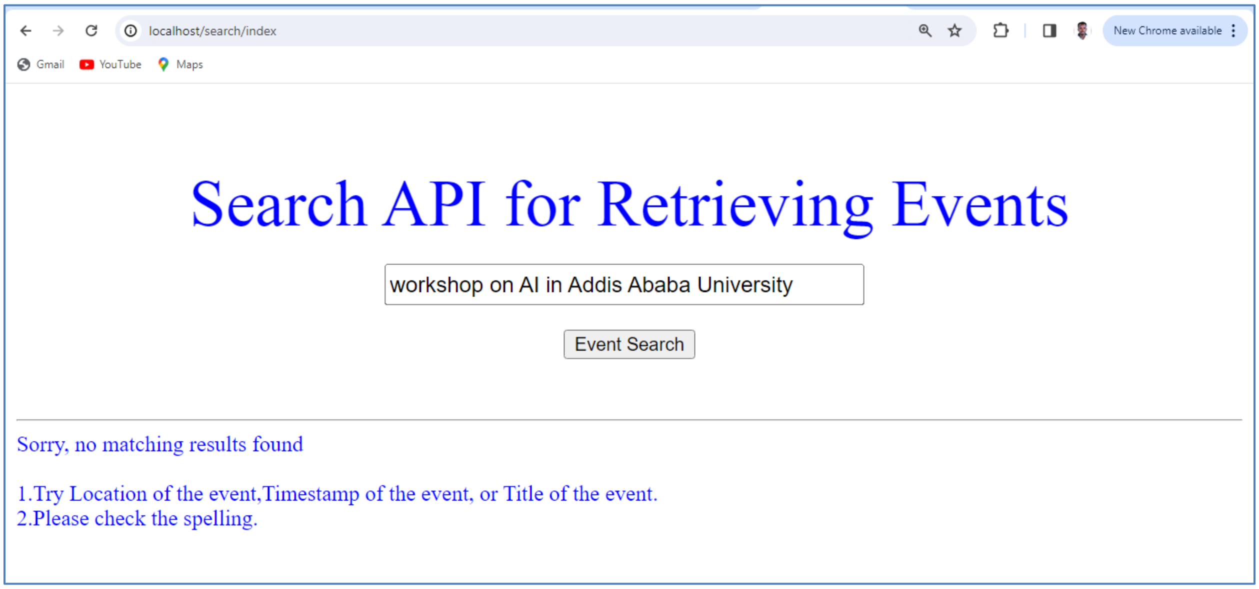


Figure 7.10: Snapshot for no matching results found.

7.4 Experimental Results

In this part, we conducted a series of three distinct experiments aimed at assessing the efficiency of our event search API. These include PageRank, similar node retrieval, and relevance of search results. Each of the experiments conducted is explained in further detail in the subsequent sub-sections.

7.4.1 PageRank

PageRank is a common algorithm used by search engines (*e.g., Goggle, Yahoo, Ask, and others*) to rank websites in their search results [**108**]. Neo4j, a graph database system, also uses the PageRank algorithm to measures the significance of nodes in the knowledge graph by measuring the number and quality of incoming edges to a node. To run the PageRank algorithm in Neo4j, we used the event knowledge graph presented in Figure 7.3. Once such a knowledge graph has been set, the next activities are: 1) defining some parameters (*such as dampingFactor, tolerance, and maxIterations*) that influence the algorithm results; and 2) running the PageRank algorithm. Our PageRank algorithm adopts an iterative approach to measures the significance of nodes, which entails performing computations iteratively until a predetermined maximum number of iteration (or *maxIterations)* is achieved. During each computation, we apply *dampingFactor = 0.85*, *tolerance = 0.1*, and an iterative approach, in which we incrementally increase the iterations by 1: starting from 1 for the first computation, then 2 for the second, 3 for the third and so on, up to its upper bound (20). Based on this approach, we conducted several experiments and obtained results. Note that only four experimental results that use maxIterations of 1, 5, 10 and 20 are presented in Table 7.1.

Table 7.1: PageRank Scores

Node	maxIterations: 1 dampingFactor: 0.85 tolerance: 0.1	maxIterations: 5 dampingFactor: 0.85 tolerance: 0.1	maxIterations: 10 dampingFactor: 0.85 tolerance: 0.1	maxIterations: 20 dampingFactor: 0.85 tolerance: 0.1
e3	0.15	0.27	0.36	0.55
e10	0.15	0.25	0.31	0.49
e59	0.15	0.21	0.26	0.28
e122	0.15	0.19	0.19	0.25
e207	0.15	0.19	0.19	0.25

The results presented in Table 7.1 reveal two primary observations: 1) the scores of nodes are influenced by the number of incoming edges it obtains. For example, as depicted in Figure 7.3, node (e3) for "Timkat" exhibits a high score against other nodes since it has three distinct incoming edges; and 2) both e122 and e207 received an equivalent experimental score, since they only have one incoming edge.

7.4.2 Retrieving the Most Similar Nodes (Events)

We evaluate the effectiveness of the event knowledge graph from an information retrieval perspective by conducting two essential tasks: measuring similarities between vertices (events) and retrieving the most similar vertices in the graph. Vertex similarity strategies can be described in three different manners: *1)* vertices that have comparable structure or role tends to have similar representations; *2)* vertices that have highest edge weight scores tend to be closer to each other; and *3)* vertices/nodes that share the same dimensions or properties with other vertices/nodes are considered to be similar to each other. This study uses node (or vertex) similarity strategy that utilizes dimensions over events.

Once node similarity strategy has been determined, the following step is to choose a similarity approach. There are three existing similarity approaches: 1) similarity based on adjacency; 2) similarity based on multi-hop; and 3) similarity based on random walks. However, the first two approaches are computationally expensive with a runtime complexity of O (|V|2) as they require iterating over all pairs of nodes. Therefore, we adopt random-walk approach, which is more efficient since it only considers nodes visited by a walker. During a random walk, nodes and their dimensions are used as input to generate a sequence of nodes, or corpus. Specifically, for each node (V) along with their dimensions(D), a random walk starts at a random node and then moves to the next node to build a corpus (C) containing a sequence of nods. For instance, our random walk operates in the following way: given nodes with dimensions $[V1\ (d1, d2, d3)$, $V2\ (d1, d2, d3), V3\ (d1, d2, d3), and\ V4\ (d1, d2, d3)]$, with walk length l = 4. To create node sequences that contain relevant information about the dimension of nodes, our algorithm starts from vertex $(V1)$, where the next node is randomly chosen from the adjacency nodes and ends at vertex $(V4)$. For creating node representations for each node in the knowledge graph, we adopt a deep learning-based auto-encoder model. The output of this model is used as input for the node similarity task. Next, we use node similarity functions to measure the similarity scores between nodes and retrieve the top 10 nodes with the highest similarity scores (refer to Table 7.2). The idea is to compute the average similarity value of the given node to the average similarity values for each node within the knowledge graph. The average function converts individual dimensional similarity values into an average value. Results presented in Table 7.2 indicate that the average value in a single node depends on the contribution of each dimension. Overall, the node with the highest similarity value is ranked first and continues until we reach its upper bound node (10). For example, we retrieve the top 10 most similar events (nodes) for “e56”. Our approach effectively retrieves similar nodes with their corresponding similarity values.

Table 7.2: 10 most similar nodes for “**e56**”.

Node Pair	Similarity Values	Node Pair	Similarity Values
(e**56**, e17)	0.99518	(e**56**, e23)	0.99375
(e**56**, e5)	0.99491	(e**56**, e66)	0.99332
(e**56**, e105)	0.99414	(e**56**, e77)	0.99263
(e**56**, e124)	0.99404	(e**56**, e134)	0.99260
(e**56**, e85)	0.99400	(e**56**, e11)	0.99258

7.4.3 Relevance of Search Results

In this part, we evaluate the relevance of the search results in the Event Knowledge Graph (EKG) utilizing f-score, precision, and recall metrics. This was done in five steps: 1) defining a set of relevant nodes (or events) that we expect to be retrieved by a query; 2) creating search queries, which were separated into three types: *textual-based queries, spatial-based queries, and temporal-based queries*. For each category, 5 queries were created; 3) running a query in the knowledge graph; 4) measuring evaluation metrics values; and 5) examining the outcomes and fine-tuning the query to improve the quality of the presented approach. For this, F-score, precision (P), and recall (R) metrics can be computed as:

$$P = \frac{\text{Number of relevant retrieved nodes}}{\text{Total number of reterieved nodes}} \tag{7.1}$$

$$R = \frac{\text{Number of relevant retrieved nodes}}{\text{Total number of relevant nodes}} \tag{7.2}$$

$$F - \text{score} = 2 * \frac{P*R}{P+R} \tag{7.3}$$

Let us see the above formulas with example, suppose a user searches for "Q_1", and the search API returns 75 nodes, out of which 67 are relevant to the query. In this cause, the precision score is 89% (67/75). This means that only 89% of the total results returned by the search API were actually relevant to the user query. Please not that there were 78 relevant nodes in the dataset that satisfy the query criteria, and then the recall is 85% (67/78). To summarize, results in textual-based queries are presented in Figure 7.11.

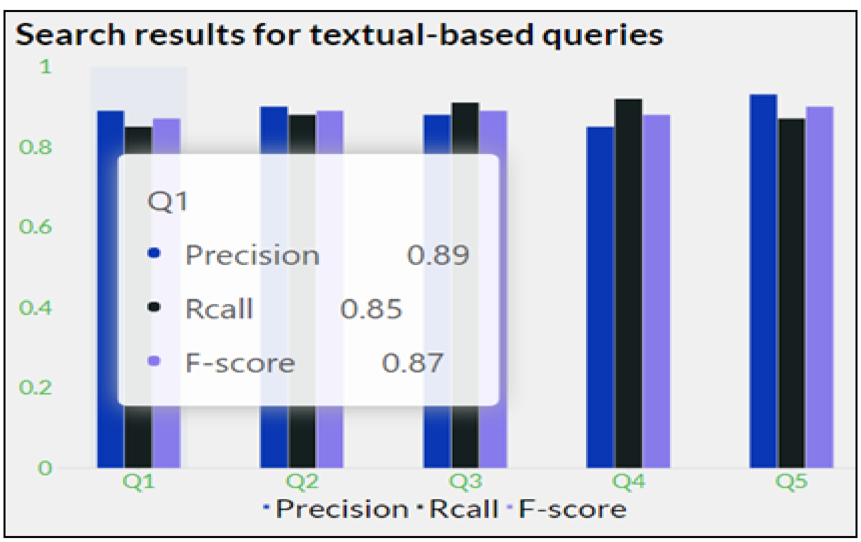


Figure 7.11: Precision, Recall, and F-score results for textual-based 5 queries (similar bar charts can be gained with reference to spatial-based queries and temporal-based queries).

7.5 Summary

This chapter demonstrated how the Event Knowledge Graph (EKG) and its search API have developed. The effectiveness of the developed EKG was evaluated through its downstream tasks, which include retrieving nodes that are most similar to each other. In addition, this chapter presented experimental findings pertaining to both PageRank and the significance of search results.

CHAPTER 8

CONCLUSION AND FUTURE WORKS

7.1 Introduction

The general objective of this research was to demonstrate the use of the 5W1H approach for detecting, linking, representing and retrieving semantically rich and multi-dimensioned events. To this end, we have developed different kinds of approaches (i.e., *detecting, linking, representing, and retrieving events*) by which we demonstrated the possibility of retrieving multi-dimensioned events in social media ecosystem. The remaining section of the chapter is focused on presenting our conclusion and recommendations for future works.

7.2 Conclusion

This research presents a generic framework for retrieving semantically rich multimedia events using knowledge graph in social media ecosystem. The framework consists of six basic stages. Here, we focused on **Stage III, IV, V, and VI. Stage III** handles the identification of multimedia events from represented multimedia data in Stage II. This stage also contains sub modules, such as similarity computation, cluster comparison strategies, and event representation. **Stage IV** identifies semantic relationships between multimedia events. The stage takes represented real multimedia events along with their dimensions in Stage III and produces three main categories of semantic relations (such as semantic, spatial, and temporal) and the types of relations under these main categories by computing dimensions over events. Each of these relationships contains their own sub-relationships. For example, event temporal relationships can be seen as before, after, starts, etc. **Stage V** performs event knowledge graph construction task using the results of Stage III and Stage IV. The stage also contains sub module, such as graph database which used to store nodes and edges in our graph**. Stage VI** uses event search API, Event Knowledge Graph (EKG) created in Stage V, and query languages (e.g., Cypher) for retrieving semantically rich and multi-dimensioned events. In each of these stages, the algorithm developed, the dataset, the experimentation, the results obtained, and discussion of the results are presented.

7.2 Revisiting the Research Questions

This section revisits the research questions formulated in Chapter 1. The goal is to provide a summary of how these research questions were addressed in this study. Each question was addressed as follows:

1. **How can the 5W1H approach be applied to multimedia event-related problems or tasks?**

In order to answer this question, we applied the 5W1H approach to (1) represent multimedia data, (2) identify the key elements of an event, (3) detect events, (4) identify different kinds of semantics relationships, (5) to provide additional information about the nodes in the Event Knowledge Graph (EKG), and (6) to describe event search results in temporal, spatial, and textual perspective (see Chapter 5, 6, and 7 for more details).

2. **What are the key components of 5W1H event management framework?**

To address this question, a novel 5W1H event management framework is introduced consisting of six interacting components: 1) social media data crawling and preprocessing; 2) social media data representation; 3) social event detection; 4) semantic relationship identification; 5) Event KG construction; and 6) event search API. These all components and their responsibilities are outlined in Chapter 4.

3. **How can real-world multimedia events and their different types of semantic relationships be identified?**

To answer this question, a detailed literature review on event detection and event relationship identification was carried out. From this point of view, various research works were assessed on the basis of the sources of data and methods used for detecting events and their relationships (see Chapter 3). The outcome of the literature review revealed three key findings: 1) researchers primarily concentrate on detecting events in an unscalable manner; 2) current event detection methods emphasize two/three aspects of the 5W1H dimensions; 3) prior studies related to events did not take into account the identification of various types of semantic relationships between events. To address these, scalable event detection and semantic event relationship identification algorithms are introduced in this study (see Chapter 5 and 6). A scalable event detection approach can addressed the continuous growth of social media data, whereas the event relationship identification approach can identifies naturally different kinds of relationships between events. So, research question 3 has been answered in this study.

4. **What type of knowledge graph database and knowledge graph model should be constructed to represent multi-links, multi-dimensions, and multi-sources events?**

To answer this question, this study adopts property graph model and Neo4j graph database. Using Neo4j, one of the world leading graph databases, we can easily represent and retrieve our complex events. Property graph model allows nodes and edges to have properties related to them. Please note that this study uses dimensions (or metadata) as properties.

5. **How can retrieve semantically meaningful multi-dimensional events?**

In order to respond this question, we developed an event knowledge graph (EKG) and a well-defined event search engine. A search engine incorporating a well-defined event knowledge graph that captures events and their relationships outperforms conventional search engines. Our event search API uses EKG and user queries as input and produces semantically meaningful multi-dimensional events as output. Moreover, search engine that use Knowledge Graph is known "semantic search," as they understand the meaning (or semantic) of search queries.

7.3 Future Work

Based on our research findings and the knowledge obtained from the relevant literatures, we plan to continue working on the following aspects.

- **Unified Data Model for Representing Social Media Data:** Researchers in event detection have developed their own data models to represent social media data, but there is still no cohesive data model that represents all aspects of social media data. The lack of a unified data model makes it difficult to manage, design, and represent social media data. Therefore, it is crucial for future research in event detection to center on developing a unified data model that can fully support the representation of social media data
- **Semantic Aspect of Social Media Data Representation:** Our experiment findings have showed that the presence of semantic dimension leads to a higher quality event detection

model. In event detect, semantic representation is a crucial component for capturing the meaning within textual features (or events). But, very little has been done to represent the semantic of textual features for measuring the semantic similarity between two concepts. To address this, this study uses the semantic network based on WordNet to represent only the semantic aspect of textual features (What), while neglecting other semantic dimensions of events such as reason (Why) and method (How). Incorporating these two dimensions is essential not only for event detection and event relation identification, but also for creating a semantically rich Event Knowledge Graph. Therefore, future research should focus on representing the semantics of textual features and utilizing them in event-related tasks.

- **Multi-media/dimension/domain Aware Event Detection:** In general, social media data are multi-domain (*as it is generated and utilized within different domain, such as personal, entertainment, politics, natural, and more*), multi-media (*referring to different media types, such as images, videos, text, etc*.), and multi-dimensional (*referring to metadata associated to each multimedia content*). Taking into account all these characteristics of social media data in event detection can lead to improved performance. Moreover, event detection algorithms that are limited to closed domains are not enough. In such situations, an open-domain event detection approach that takes into account the characteristics of social media documents is expected.
- **Comprehensive Social Event Detection:** This study utilized only four even dimensions as shown in Sections 3.2. This implies that dimensions like "Why" and "How" were not used in this study and their importance was not investigated. Therefore, investigating the importance of these two dimensions remains an active research topic. To put is simply, there is a need for an approach that encompasses all aspects of social events.
- **Semantic Relationship Identification between Events:** Detecting relationships between events in social media documents is more difficult than in textual sources [**33**]. This is because social media documents often contain noise, grammatical errors, and abbreviations, making them unsuitable for supervised relationship identification methods. Additionally, multi-dimensional nature of social media data creates complex structures for each type of event relationship. To create effective unsupervised event relationship identifiers, it is important to consider the properties of social media data and the diverse types of event relationships. Note that this study did not encompass all possible forms of event relationships

within each of the three primary categories of relationships. For instance, only 7 temporal relationships were extracted out of a total of 13. The same is true for spatial and semantic relationships.

- **5W1H-Aware Comprehensive Event Knowledge Graph:** Entity-based event representation has been achieved through the use of the pairwise graph representation model. However, when dealing with events that have multiple features and media, this structured format may not be effective. Furthermore, there are multiple types of relationships that can exist between two events, and there is a single relationship that might connect two or more independent events, making it difficult to illustrate using the pairwise graph model. Providing 5W1H aware knowledge graph model that can represent all aspects of the event remains an open problem. As can been seen from Chapter 7, only three event dimensions (e.g., temporal, spatial, and semantic) are used in our Event Knowledge Graph (EKG). An Event Knowledge Graph that covers all dimensions of events is anticipated.
- **Event Search API:** The event search API presented here does not include images or videos in the search results. Therefore, new search API is expected to include the above multimedia contents.

References

[1] Digital Banking Ecosystem: strategies, investments, and digital transformation in 2020, https://www.businessinsider.com/digital-banking-ecosystem-report?IR=T, last accessed 2020/11/01

[2] Serbanati, L. D., Ricci, F. L., Mercurio, G., & Vasilateanu, A. (2011). Steps towards a digital health ecosystem. *Journal of biomedical informatics*, *44*(4), 621-636.

[3] Digital Ecosystems: An Imperative for the Manufacturing Industry, https://www.logicbay.com/digital-ecosystems-for-manufacturing, last accessed 2020/11/02

[4] Suseno, Y., Laurell, C., & Sick, N. (2018). Assessing value creation in digital innovation ecosystems: A Social Media Analytics approach. *The Journal of Strategic Information Systems*, *27*(4), 335-349.

[5] Kidanu, S. A., Cardinale, Y., Chbeir, R., De Ponte, V., Figueroa, A., Rodríguez, R., & Ibanez, C. A. R. (2016, August). Mmdes: Multimedia digital ecosystem-new platform for collaboration and sharing. In *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)* (pp. 393-400). IEEE.

[6] Jain, R. (2009). Multimedia Data. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_1008.

[7] Kidanu, S. A. (2016). *Digital Ecosystem: For Better Management of Multimedia Contents* (Doctoral dissertation, Université de Pau et des Pays de l'Adour).

[8] Becker, H., Naaman, M., & Gravano, L. (2010, February). Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining* (pp. 291-300).

[9] Abebe, M. A., Tekli, J., Getahun, F., Chbeir, R., & Tekli, G. (2020). Generic metadata representation framework for social-based event detection, description, and linkage. *Knowledge-Based Systems*, *188*, 104817.

[10] Li, Q., Nourbakhsh, A., Shah, S., & Liu, X. (2017, April). Real-time novel event detection from social media. In *2017 IEEE 33Rd international conference on data engineering (ICDE)* (pp. 1129-1139). IEEE.

[11] Dong, X., Mavroeidis, D., Calabrese, F., & Frossard, P. (2015). Multiscale event detection in social media. *Data Mining and Knowledge Discovery*, *29*(5), 1374-1405. https://doi.org/10.1007/s10618-015-0421-2.

[12] Hamborg, F., Breitinger, C., & Gipp, B. (2019). Giveme5w1h: A universal system for extracting main events from news articles. *arXiv preprint arXiv:1909.02766*.

[13] Hamborg, F., Lachnit, S., Schubotz, M., Hepp, T., & Gipp, B. (2018). Giveme5W: main event retrieval from news articles by extraction of the five journalistic W questions. In *Transforming Digital Worlds: 13th International Conference, iConference 2018, Sheffield, UK, March 25-28, 2018, Proceedings 13* (pp. 356-366). Springer International Publishing.

[14] Nurdin, A., & Maulidevi, N. U. (2018, March). 5w1h information extraction with CNN-bidirectional LSTM. In *Journal of Physics: Conference Series* (Vol. 978, No. 1, p. 012078). IOP Publishing.

[15] Reuter, T., & Cimiano, P. (2012, June). Event-based classification of social media streams. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval* (pp. 1-8).

[16] Alireza Hajarian. "What is the difference and similarity between qualitative and quantitative research", Published on Dec 6, 2016.

[17] Abiy Zegeye, Alemayehu Worku, Daniel Tefera, Melese Getu and Yilma Sileshi, "Introduction to Research Methods (Preparatory module for Addis Ababa University graduate programs)," *Graduate Studies and Research Office. Addis Ababa University*, September 2009.

[18] Hevner, Alan and Chatterjee, Samir, "Design Science Research in Information Systems." Design Research in Information Systems: *Theory and Practice*, pp.9-22, June, 2010.

[19] Pfeffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. and Bragge, J, "The Design Science Research Process: A Model for Producing and Presenting Information Systems Research." *Journal of Management Information Systems*, vol. 24 Issue 3, pp. 45-77, Number 3 2008.

[20] Genemo, H., Miah, S.J. & McAndrew, "A Design Science Research Methodology for Developing a Computer-Aided Assessment Approach Using Method Marking Concept," *Journal of Education and Information Technology*, vol. 21, Issue 6, pp. 1769–1784, November 2016.

[21] Geerts, Guido L., "A design science research methodology and its application to accounting information systems research," *International Journal of Accounting Information Systems, Elsevier*, vol. 12(2), pp. 142-151, 2011.

[22] Sutanto, T., & Nayak, R. (2013, October). ADMRG@ MediaEval 2013 Social Event Detection. *Proceedings of the MediaEval 2013 Multimedia Benchmark Workshop [CEUR Workshop Proceedings, Volume, 1043]*. Sun SITE Central Europe, http://www.ceur-ws.org/, pp. 1-2.

[23] Kayesh, H., Islam, M., & Wang, J. (2019). On event causality detection in tweets. *arXiv preprint arXiv:1901.03526*. https://doi.org/10.48550/arXiv.1901.03526.

[24] Mulegeta, W., (2015). Machine Learning of Complex Morphology: the Case of Amharic Verbs. Dissertation, Addis Ababa University.

[25] Pomp, A., Nitsch, V., & Meisen, T. (2020). *Bottom-up knowledge graph-based data management* (No. RWTH-2020-09733). Lehrstuhl für Informationsmanagement im Maschinenbau.

[26] Kidanu, S. A., Cardinale, Y., Tekli, G., & Chbeir, R. (2015, June). A Multimedia-Oriented Digital Ecosystem: a new collaborative environment. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)* (pp. 411-416). IEEE.

[27] Abebe, M. A., Tekli, J., Getahun, F., Chbeir, R., & Tekli, G. (2017). Overview of event-based collective knowledge management in multimedia digital ecosystems. In *2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 40-49). IEEE.

[28] Abebe, M. A., Getahun, F., Asres, S., & Chbeir, R. (2015, September). Event extraction for collective knowledge in multimedia digital EcoSystem. In *AFRICON 2015* (pp. 1-5). IEEE.

[29] Ashagrie, M. (2018, February 4). Event Extraction Framework in Multimedia Digital Ecosystem. 213.55.95.56. http://213.55.95.56/handle/123456789/19369

[30] Jain, R. (2009). Multimedia Data. In: LIU, L., ÖZSU, M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-39940-9_1008.

[31] Dou, W., Wang, X., Ribarsky, W., & Zhou, M. (2012, October). Event detection in social media data. In *IEEE VisWeek workshop on interactive visual text analytics-task driven analytics of social media content* (pp. 971-980).

[32] NIST, *2011 TRECVID Multimedia Event Detection Evaluation Plan - Nist.gov*. (n.d.). Retrieved September 3, 2022.

[33] Raad, E. J., & Chbeir, R. (2014, December). Foto2events: From photos to event discovery and linking in online social networks. In *2014 IEEE Fourth International Conference on Big Data and Cloud Computing* (pp. 508-515). IEEE.

[34] Dou, W., Wang, X., Ribarsky, W., & Zhou, M. (2012, October). Event detection in social media data. In *IEEE VisWeek workshop on interactive visual text analytics-task driven analytics of social media content* (pp. 971-980).

[35] Mansour, E., Tekli, G., Arnould, P., Chbeir, R., & Cardinale, Y. (2017, August). F-SED: feature-centric social event detection. In *International Conference on Database and Expert Systems Applications* (pp. 409-426). Springer, Cham.

[36] Becker, H., Naaman, M., & Gravano, L. (2011). 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, pp. 438-441).

[37] Zhou, X., & Chen, L. (2014). Event detection over twitter social media streams. *The VLDB journal*, *23*(3), 381-400. https://doi.org/10.1007/s00778-013-0320-3.

[38] Choudhury, Smitashree and Alani, Harith (2014). Personal life event detection from social media. In: Workshop Proceedings, CEUR. http://ceur-ws.org/Vol-1210/SP2014_04.pdf.

[39] Salas, A., Georgakis, P., Nwagboso, C., Ammari, A., & Petalas, I. (2017, July). Traffic event detection framework using social media. In *2017 IEEE International Conference on Smart Grid and Smart Cities (ICSGSC)* (pp. 303-307). IEEE.

[40] Zeng, K. (2021). BEHIND: a 4W-oriented Method for Event Detection from Twitter. *International Conferences on Software Engineering and Knowledge Engineering*. https://doi.org/10.18293/seke2021-092.

[41] Becker, H., Naaman, M., & Gravano, L. (2010, February). Learning similarity measures for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining* (pp. 291-300).

[42] Nguyen, T. V., Dao, M. S., Mattivi, R., Sansone, E., De Natale, F. G., & Boato, G. (2013, January). Event clustering and classification from social media: Watershed-based and kernel methods. In *MediaEval*.

[43] Jain, T., Singh, B., & Dewang, R. K. (2022). Event Detection in Live Twitter Streams Using Tf-Idf and Clustering Algorithms. In *Data, Engineering and Applications: Select Proceedings of IDEA 2021* (pp. 469-480). Singapore: Springer Nature Singapore.

[44] Team, P. (2023). K-means Clustering in Machine Learning. Python Geeks. https://pythongeeks.org/k-means-clustering-in-machine-learning/.

[45] Jarman, A. M. (2020). Hierarchical cluster analysis: Comparison of single linkage, complete linkage, average linkage and centroid linkage method. *Georgia Southern University*.

[46] Khandelwal, R. (2022, January 6). Everything on Hierarchical Clustering-Towards AI. Medium. https://pub.towardsai.net/everything-on-hierarchical-clustering-60bf613377a2.

[47] Chopade, N., & Sheetlani, J. (2017). Recent trends in incremental clustering: A review. *IOSR Journal of Computer Engineering*, *19*(1), 19-24.

[48] Jovel, J., & Greiner, R. (2021). An introduction to machine learning approaches for biomedical research. *Frontiers in Medicine*, *8*, 2534.

[49] Psallidas, F., Becker, H., Naaman, M., & Gravano, L. (2013). Effective Event Identification in Social Media. *IEEE Data Eng. Bull.*, *36*(3), 42-50.

[50] Reuter, T., Cimiano, P., Drumond, L., Buza, K., & Schmidt-Thieme, L. (2011). Scalable event-based clustering of social media via record linkage techniques. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 5, No. 1, pp. 313-320).

[51] Chopde, N. R., & Nichat, M. (2013). Landmark based shortest path detection by using A* and Haversine formula. *International Journal of Innovative Research in Computer and Communication Engineering*, *1*(2), 298-302.

[52] Gupta, I., Gautam, K., & Chandramouli, K. (2013, October). VIT@ MediaEval 2013 Social Event Detection Task: Semantic Structuring of Complementary Information for Clustering Events. In *MediaEval*.

[53] Fellbaum, C. (2010). WordNet. In *Theory and applications of ontology: computer applications* (pp. 231-243). Springer, Dordrecht.

[54] Zaharieva, M., Schopfhauser, D., Del Fabro, M., & Zeppelzauer, M. (2014). Clustering and Retrieval of Social Events in Flickr. *MediaEval*, *14*, 1263.

[55] Burel, G., Saif, H., Fernandez, M., & Alani, H. (2017). On semantics and deep learning for event detection in crisis situations. *In: Workshop on Semantic Deep Learning (SemDeep), at ESWC 2017*, 29 May 2017, Portoroz, Slovenia. http://semdeep.iiia.csic.es/files/SemDeep-17_paper.

[56] Kayesh, H., Islam, M. S., Wang, J., Kayes, A. S. M., & Watters, P. A. (2022). A deep learning model for mining and detecting causally related events in tweets. *Concurrency and Computation: Practice and Experience*, *34*(2), e5938.

[57] Kayesh, H., Islam, M. S., & Wang, J. (2019, December). Event causality detection in tweets by context word extension and neural networks. In *2019 20th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)* (pp. 352-357). IEEE.

[58] Khoo, C. S., & Na, J. C. (2006). Semantic relations in information science. *Annu. Rev. Inf. Sci. Technol.*, *40*(1), 157-228.

[59] Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, *26*(11), 832-843.

[60] Skiadopoulos, S. (2015). Directional Relations. Encyclopedia of GIS, 1–7. doi:10.1007/978-3-319-23519-6_1539-1

[61] Huang, Z. (2010). Link prediction based on graph topology: The predictive value of generalized clustering coefficient. *Available at SSRN 1634014*.

[62] Wood, P. T. (2012). Query languages for graph databases. *ACM Sigmod Record*, *41*(1), 50-60.

[63] Reinanda, R., Meij, E., & de Rijke, M. (2020). Knowledge graphs: An information retrieval perspective. *Foundations and Trends® in Information Retrieval*, *14*(4), 289-444.

[64] Chen, L., & Roy, A. (2009, November). Event detection from flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 523-532).

[65] Liu, X., Troncy, R., & Huet, B. (2011, November). Using social media to identify events. In *Proceedings of the 3rd ACM SIGMM international workshop on Social media* (pp. 3-8).

[66] Qian, S., Zhang, T., Xu, C., & Hossain, M. S. (2015). Social event classification via boosted multimodal supervised latent dirichlet allocation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *11*(2), 1-22.

[67] Alsaedi, N., Burnap, P., & Rana, O. F. (2014). A combined classification-clustering framework for identifying disruptive events. *In Proceedings of the 6th ASE International Conference on Social Computing* (SocialCom'14).

[68] Huang, L., Liu, G., Chen, T., Yuan, H., Shi, P., & Miao, Y. (2021). Similarity-based emergency event detection in social media. *Journal of Safety Science and Resilience*, *2*(1), 11-19.

[69] Afyouni, I., Khan, A. & Aghbari, Z.A (2022). Deep-Eware: spatio-temporal social event detection using a hybrid learning model. *J Big Data* **9**, 86 (2022). https://doi.org/10.1186/s40537-022-00636-w.

[70] Zhang, Q., Chen, M., & Liu, L. (2017, December). A review on entity relation extraction. In *2017 second international conference on mechanical, control and computer engineering (ICMCCE)* (pp. 178-183). IEEE.

[71] Newton, Casey (December 14, 2012). "How Google is taking the Knowledge Graph glob al". CBS Interactive. https://www.cnet.com/news/how-google-is-taking-the-knowledge-graph -global/. Retrieved December 10, 2017.

[72] Mendes, P. N., Jakob, M., & Bizer, C. (2012). *DBpedia: A multilingual cross-domain knowledge base* (pp. 1813-1817). European Language Resources Association (ELRA).

[73] Pellissier Tanon, T., Weikum, G., & Suchanek, F. (2020). Yago 4: A reason-able knowledge base. In *The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17* (pp. 583-596). Springer International Publishing.

[74] Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., & Vrandečić, D. (2014). Introducing wikidata to the linked data web. In *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13* (pp. 50-65). Springer International Publishing.

[75] Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33-38.

[76] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008, June). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (pp. 1247-1250).

[77] Bauer, F., & Kaltenböck, M. (2011). Linked open data: The essentials. *Edition mono/monochrom, Vienna*, *710*, 21.

[78] Gottschalk, S., & Demidova, E. (2018). Eventkg: A multilingual event-centric temporal knowledge graph. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15* (pp. 272-287). Springer International Publishing.

[79] Gottschalk, S., Kacupaj, E., Abdollahi, S., Alves, D., Amaral, G., Koutsiana, E., ... & Thakkar, G. (2023). Oekg: The open event knowledge graph. *arXiv preprint arXiv:2302.14688*.

[80] Guan, S., Cheng, X., Bai, L., Zhang, F., Li, Z., Zeng, Y., ... & Guo, J. (2022). What is event knowledge graph: a survey. *IEEE Transactions on Knowledge and Data Engineering*.

[81] Nguyen, H. L., & Jung, J. J. (2019). Social event decomposition for constructing knowledge graph. *Future Generation Computer Systems*, *100*, 10-18.

[82] Sun, X., Qi, L., Sun, H., Li, W., Zhong, C., Huang, Y., & Wang, P. (2020). Earthquake knowledge graph constructing based on social intercourse using bilstm-crf. In *IOP Conference Series: Earth and Environmental Science* (Vol. 428, No. 1, p. 012080). IOP Publishing.

[83] Yan, Z., & Tang, X. (2022, June). Hierarchical Storyline Generation Based on Event-centric Temporal Knowledge Graph. In *Knowledge and Systems Sciences: 21st International Symposium, KSS 2022, Beijing, China, June 11–12, 2022, Proceedings* (pp. 149-159). Singapore: Springer Nature Singapore.

[84] Wu, H., Liu, M., Zhang, S., Wang, Z., & Cheng, S. (2020). Big data management and analytics in scientific programming: A deep learning-based method for aspect category classification of question-answering-style reviews. *Scientific Programming*, *2020*, 1-10.

[85] Chua, T. S. (2014, November). The Multimedia Challenges in Social Media Analytics. In *Proceedings of the 3rd International Workshop on Socially-Aware Multimedia* (pp. 17-18).

[86] Kromann, F. M., & Kromann, F. M. (2018). Date and Time. *Beginning PHP and MySQL: From Novice to Professional*, 319-341.

[87] Xu, M., Zhao, C., Rojas, D. S., Thabet, A., & Ghanem, B. (2020). G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10156-10165).

[88] Panagiotou, N., Katakis, I., & Gunopulos, D. (2016). 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*, 42-84.

[89] Menon, V., Muthukrishnan, G., & Kalyani, S. (2020). Subspace clustering without knowing the number of clusters: A parameter free approach. *IEEE Transactions on Signal Processing*, *68*, 5047-5062.

[90] Jin, X., Han, J. (2011). *K*-Means Clustering. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_425.

[91] Kaski, S. (2011). Self-Organizing Maps. In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_746.

[92] Bezdek, J. C., Ehrlich, R., & Full, W. (1984). FCM: The fuzzy c-means clustering algorithm. *Computers & geosciences*, *10*(2-3), 191-203.

[93] Sanders, P., & Schulz, C. (2012). High quality graph partitioning. *Graph Partitioning and Graph Clustering*, *588*(1), 1-17.

[94] Sasirekha, K., & Baby, P. (2013). Agglomerative hierarchical clustering algorithm-a. *International Journal of Scientific and Research Publications*, *83*(3), 83.

[95] Shirkhorshidi, A. S., Aghabozorgi, S., & Wah, T. Y. (2015). A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one*, *10*(12), e0144059.

[96] Charbel, N., Tekli, J., Chbeir, R., & Tekli, G. (2015). Resolving XML Semantic Ambiguity. In *Proceedings of the 18th International Conference on Extending Database Technology, EDBT*, pp. 277–288. https://doi.org/10.5441/002/edbt.2015.25.

[97] Tekli, J., Charbel, N., & Chbeir, R. (2016). Building semantic trees from XML documents. *Journal of Web Semantics, 37*, 1–24. https://doi.org/10.1016/j.websem.2016.03.002

[98] Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(1), 86-97.

[99] Pfitzner, D., Leibbrandt, R., & Powers, D. (2009). Characterization and evaluation of similarity measures for pairs of clusterings. *Knowledge and Information Systems*, *19*, 361-394.

[100] Vinh, N. X., Epps, J., & Bailey, J. (2009, June). Information theoretic measures for clusterings comparison: is a correction for chance necessary?. In *Proceedings of the 26th annual international conference on machine learning* (pp. 1073-1080).

[101] Sripada, S. C., & Rao, M. S. (2011). Comparison of purity and entropy of k-means clustering and fuzzy c means clustering. *Indian journal of computer science and engineering*, *2*(3), 343-346.

[102] De Souto, M. C., Coelho, A. L., Faceli, K., Sakata, T. C., Bonadia, V., & Costa, I. G. (2012, October). A comparison of external clustering evaluation indices in the context of imbalanced data sets. In *2012 Brazilian Symposium on Neural Networks* (pp. 49-54). IEEE.

[103] Mirza, P. (2016). Extracting temporal and causal relations between events. *arXiv preprint arXiv:1604.08120*.

[104] Zuo, X., Cao, P., Chen, Y., Liu, K., Zhao, J., Peng, W., & Chen, Y. (2021). Improving event causality identification via self-supervised representation learning on external causal statement. *arXiv preprint arXiv:2106.01654*.

[105] Vo, D. T., Al-Obeidat, F., & Bagheri, E. (2020). Extracting temporal and causal relations based on event networks. *Information Processing & Management*, *57*(6), 102319.

[106] Yang, C. H., Jie, Y. S., Chin, B. K., Chin, L., & Ng, L. H. X. (2020). KOSMOS: Knowledge-graph Oriented Social media and Mainstream media Overview System. *arXiv preprint arXiv:2012.06209*.

[107] Fensel, D., Simsek, U., Angele, K., Huaman, E., Kärle, E., Panasiuk, O., & Wahler, A. (2020). *Knowledge graphs*. Cham: Springer International Publishing.

[108] Gleich, D. F. (2015). PageRank beyond the web. *siam REVIEW*, *57*(3), 321-363.