Leveraging Social Media in Disaster Situations

Hamada Mohamed Abdelsamee Zahera

July 30, 2023
Version: Final

Department of Computer Science
Data Science Group (DICE)

Doctoral Dissertation

Leveraging Social Media in Disaster Situations

A dissertation presented by

Hamada Mohamed Abdelsamee Zahera

to the
Faculty of Computer Science, Electrical Engineering and Mathematics
of
Paderborn University
in partial fulfillment of the requirements for the degree of
Dr. rer. nat.

1. Reviewer Prof. Dr. Axel-Cyrille Ngonga Ngomo
Department of Computer Science
Paderborn University

2. Reviewer Prof. Dr. Michael Cochez
Department of Computer Science
Vrije Universiteit Amsterdam

Supervisor Prof. Dr. Axel-Cyrille Ngonga Ngomo

July 30, 2023

Hamada Mohamed Abdelsamee Zahera
Leveraging Social Media in Disaster Situations
Doctoral Dissertation, July 30, 2023
Reviewers: Prof. Dr. Axel-Cyrille Ngonga Ngomo and Prof. Dr. Michael Cochez
Supervisors: Prof. Dr. Axel-Cyrille Ngonga Ngomo and Dr. Mohamed Ahmed Sherif
The thesis was defended on 17.07.2023 in Paderborn.

Paderborn University
Data Science Group (DICE)
Department of Computer Science
Faculty of Computer Science, Electrical Engineering and Mathematics
Warburger Str. 100
33098, Paderborn

Abstract

Social media is now widely recognized as a valuable source of information during disaster situations due to its real-time communication and information-sharing capabilities. In the aftermath of a disaster, social media can provide significant information about the location, damage, and needs of affected individuals. This can help emergency responders to better understand disaster situations and make informed decisions. Social media can also facilitate rescue and relief efforts, as well as resource distribution to those in need. Despite these benefits, the massive volume of social media data poses significant challenges to find informative messages. Furthermore, processing social media data such as tweets involves additional challenges due to their limited content, noise, and informality. In this thesis, we aim to leverage social media data, in particular tweets, to improve situational awareness during disasters. Towards this goal, we developed various approaches for processing social media data to extract relevant features and actionable information. Concretely, we present the following contributions in this thesis:

- We propose a joint learning approach that integrates social media and environmental data to classify disaster events. Our approach extracts relevant features from social media data and outperforms baseline methods.
- Tweet classification is an essential task for finding relevant information. However, most previous approaches consider this task as a binary or multi-class problem and ignore the fact that a tweet may belong to one or more classes simultaneously. For this purpose, we propose our approach, UPB-BERT, which fine-tunes the BERT language model for multi-label classification of disaster-related tweets. Our approach demonstrates significant performance on a real-world dataset of tweets collected from different disaster events.
- We propose our approach, I-AID, to identify actionable information in disaster-related tweets. Such information is critical for emergency managers and relief organizations to respond faster and prepare efficient disaster mitigation plans.
- Situational insights can be obtained from social media data by identifying topics, trends or hashtags. However, a large volume of shared social media data without hashtags makes it difficult to find or categorize it efficiently. In our studies, we propose our approach, MULTPAX, to summarize disaster-related tweets by extracting salient phrases. This approach enables the identification of topics, trends, or hashtags in social media data, even without hashtags. We evaluate

our approach on different benchmark datasets for keyphrase extraction, as well as domain-specific disaster tweets. Our evaluation results demonstrate that MULTPAX performs better than state-of-the-art baselines in extracting present keyphrases and generating absent ones.

Abstract (German language)

Soziale Medien werden mittlerweile als wertvolle Informationsquelle in Katastrophensituationen anerkannt, da sie eine Echtzeitkommunikation und Informationsweitergabe ermöglichen. Im Nachgang einer Katastrophe können soziale Medien wichtige Informationen über den Ort, die Schäden und die Bedürfnisse der betroffenen Personen liefern. Dies kann den Einsatzkräften helfen, Katastrophensituationen besser zu verstehen und fundierte Entscheidungen zu treffen. Soziale Medien können auch Rettungs- und Hilfsmaßnahmen sowie die Verteilung von Ressourcen an Bedürftige erleichtern. Trotz dieser Vorteile stellt das massive Volumen an sozialen Medien Daten eine erhebliche Herausforderung dar, um informative Nachrichten zu finden. Darüber hinaus bringt die Verarbeitung von sozialen Medien Daten wie Tweets zusätzliche Herausforderungen mit sich, da sie einen begrenzten Inhalt, Lärm und Informalität aufweisen. In dieser Arbeit zielen wir darauf ab, soziale Medien Daten, insbesondere Tweets, zu nutzen, um das situative Bewusstsein während Katastrophen zu verbessern. Zu diesem Zweck haben wir verschiedene Ansätze für die Verarbeitung von sozialen Medien Daten entwickelt, um relevante Merkmale und handlungsrelevante Informationen zu extrahieren. Konkret präsentieren wir die folgenden Beiträge in dieser These:

- Wir schlagen einen gemeinsamen Lernansatz vor, der soziale Medien und Umweltdaten integriert, um Katastrophenereignisse zu klassifizieren. Unser Ansatz extrahiert relevante Merkmale aus sozialen Medien Daten und übertrifft bestehende Basismethoden.

- Die Klassifizierung von Tweets ist eine wesentliche Aufgabe, um relevante Informationen zu finden. Die meisten bisherigen Ansätze betrachten diese Aufgabe jedoch als ein binäres oder multiklassen Problem und ignorieren die Tatsache, dass ein Tweet gleichzeitig zu einer oder mehreren Klassen gehören kann. Zu diesem Zweck schlagen wir unseren Ansatz UPB-BERT vor, der das BERT-Sprachmodell für die multilabel-Klassifizierung von katastrophenbezogenen Tweets feinabstimmt. Unser Ansatz zeigt eine signifikante Leistung auf einem realen Datensatz von Tweets, die aus verschiedenen Katastrophenereignissen gesammelt wurden.

- Wir schlagen unseren Ansatz I-AID vor, um handlungsrelevante Informationen in katastrophenbezogenen Tweets zu identifizieren. Solche Informationen sind entscheidend für Einsatzleiter und Hilfsorganisationen, um schneller zu reagieren und effiziente Katastrophenschutzpläne zu erstellen.

- Situative Erkenntnisse können aus sozialen Medien Daten gewonnen werden, indem Themen, Trends oder Hashtags identifiziert werden. Ein großes Volumen an

geteilten sozialen Medien Daten ohne Hashtags erschwert es jedoch, diese effizient zu finden oder zu kategorisieren. In unseren Studien schlagen wir unseren Ansatz MULTPAX vor, um katastrophenbezogene Tweets durch Extraktion von salienten Phrasen zusammenzufassen. Dieser Ansatz ermöglicht die Identifizierung von Themen, Trends oder Hashtags in sozialen Medien Daten, auch ohne Hashtags. Wir evaluieren unseren Ansatz auf verschiedenen Benchmark-Datensätzen für die Schlüsselwortextraktion sowie domänenspezifischen Katastrophentweets. Unsere Evaluierungsergebnisse zeigen, dass MULTPAX eine überlegene Leistung im Vergleich zu state-of-the-art Baselines bei der Extraktion von vorhandenen Schlüsselwörtern und der Generierung von abwesenden erreicht.

Acknowledgement

First, and foremost, I want to acknowledge and thank everyone who helped me get to this point, I would not have been able to do this without the many people that have helped me along over the years.

I would like to express my deepest gratitude to my supervisor Prof. Dr. Axel-Cyrille Ngonga Ngomo and my advisor Dr. Mohamed Ahmed Sherif. My research advisors not only taught me invaluable skills but, most importantly, they gave me an opportunity and a chance in the first place. My PhD contributions would not have been possible without their endless support and understanding, allowing me to adjust my research agenda to my own interests. I remain indebted to them for the guidance and encouragement they provided me since day one.

I would like to express my sincere gratitude to my second reviewer, Prof. Dr. Michael Cochez, as well as the remaining members of my thesis committee —Prof. Dr. Yasemin Acar, Jun.-Prof. Dr. Sebastian Peitz, and Dr.-ing. Philipp Terhörst— for their insightful comments and constructive feedback.

I was fortunate to be part of the DICE group, where I met wonderful researchers and fellow PhD students who enriched my academic experience and made my stay in Paderborn enjoyable. I thank them all for their collaboration, assistance, and camaraderie.

Moreover, I acknowledge the financial support from the DAAD-GERLS, which enabled me to conduct this research in Germany and access excellent academic resources and facilities. I am thankful for their generous scholarship and trust in my potential.

Last, but not least, I would like to thank my family for their unconditional love and support. I dedicate this thesis to my father (R.I.P.) for giving me a lifelong love of learning, and to whom I owe more than words could ever express.

Dedicated to my parents ♡

Contents

List of Figures

List of Tables

1 Introduction

1.1 Social Media during Disasters

Social media platforms have become an integral part of everyday life and have transformed the way people communicate, especially during crises[1]. These platforms allow eyewitnesses to share real-time information about the damages, risks, and needs of affected people (Ogie et al., 2022). People often rely on platforms such as Twitter to disseminate information related to updates, alerts, rescues, and relief requests. Therefore, emergency response organizations (e.g., Red Cross) monitor social media data to obtain valuable insights into the situation and the needs of disaster victims. For instance, during hurricane *Harvey* in 2017, the emergency telephone number (911) in the USA was overwhelmed with thousands of calls from people requiring immediate assistance. As an alternative, many people turned to social media to seek help and access disaster relief information (Villegas et al., 2018).

Various applications have been developed and integrated into social media platforms in recent years to facilitate crisis communication and improve situational awareness. For example, *Facebook Crisis Response*[2], is a channel for crisis communications that allows users to mark themselves as safe and notify their families during nearby crises. Moreover, organizations such as the federal emergency management agency (FEMA) launched their own social media platform that allows users to receive information and submit images related to disasters. These applications enable people to share large volumes of data which pose challenges for filtering valuable information. Hence, several studies have explored the use of social media in disaster management and relief responses (Houston et al., 2015; Landwehr and Carley, 2014; Simon et al., 2015). Effective disaster management can be achieved by detecting crisis events, filtering useful information, and providing situational insights (e.g., event summaries) (Saroj and Pal, 2020). However, different challenges have arisen due to the unstructured nature, limited content, and informal style of social media data (Chy et al., 2021). Therefore, our main objective of this study is to develop

[1] Throughout this thesis, we use *crisis*, *disaster*, and *emergency* as interchangeable terms
[2] https://www.facebook.com/about/crisisresponse/

efficient approaches that can detect events, provide valuable information, and offer situational insights during disaster situations.

1.2 Mining and Processing Social Media Data

Collecting Social Media Data. Twitter[3] is one of the most popular social media platforms, which allows users to communicate and share information through short posts called tweets.[4] During crisis events, such as natural disasters or political upheavals, millions of tweets are generated and disseminated, containing valuable information about the situation, the needs of the affected people, and the images of the impacted locations. However, collecting and analyzing relevant tweets is not a trivial task, as it requires filtering out noise and irrelevant information from the massive stream of data. One common way of filtering is to use keywords and hashtags that are related to the event of interest. For instance, during hurricane SANDY, researchers collected more than 20 million tweets using hashtags *#sandy* and *#hurricane* (Dong et al., 2013). However, this method has limitations, as not all relevant tweets may contain keywords or hashtags of interest. Another possible method is to use geolocation information to identify tweets that originate from the event location. However, only a small percentage of tweets have geolocation information attached to them (Middleton et al., 2013). To address these challenges, various applications have been developed and integrated into social media platforms, such as Twitter and Facebook, to collect and monitor social media data during crisis events. In Section 2.8, we summarize some of the most popular applications that have been developed in recent years.

Social Media Preprocessing. Through the use of search APIs and endpoint services, various approaches have been proposed for mining large-scale social media streams. These approaches can be categorized into supervised and unsupervised methods. Supervised methods (e.g., Support Vector Machine (SVM), NavieBayes) have shown significant performance in filtering data when they are trained on large labelled data. On the other hand, unsupervised methods do not require any labelled data. They use techniques like clustering to group similar data together and separate them from anomalies or noisy data. However, unsupervised approaches rely on a set of preprocessing steps to clean social media data from redundant and irrelevant data. In Section 2.2, we describe how social media data, in particular tweets, is processed.

[3]We used Twitter to collect social media data since it is the most popular social platform
[4]Short posts currently limited to 280 characters

Figure 1.1: A tweet example requested a help during hurricane *Sandy*

1.3 Motivation

Data is meaningless unless it is processed into information. Several studies demonstrated that social media data can be a valuable source of information about affected persons, donation offers, help requests, and advice during disasters (Nazer et al., 2016). An example of such a tweet, posted during hurricane *Sandy*, is presented in Figure 1.1[5]. The tweet expresses an urgent request for emergency aid from a stranded person. Social media data is a form of big data that has high *volume*, *velocity*, and *variety*. However, it also poses many challenges for data analysis, such as incompleteness, noise, and informality (e.g., slang and abbreviations). Therefore, advanced techniques are needed to extract relevant information and turn it into actionable insights and timely actions (Villegas et al., 2018).

Previous studies have mainly focused on the binary classification of social media data (relevant vs. irrelevant) for disaster management (To et al., 2017). However, recent studies have highlighted the importance of fine-grained filtering of social media data into different types of information, which can facilitate more effective disaster responses (Olteanu et al., 2015). By categorizing crisis data into deeper levels of information types, emergency organizations (e.g., humanitarian relief, governments, or local police departments) can develop a taxonomy of disaster-related information, allowing each organization to access a specific subset of crisis data based on their information needs. For example, the Red Cross organization can benefit from information about emergency cases such as medical reports on severely injured or trapped individuals. On the other hand, information about resultant damages is essential for governments to initially assess losses. In general, there are two ways to analyze disaster events: i) understanding *"the big picture"*, which provides an overview of the situation, and ii) identifying *"actionable information"*, which

[5]The content is used under the Twitter licence: https://developer.twitter.com/en/developer-terms/agreement-and-policy

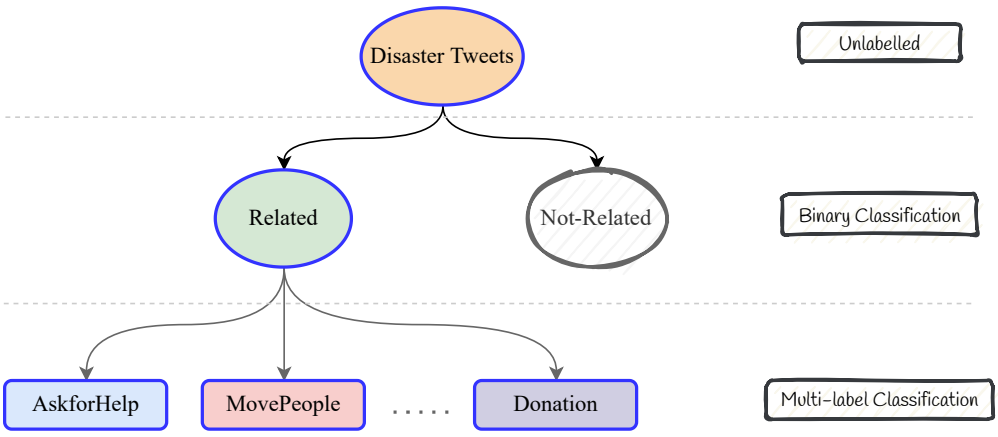


Figure 1.2: Fine-grained classification of disaster-related tweets

requires immediate actions or interventions. Figure 1.2 illustrates the classification of disaster-related tweets into binary and fine-grained information types.

Previous approaches have addressed the use of social media in disaster management by tackling three main challenges:

- **Detecting events from social media data**: One of the essential functions of disaster preparedness is detecting crisis events. By tracking the situation of a crisis, first-aid responders can implement timely and appropriate responses, thereby reducing disaster losses (Nazer et al., 2016).
- **Classifying social media data into fine-grained information types**: During disasters and emergencies, emergency managers need a comprehensive view of the crisis situation to coordinate efforts and make decisions effectively. Several studies have focused on filtering useful information from social media (Imran et al., 2018; Nazer et al., 2017). These approaches consider filtering crisis data as a binary classification task, i.e., data is classified as relevant or irrelevant. However, Olteanu et al. (2015) highlighted the importance of filtering disaster-related data into fine-grained information types, which can support emergency responders in taking appropriate actions.
- **Summarizing events and getting situational insights**: Massive amounts of social media data (e.g., tweets) are generated during crisis events, making it difficult to obtain situational insights from the topics discussed in crisis data (Yu et al., 2018). Most existing approaches use keyphrase extraction from social media data to summarize crisis events. Nevertheless, some important terms may not be present in social media data due to their shortness and

incompleteness. In our study, we addressed this problem into two sub-tasks: i) present keyphrase extraction and ii) absent keyphrase generation.

These findings provide solid evidence of the crucial role that social media can play in disaster situations. In this thesis, we identified these areas as potential opportunities to effectively leverage social media in managing disaster situations. Our research focused on developing efficient approaches for processing crisis information on social media to improve situational awareness.

1.4 Research Questions and Contributions

In this section, we present the research questions derived from Section 1.3 and our contributions.

Challenge I: Detecting disaster events using social media data

Research Question 1

How does social media data (tweets) enhance the performance of disaster prediction models that use only environmental data?

We addressed the problem of disaster prediction by considering social media as a complementary source of information to environmental data. We propose a joint learning approach that integrates features from both social media and environmental data (Zahera et al., 2019b). Specifically, we analyzed disaster-related tweets about typhoons to identify additional features for predicting typhoon categories. Moreover, we derived adaptive features based on two joint training models (e.g., BiLSTM+CNN). The first model (BiLSTM) acts as a *Feature Extractor* from social media data, while the second model (CNN) combines features from tweets and environmental data. We provide more details about our approach and experimental results in Chapter 4.

Research Question 2

What is the impact of semantic embeddings in tweet representation on the performance of disaster prediction?

To answer this question, we investigated the application of semantic embeddings from the ConceptNet knowledge graph in predicting disasters (Zahera et al.,

2019b). We identified named entities from tweets and represented them using a fusion of traditional word embeddings and semantic embeddings from CONCEPTNET. Our experiments demonstrate a significant performance (in terms of *Accuracy*, *Precision*, *Recall* and F_1) when incorporating semantic embeddings in tweet representation compared to traditional word embeddings. Further details of our approach can be found in Chapter 4.

Challenge II: Classifying disaster-related media data into fine-grained information types

Research Question 3

How effective is fine-tuning a pre-trained language model in categorizing disaster-related tweets into multiple information types?

To address this question, we explored the effectiveness of fine-tuning a pre-trained BERT language model for multi-label classification of disaster-related tweets. To the best of our knowledge, our approach is the first study that fine-tunes the pre-trained BERT model for this task (Zahera et al., 2019a). We developed two variants of fine-tuned BERT models: the first one (UPB-BERT) optimizes a binary Cross-entropy loss function to reduce training errors, while the second one (UPB-FOCAL) employs a Focal loss function to mitigate the class imbalance problem in the TREC-IS dataset. Furthermore, our approach utilizes contextualized word embeddings from a pre-trained BERT model to capture the semantic features of tweets. The experimental results demonstrate that fine-tuning BERT model achieves superior performance in classifying tweets into multiple information types. We present more details of our approach and experiments in Chapter 5.

Research Question 4

What is the impact of incorporating a pre-trained language model and graph attention network on the categorization of disaster-related tweets into multiple information types?

To address the problem of multi-label tweet classification, we propose I-AID, a multimodel approach that automatically categorizes tweets into multiple information types (i.e., classes or labels) (Zahera et al., 2021). I-AID consists of three components: i) a BERT-ENCODER to represent tweets as contextualized embedding vectors. ii) TEXTGAT, a graph attention network to identify correlations between

tweets' words, entities or information types, and iii) a *Relation Network* as a learnable distance metric that measures the similarity of tweets and their corresponding information types in a supervised way. We present more details of our approach and experiments in Chapter 6.

Research Question 5

How effective is our approach (I-AID) in identifying actionable information from disaster-related tweets?

To answer this question, we evaluated the performance of our approach in alerting actionable information in tweets using the *Accumulated Alert Worth* (AAW) metric (McCreadie et al., 2019). Actionable information can be defined in two ways: i) as high-priority information, commonly labelled as critical by human assessors, and ii) as information type, for instance, tweets with labels *MovePeople* or *Donations* are actionable as opposed to *News* or *Multimediashare*. In our study, we adopted the second definition of *actionable* posts as highly prioritized tweets. We provide more details in Chapter 6.

Challenge III: Summarizing disaster-related social media data for situational insights

Research Question 6

How do pre-trained language models and knowledge graphs improve keyphrase extraction compared to state-of-the-art baselines?

There are two ways for summarizing text: extracting salient phrases (a.k.a. extractive summarization) or generating human-like summaries (a.k.a. abstractive summarization). In our study, we consider the first approach to summarize social media data to obtain situational insights. For this purpose, we propose MULTPAX, a multi-task framework for extracting *present* using pre-trained language models and generating *absent* ones using knowledge graphs (Zahera et al., 2022). The pipeline of our approach consists of three steps: i) MULTPAX extracts present keyphrases from input disaster-related tweets, ii) MULTPAX links the present keyphrases with knowledge graphs to get more relevant phrases, and iii) finally, MULTPAX ranks both the present and absent keyphrases based on their semantic relatedness to the input tweets. More details about our approach can be found in Chapter 7.

Research Question 7

How suitable are the exact-matching metrics (Precision, Recall and F_1-score) for evaluating absent keyphrases?

To answer this question, we evaluated the performance of absent keyphrases generation using the exact matching metrics. We found that evaluation based on exact matching is not suitable for this task, as it ignores semantically similar words and only counts matches when predicted and ground-truth keyphrases are identical. For example, if two keyphrases are semantically similar, such as *"disaster relief organization"* and *"crisis responses institute"*, these keyphrases are not considered as a match by the exact matching metrics. Therefore, we propose a semantic matching evaluation, which takes into account semantically similar keyphrases. We provide more details in Chapter 7.

1.5 Thesis Outline

In this section, we describe the thesis structure, which includes eight Chapters. The remaining Chapters of the thesis are summarized as follows:

- **Chapter 2** introduces the background knowledge on disaster management using social media, describes the fundamental methods for collecting and processing social media data, and provides a list of applications developed for leveraging social media during disasters.
- **Chapter 3** reviews the state-of-the-art approaches in leveraging social media for disaster situations, summarizing the approaches related to three disaster management tasks: event detection using social media, filtering useful information from massive social media, and event summarization.
- **Chapter 4** describes our approach that predicts typhoon intensities using *joint learning* from social media and environmental data. We investigated the impact of applying semantically enriched data representation on the performance of our approach. Specifically, we used semantic embeddings from the ConceptNet knowledge graph to represent tweets. Our experimental results demonstrate superior performance compared to state-of-the-art baselines in typhoon prediction when integrating features from both social media data and environmental data.
- **Chapter 5** describes our approach (UPB-BERT), a fine-tuned BERT model for multi-label tweet classification during crises. We have collected and annotated a

large-scale dataset of real-world tweets from various crisis events and evaluated our model on it. To the best of our knowledge, this is the first study that applies pre-trained language models to this task (Zahera et al., 2019a).

- **Chapter 6** introduces our approach (I-AID), a multi-modal approach for multi-label tweet classification that combines three components: BERT-ENCODER, TEXTGAT, and RELATION NETWORK. We have shown that our approach can effectively capture both local and global information in short texts and outperform existing methods on several benchmark datasets. We have also discussed the challenges and limitations of multi-label classification and suggested proper metrics for fine-grained evaluation.
- **Chapter 7** presents our approach (MULTPAX), a multi-task framework for extracting present and absent keyphrases from crisis data. We have utilized the pre-trained BERT model and knowledge graphs (DBPEDIA and BABELNET) to obtain present and absent keyphrases. Our experimental results demonstrate that knowledge graphs are valuable resources for generating keyphrases.
- **Chapter 8** concludes the thesis, summarized the main findings of our research, and suggests potential future directions for further improvement and exploration

1.6 Own publications

The contents of Chapters 4–7 are mainly based on the following research papers published at international peer-reviewed conferences and journals:

1. **Hamada M. Zahera**, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. "*Jointly learning from social media and environmental data for typhoon intensity prediction*". In Proceedings of the 10th International Conference on Knowledge Capture (**K-CAP**), pp. 231-234. 2019.
2. **Hamada M. Zahera**, Ibrahim A. Elgendy, Rricha Jalota, Mohamed Ahmed Sherif, E. M. Voorhees, and A. Ellis. *"Fine-tuned BERT Model for Multi-Label Tweets Classification"*. In Text Retrieval Conference (**TREC**), pp. 1-7. 2019.
3. **Hamada M. Zahera**, Rricha Jalota, Mohamed Ahmed Sherif and Axel-Cyrille Ngonga Ngomo. *"I-AID: Identifying Actionable Information From Disaster-Related Tweets"*. **IEEE Access** 9 (2021): 118861-118870.
4. **Hamada M. Zahera**, Daniel Vollmers, Mohamed Ahmed Sherif, Axel-Cyrille Ngonga Ngome. *"MultPAX: Keyphrase Extraction using Language Models and*

Knowledge Graphs". In the proceeding of the 21st International Semantic Web Conference (**ISWC**), pp. 303–318, 2022.

The author also contributed to the following publications during his PhD studies (not included in this thesis):

5. **Hamada M. Zahera** and Sherif, Mohamed Ahmed. *"ProBERT: Product Data Classification with Fine-tuning BERT Model"*. In the Proceedings of Mining the Web of HTML-embedded Product Data Workshop (**MWPD**) at the International Semantic Web Conference (**ISWC**), 2020.
6. **Hamada M. Zahera**, Stefan Heindorf, and Axel-Cyrille Ngonga Ngomo. "ASSET: A Semi-supervised Approach for Entity Typing in Knowledge Graphs." In Proceedings of the 11th on Knowledge Capture Conference (**K-CAP**), pp. 261-264. 2021.
7. Chakraborty, Jaydeep, **Hamada M. Zahera**, Mohamed Ahmed Sherif, and Srividya Bansal. *"OntoConnect: Domain-Agnostic Ontology Alignment using Graph Embedding with Negative Sampling"*. In the Proceedings of the IEEE International Conference on Machine Learning and Applications (**ICMLA**), 2021.
8. **Hamada M. Zahera**, Stefan Heindorf, Stefan Balke, Jonas Haupt, Martin Voigt, Carolin Walter, Fabian Witter, and Axel-Cyrille Ngonga Ngomo.*"Tab2Onto: Unsupervised Semantification with Knowledge Graph Embeddings"*. In the Proceeding of Extend Semantic Web Conference (**ESWC**) 2022, Poster Track.
9. N'Dah Jean Kouagou, Caglar Demir, **Hamada M. Zahera**, Adrian Wilke, Stefan Heindorf, and Axel-Cyrille Ngonga Ngomo. *"Universal Knowledge Graph Embedding"*. (Under submission)
10. **Hamada M. Zahera**, Fedor Vitiugin, Mohamed Ahmed Sherif, Carlos Castillo, Axel-Cyrille Ngonga Ngomo "*Towards Abstractive Summarization of DBpedia Abstracts Using Language Models*". (accepted at SEMANTiCS 2023)

1.7 Source Code

We developed four different approaches for analyzing crisis data in our studies, which are available in different Github repositories. These repositories are publicly accessible and contain the source code, documentation, and data sets that we have used in our experiments. Furthermore, we provide a detailed experimental setup and instructions to reproduce the results that we have reported in this thesis.

- **JOINT LEARNING**: This repository provides the source code of our novel approach that leverages both environmental and social media data to predict the categories of typhoons. Additionally, we provide the instructions to download our Typhoon Events Dataset (TED), which contains thousands of tweets collected during various typhoon events. More details can be found at https://github.com/dice-group/joint-model

- **I-AID**: This repository contains the source code of our methods to annotate social media data (tweets) with multiple information types. We employed the dataset (*trecis2019-B*), which was provided by the TREC-IS Challenge, to evaluate our methods. For more information, please visit https://github.com/dice-group/I-AID.

- **MULTPAX**: This responsory contains the source code and datasets used in our experiments for summarizing disaster-related tweets. We employed different benchmark datasets, as well as crisis tweets to investigate the performance of our approach in extracting present keyphrases and generating absent ones. For more details, please visit https://github.com/dice-group/MultPAX.

2 Background

This chapter provides the preliminaries necessary for the subsequent chapters in this thesis. The chapter is organized into eight sections that cover the following topics: 1) Definitions of disaster-related concepts, 2) Collecting and preprocessing of social media data, 3) Representation learning from social media, 4) Deep learning models, 5) Knowledge graphs, 6) Evaluation metrics and 7) Datasets, 8) Applications.

2.1 Concepts and Terminology

Emergency-related terms should be defined precisely since many of them have special meanings depending on the context. We define the terms and concepts used in our study as follows:

Disaster management is the process of identifying, assessing, and addressing the risks and impacts of natural and human-made disasters (Stoyanov, 2017). It involves activities such as emergency planning, evacuation, relief, assistance to affected people, and infrastructure recovery. The aim of disaster management is to mitigate the negative impact of disasters on individuals, communities, and societies.

Crisis informatics is an interdisciplinary field that focuses on how technology and information systems can help manage and respond to crisis situations (Reuter and Kaufhold, 2018). It develops tools, systems, and processes for collecting, analyzing, and disseminating information during crises, as well as for coordinating and supporting emergency response efforts.

Situational awareness is the ability to understand what is happening in ongoing events, especially in the context of response and control operations (Vieweg et al., 2010). In crisis management, situational awareness involves collecting and analyzing information about the crisis, such as its cause, potential impacts, and possible response strategies. By maintaining situational awareness, decision-makers can identify urgent needs and prioritize their response efforts accordingly.

Information types are categories of social media data generated during crisis events, based on specific types of information (Olteanu et al., 2015). For example, the information type *"Donation"* indicates that a collection of social media data (e.g., tweets) contains valuable information about donation campaigns (see Table 5.1 for more details). A single tweet may belong to one or more information types at the same time.

Actionable information is factual information that can be automatically acted upon (e.g., moving people, requesting volunteers) (Zade et al., 2018). We are concerned with social media data (e.g., tweets) that generate immediate alerts for emergency responders, such as situational information, sentiment, and personal opinions. Situational data is essential for assisting authorities to understand the current disaster situation (e.g., the number of people affected) so that relief efforts can be coordinated appropriately.

2.2 Collecting and Processing Social Media Data

In our study, we collected social media data from the *Twitter* platform to analyze disaster-related posts. We used the *Twitter stream API*[1] to search for disaster-related tweets using *keywords* in search terms (i.e., hashtags). We retrieved the tweets and their corresponding information from Twitter by passing the "id" parameter to the REST endpoint if the tweets' IDs were available. We collected approximately $1,3$ million tweets related to different typhoon events. Due to Twitter's copyright license, we shared only the tweets ids on the GitHub repository.[2] The collected tweets are unstructured and varied in their readability, grammar, and syntax. To preprocess these tweets, we used the TWEETARC[3] library, as it provides specialized preprocessing steps for tweets. For example, TWEETARC library can identify the emojis, '@' mentions (*usernames*), and *RT* tags (*retweets*) commonly found in tweets. We also removed *URLs*, *mentions*, *hashtags*, *emojis*, *smileys*, *special characters*, and *stop words* since they did not contribute semantic information. Finally, we used tokenized and lowercase words to reduce typographical errors. Figure 2.1 shows an example of preprocessing a tweet shared during a tornado in Kansas City in 2022. The preprocessing steps are described as follows:

- **Tokenization**: This process splits text into individual tokens, which can be words, characters, symbols, or n-grams. Different delimiters, such as #, tabs,

[1] https://tinyurl.com/yc6wufp5
[2] https://github.com/dice-group/joint-model/tree/master/TED%20Dataset/Tweets_IDs
[3] http://www.cs.cmu.edu/~ark/TweetNLP/

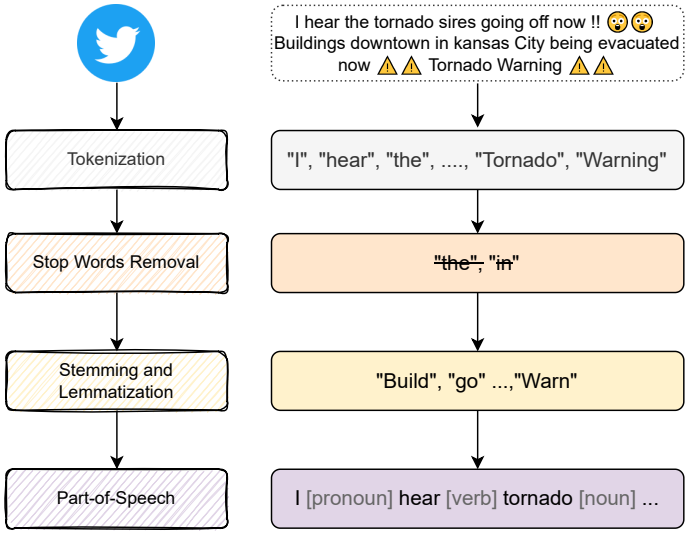


Figure 2.1: The pipeline of tweets preprocessing

whitespace, and newlines, can be used to separate these tokens. The most common tokenization technique is whitespace splitting, which segments text into words separated by whitespace.

- **Stop Words Removal**: This step removes common words with minimal semantic value. Stop words, such as *'the', 'a'*, and *'in'*, are frequently used in English but contribute little to the meaning of a sentence.

- **Stemming and Lemmatization**: These techniques reduce words to their base forms, also known as the root word or lemma (Sharma and Cse, 2012). Stemming trims words of their affixes (prefixes or suffixes) to approximate their base forms, also known as the root word or lemma (Sharma and Cse, 2012). For instance, the word *"warning"* is transformed to *"warn"* by removing the *"ing"* suffix during stemming. Lemmatization, on the other hand, uses a dictionary to look up the word's lemmas (Balakrishnan and Lloyd-Yemoh, 2014). For example, the word *"went"* is transformed to *"go"* by finding its lemma in the dictionary. Stemming is faster and simpler than lemmatization, but it may produce inaccurate or non-existent words.

- **Part of Speech Tagging (POS)**: This process annotates text with labels (i.e., tags) based on the roles of words in a sentence (e.g., subject, verb, adjective, etc) (Brill, 1995). This helps determine the importance of words based on their tags (e.g., assigning more weight to nouns than adjectives).

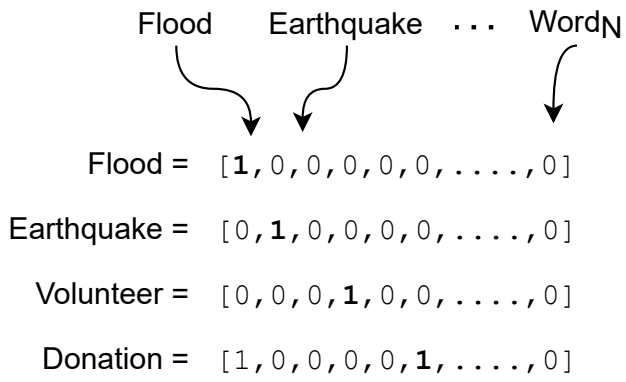


Figure 2.2: One-hot encoding of words *Flood*, *Earthquake*, *Volunteer*, and *Donation*

2.3 Representation Learning Approaches

One of the key steps in preparing data for machine learning models is feature engineering, which involves transforming raw data into features that capture the essential patterns and relationships within data. These features should be relevant and informative for the specific machine learning task and should reflect the domain expertise and problem understanding of the data analyst. Thus, feature engineering depends on knowledge and a comprehensive understanding of the data and problem domain. Different approaches have been developed to extract features from data, including *traditional* (i.e., classical) approaches, which mainly rely on statistical features, and *neural* approaches that utilize deep neural networks.

2.3.1 Traditional Representation

One-hot encoding (Bernard and Lebboss, 2017) is a simple approach for text encoding, wherein words are represented with numerical vectors with dimensionality equal to the vocabulary size. Each word has a unique dimension, indicated by a one in that dimension and zeros elsewhere. Figure 2.2 illustrates the one-hot encoding of words (*Flood*, *Earthquake*, *Volunteer*, *Donation*) are encoding as one-hot vectors, where only the dimension corresponding to the words is marked by one. However, one-hot encoding can suffer from the problem of high dimensionality, especially when dealing with categorical variables that have many categories. This can cause the *"curse of dimensionality"*, where the high-dimensional feature space can pose challenges for machine learning algorithms to analyze the data efficiently.

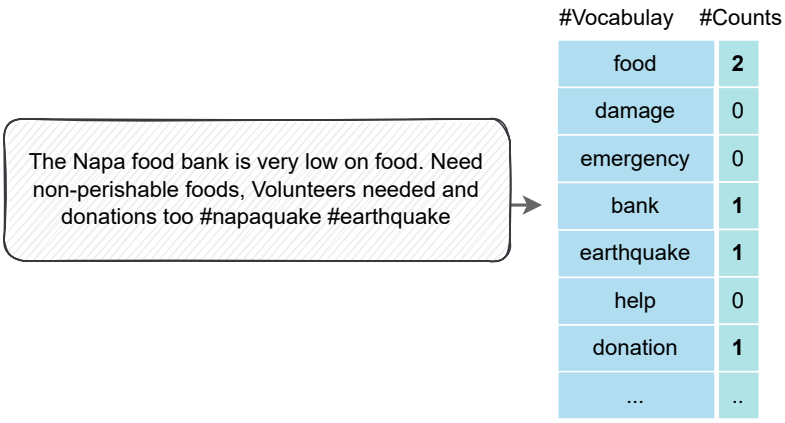


Figure 2.3: A bag-of-words representation of a tweet during California earthquake 2014

Bag-of-words (BoW) is a similar approach to one-hot encoding, which counts the frequency of words in each sentence and represents them as a vector of word counts (Zhang et al., 2010). For instance, in Figure 2.3, the sentence "*The Napa food bank is very low on food. Need non-perishable foods, Volunteers needed and donations too #napaquake #earthquake*" is encoded as a vector of word counts $[2, 0, 1, 1, 0, 0, 1, 1, \cdots]$, where the index corresponds to the word id and the value represents its frequency. However, this approach also has some limitations; it can result in sparse text representation when the input data is large, and the vocabulary is huge. Moreover, it does not capture the semantics and meaning of words, as it only considers their frequency.

Term frequency–inverse document frequency (TF-IDF) is a statistical method that measures the importance of each word in a document relative to a collection or corpus of documents (Ramos et al., 2003). TF-IDF assigns high weights to the words that are frequent in a document but are rare in other documents. This way, it reduces the impact of words (e.g., this, that, are, etc.) that are common across all documents. The TF-IDF method consists of two components, as follows:

$$\text{TF}\ (t, d) = \frac{\text{count}\ (t, d)}{||T||}, \tag{2.1}$$

where count (t, d) is the frequency of term t in document d, and $||T||$ is the total number of terms in document d.

The Inverse Document Frequency (IDF) of term t is given by:

$$\text{IDF}\ (t) = \log\ \frac{||D||}{1 + ||d \in D : t \in T||}, \tag{2.2}$$

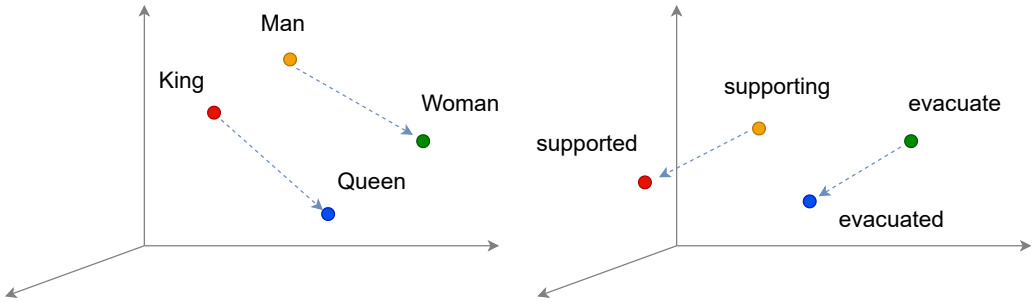


Figure 2.4: An example of words embeddings in the Semantic Space

where $||D||$ is the number of documents in the corpus, and $|d \in D : t \in T|$ is the number of documents $d \in D$ that contain the term $t \in T$. The TF-IDF weight is then computed by multiplying the TF and IDF values.

$$\text{TF-IDF}\ (t, d) = \text{TF}\ (t, d) \times \text{IDF}\ (t), \tag{2.3}$$

For instance, suppose we have a document with 100 terms, where *tsunami* term occurs three times. The TF of $tsunami$ is calculated as $(3/100) = 0.03$. Suppose also that we have a corpus of 1000 documents and the term *tsunami* appears in 300 of these documents. The IDF of *tsunami* term is $\log\ (10,000/300) = 0.52$ and the TF-IDF weight is then $0.03 \times 4 = 0.015$.

2.3.2 Embeddings-based Representation

Word Embeddings (Word2vec) is a widely used technique that represents words in the text as numerical vectors (called embedding vectors) in a high-dimensional semantic space (Mikolov et al., 2013b). This representation captures the semantic similarity between words, such that words with similar meanings have similar embedding vectors. Figure 2.4 shows two examples of word embeddings (nouns and verbs). We can observe that the embedding vectors of *"man"* and *"king"* are similar to those *"queen"* and *"woman"*. Consequently, their relationship is established as follows:

$$\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{queen}.$$

Moreover, the embedding vectors of verbs preserve similarity across different tenses. For instance, the embeddings vector of the verb *"supporting"* is close to its past tense form *"supported"* in the semantic space, as shown in Figure 2.4. Various methods

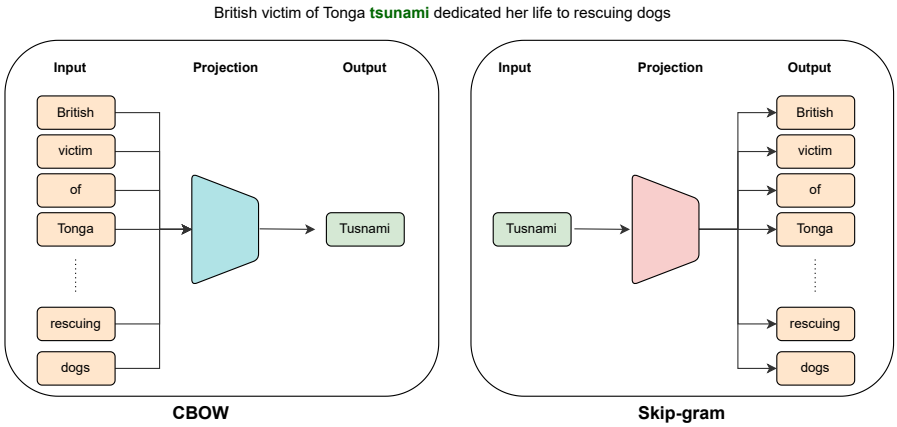


Figure 2.5: An illustration of the CBOW model vs the Skip-gram model

have been developed to learn word embeddings in semantic space. We give a brief overview of the most common methods as follows:

- **Continuous bag-of-words (CBOW)** learns an embedding vector for a target word from its context (Mikolov et al., 2013b). Specifically, CBOW takes the surrounding words within a specified window size as an input context. Then, it uses a projection layer to predict the target word in the centre of the window using a weight matrix (see Figure 2.5). Finally, the predicted word and the target word are compared to update the embedding representation based on the gradient errors.

- **Skip-gram** is the reverse of the CBOW model, where the context words are predicted from the target word. The input layer contains the target word, while the output layer consists of multiple words from its context. In this way, Skip-gram infers the context given a target word, unlike CBOW. Then, the similarity between the predicted word and the context words is used to adjust the embedding representation based on the gradient errors. Although the CBOW model trains faster than Skip-gram, the latter performs better for rare words (Mikolov et al., 2013a).

- **GloVe** is a variant of the Word2vec model that learns word representation based on global word co-occurrence (Socher and Manning, 2014). Glove involves two main steps: i) creating a co-occurrence matrix from the corpus, where each cell counts how often a pair of words appear together in a context window, and ii) applying factorization to the matrix to obtain the vectors for each word.

Contextualized embeddings: While static word embeddings have shown remarkable performance in various natural language processing (NLP) applications, such

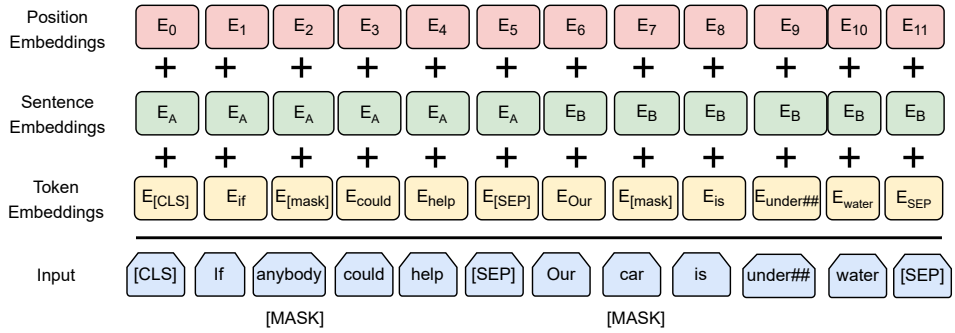


Figure 2.6: The architecture of BERT encoder for sentence representation

approaches assign a single representation for each word, ignoring the context in which words may have different meanings. For example, given two sentences with *"bank"* word: i) *"My friend is considering taking a loan from a bank"* and ii) *"Rainfall caused the Rhine river to overflow its bank"*. Traditional word embeddings approaches (e.g., Skip-gram, GloVe) learn a single embedding for the word *"bank"*, regardless of its different meanings depending on the sentence context. Hence, contextualized embeddings has been proposed to learn word embeddings based on their context (Hofmann et al., 2021). Recently, various models have been developed to provide contextualized embeddings, such as ELMo (Embeddings from Language Models), BERT (Bidirectional Encoder Representation from Transformers), and GPT (Generative Pre-trained Transformer). We refer readers to this survey (Ethayarajh, 2019) for more details about contextualized embedding models. In the next section, we briefly describe the BERT's contextualized embeddings, which we used to represent tweets in our studies.

Bidirectional Encoder Representations from Transformers (BERT) is one of the most influential language models that employ a transformer architecture with parallel attention layers to encode both the left and right context of each word (Devlin et al., 2019). BERT is pre-trained on two unsupervised language tasks, which aim to improve bidirectional prediction and sentence-level understanding. The first task is *masked language model (MLM)*, where the model randomly masks (i.e., replaced with the "[*MASK*]" token) 15% of tokens in the input and tries to predict them from the remaining tokens. The second task is *next sentence prediction (NSP)*, where the model learns to classify whether two input sentences are consecutive or not. BERT has two main variants: BERT_{base} and BERT_{large}, which differ in the number of layers, hidden units, and attention heads. Both variants are pre-trained on a large-scale dataset consisting of books corpora and English Wikipedia articles. A key advantage of BERT is that it can be easily fine-tuned for various NLP downstream tasks without requiring re-training from scratch. In our study, we used BERT to

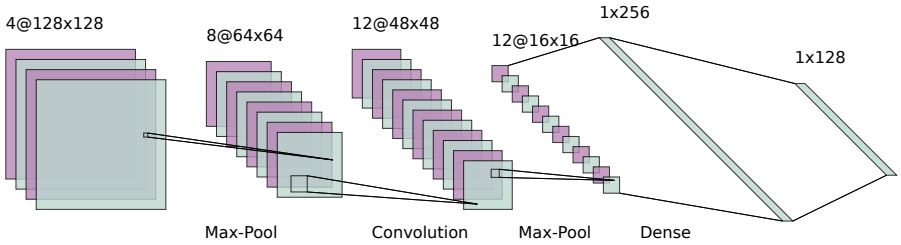


Figure 2.7: An example architecture of a convolutional neural network with Convolution, Max-pool and Dense layers

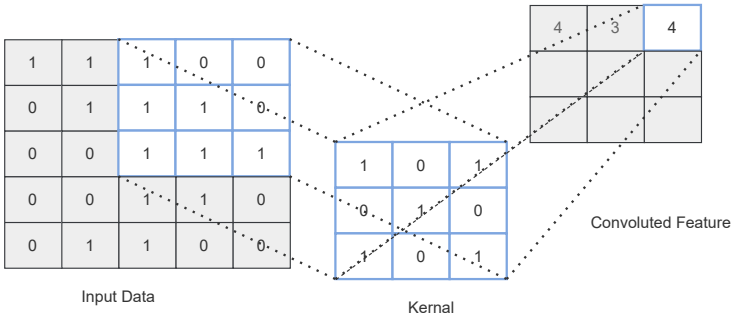


Figure 2.8: An example of convolution operation

generate contextualized embeddings of tweets. Moreover, we fine-tuned the BERT model for classifying disaster-related tweets into different information types. Interestingly, Liu et al. (2021) developed a specialized crisis embedding model (called CrisisBERT) that was trained on a large corpus of crisis-related data. CrisisBERT can detect emerging crisis events on social media, such as natural disasters, terrorist attacks, and pandemics. Additionally, it can be used to provide contextualized embedding vectors for social media data analysis.

2.4 Deep Learning Models for Natural Language Processing

In recent years, deep learning approaches have achieved remarkable results in various applications. Researchers have developed several neural architectures to address different NLP tasks, such as sentiment analysis (Yoon and Kim, 2017), text classification (Kim, 2014), and question answering (Sharma and Gupta, 2018). In our study, we explored different neural models for processing crisis data. We provide a brief overview of these models as follows:

2.4.1 Convolutional Neural Network

Convolutional neural network (CNN) is a type of deep learning model that can handle grid-patterned data (e.g., images). This model is inspired by the structure of animal visual cortex (Albawi et al., 2017) and learns spatial features in a hierarchical and adaptive manner, from low to high levels. CNN consists of three kinds of layers: *convolution*, *pooling*, and *fully connected*. The convolution layer is the core component of CNN, which performs a series of mathematical operations called convolution. The convolution and pooling layers extract features from the input data, while the fully connected layer maps those features into the final output. Interestingly, Kim (2014) adapted a CNN model for text classification by using a single convolution layer on top of a sentence representation, which is a matrix of word embeddings ($n \times \mathbb{R}^d$) with n words, each one represented by a d-dimension vector.

2.4.2 Recurrent Neural Network

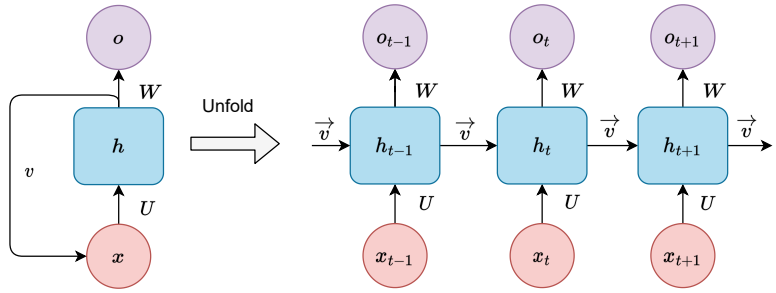


Figure 2.9: The architecture of recurrent neural network

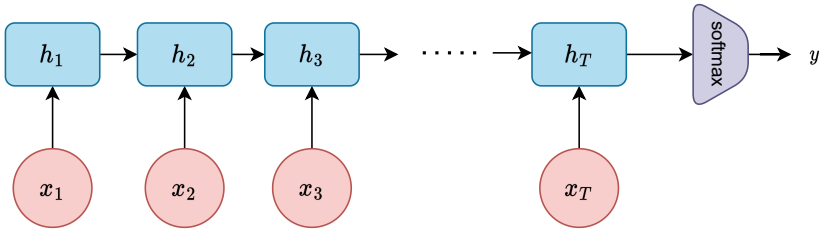


Figure 2.10: An example of RNN model for text classification

Although CNN has achieved remarkable results in various NLP tasks, they are not able to capture long-term dependencies and preserve the contextual word sequence, which is essential for understanding the overall meaning of a text in advanced tasks such as machine translation. While text data can be processed using a simple feedforward neural network, this approach does not account for the sequential information of words. To address this limitation, recurrent neural networks (RNN)

have been designed for processing sequential data, such as text (Marhon et al., 2013)). RNN operates on each word in a sequence and produces an output that depends on the previous computation. In RNN, the hidden layers are connected through recurrent connections, unlike feedforward connections in CNN. Additionally, RNN employs a *"memory"* cell to store information from the previous computations up to the current word of a sequence (Zhang et al., 2018). For example, given an input text $X = \{w_1, w_2, \cdot w_n\}$, with n words, each w_i is represented as an embedding vector $w_i \in \mathbb{R}^d$ with d-dimensions at time t. The memory cell is then used to track previous states up to time t as $\mathbf{h_t}$, which represents a hidden state at time t and acts as the network's memory. The hidden state h_t is computed based on the current input w_i and the hidden state of the previous steps. The output of RNN (o_t) is subsequently passed to a Softmax function to compute the final predictions ($\hat{y}$).

$$\mathbf{a_t} = b + \theta \times h_{(t-1)} + \beta x_t. \tag{2.4}$$

$$\mathbf{h_t} = tanh(a_t). \tag{2.5}$$

$$\mathbf{o_t} = c + V\mathbf{h}_t. \tag{2.6}$$

$$\mathbf{\hat{y}_t} = softmax(\mathbf{o_t}_t). \tag{2.7}$$

Where $\mathbf{h_t}$ is the hidden state, $\mathbf{o_t}$ is the output, and $\mathbf{\hat{y}_t}$ is the predicted label of an RNN at time step t. The hidden state $\mathbf{h_t}$ is a function of the previous hidden state $\mathbf{h}_{t--1}$ and the current input x_t, with parameters θ and β and bias b. The output $\mathbf{o_t}$ is a linear transformation of the hidden state $\mathbf{h_t}$, with parameter V and bias c. The predicted label $\mathbf{\hat{y}_t}$ is obtained by applying a Softmax function to the output $\mathbf{o_t}$, which normalizes the scores into probabilities.

One of the major challenges of training RNNs is the vanishing gradient problem (or the exploding gradient problem) (Hochreiter, 1998). This problem occurs when the gradients are multiplied by the weight contributions at each step during back-propagation. As a result, the gradient propagated to the previous time step can either shrink or grow significantly. A possible solution to this problem is to incorporate additional gates into the RNN that can selectively filter the relevant information to keep or discard for the subsequent steps. This is the idea behind Long Short-Term Memory (LSTM), which is a mechanism for controlling gradient propagation.

2.4.3 Long-Short Term Memory

Long-Short Term Memory is designed to address the vanishing gradient problem by altering the recurrence connections of the hidden states (Yu et al., 2019). LSTM

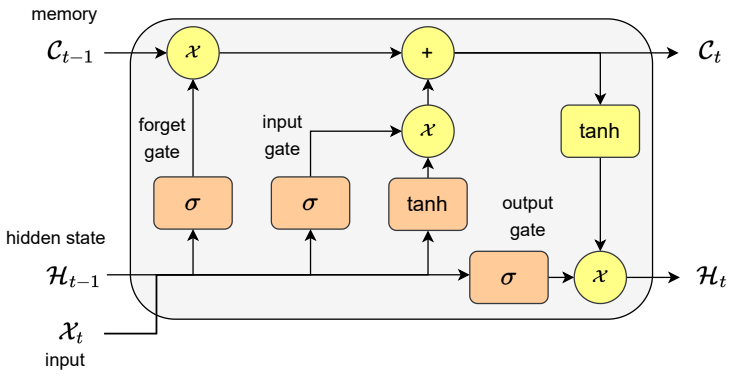


Figure 2.11: The architecture of LSTM cell

uses gates to control the gradient computation in the memory cell of the recurrent network. These gates (called the *input*, *output*, and *forget*) are used to update a memory cell with the hidden states until the next time step. The gating mechanism consists of neural network layers that decide when to forget or keep information in the memory cell (Yu et al., 2019). Figure 2.11 illustrates the diagram of an LSTM unit, which operates as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U}_i\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{2.8}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + \mathbf{b}_f), \tag{2.9}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{2.10}$$

$$\widetilde{\mathbf{c}}_\mathbf{t} = tanh(\mathbf{W}_c\mathbf{x}_t + \mathbf{U}_c\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{2.11}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \widetilde{c}_t), \tag{2.12}$$

$$\mathbf{h}_t = \mathbf{o}_t \circ tanh(\mathbf{c}_t). \tag{2.13}$$

The input, forget, and output gate vectors $\mathbf{i}_t$, $\mathbf{f}_t$, and $\mathbf{o}_t$ respectively are computed by applying a Sigmoid activation function to a linear combination of the input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$, plus a bias term. The candidate cell state $\mathbf{c}_t$ represents the potential new information to be added to the cell state $\mathbf{c}_t$, which stores the long-term memory. It is obtained by applying Tanh to a linear combination of the input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$, plus a bias term. The cell state $\mathbf{c}_t$ is updated by adding the element-wise product of the forget gate $\mathbf{f}_t$ and the previous cell state $\mathbf{c}_{t-1}$, and the element-wise product of the input gate $\mathbf{i}_t$ and the candidate cell state $\mathbf{c}_t$. The hidden state $\mathbf{h}_t$, which stores the short-term memory, is computed by applying Tanh to the current cell state $\mathbf{c}_t$ and multiplying it element-wise with

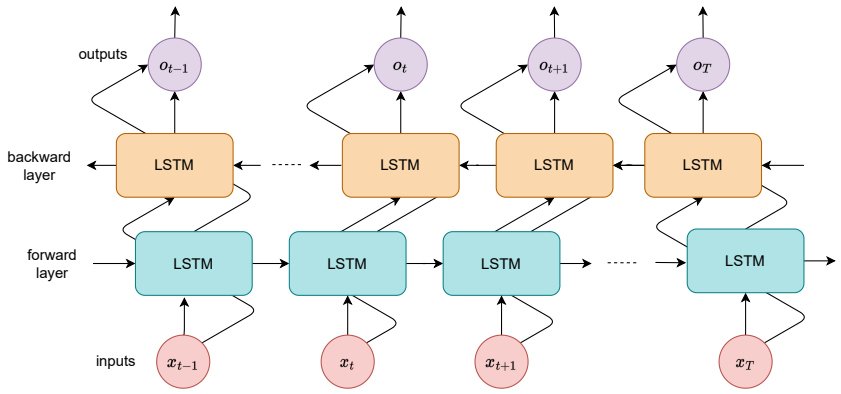


Figure 2.12: The architecture of BiLSTM model.

the output gate $\mathbf{o}_t$. These equations enable LSTM to learn long-term dependencies and avoid vanishing or exploding gradients

Bidirectional LSTM (BiLSTM) is a sequence processing model that consists of two LSTM layers; one processing the input in a forward direction, and the other in a backward direction (Yu et al., 2019). The outputs or hidden states from both directions are combined using operations such as concatenation, sum, averaging, or multiplication. BiLSTM can increase the amount of information extracted from the input text and capture the entire context. The BiLSTM architecture has many advantages in real-world problems, especially in NLP, as every component of an input sequence contains information from both left and right directions. Consequently, BiLSTM can provide better contextualized representation by integrating LSTM layers in both directions.

2.4.4 Graph Attention Network

Graph Neural Networks (GNNs) are a class of artificial neural networks for processing graph data (Scarselli et al., 2008). GNNs can tackle complex problems in various domains, such as content recommendation (Gao et al., 2022) and drug discovery (Cheung and Moura, 2020). Graph data, unlike other data types such as images, requires specialized methods for learning. To perform specific tasks on graphs (e.g., node classification, link prediction), the GNN layer computes node and edge representations using a technique called *message passing*. This technique involves each graph node receiving and aggregating features from its neighbours to capture the local graph structure. Different types of message passing layers perform different aggregation strategies.

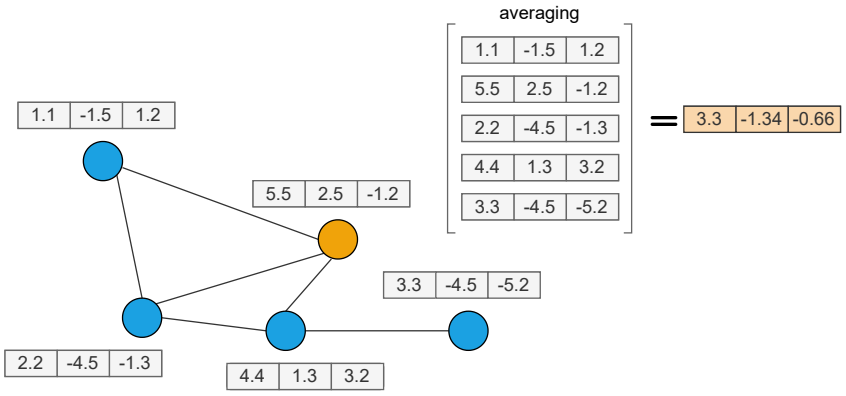


Figure 2.13: An example of the graph convolution operation.

A simple way to implement a GNN layer is to apply a convolution operation on a graph, which is known as Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017). GCNs perform a uniform aggregation, i.e., each neighbour node has the same weight in updating the target node's representation. Formally, given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ and $\mathcal{E}$ represent the set of nodes and edges, respectively. Each node is connected to itself $(v_i, v_i) \in \mathcal{E}$. Let $X \in \mathbb{R}^{n \times d}$ be the nodes' representation matrix, where each node v_i has a d-dimension feature vector. Given an adjacency matrix $\mathcal{A}$ and its degree matrix $\mathcal{D}_{ii} = \sum_j \mathcal{A}_{ij}$, the diagonal elements of A are set to 1 due to self-loops. To learn a feature representation of a target node v_i, GCN aggregates features from its neighbours $(v_1, v_2, \cdots v_A)$, i.e., each neighbour node contributes equally in updating the target node's representation.

$$\mathcal{H}^{l+1} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{\mathcal{A}}\widetilde{D}^{-\frac{1}{2}}\mathcal{H}^{l}\mathcal{W}^{l}\right), \tag{2.14}$$

where $\widetilde{\mathcal{A}} = \mathcal{A} + \mathcal{I}$ and $\widetilde{\mathcal{D}_{ii}} = \sum_j \widetilde{\mathcal{A}}_{ij}$. $\mathcal{H}^l$ is the feature representation of all nodes at GCN layer l. Interestingly, Yao et al. (2019) adapted GCN for text classification by constructing a convolution network (called TEXTGCN) from an input corpus based on word co-occurrences and relations within the corpus. Words and documents are initially represented using one-hot encoding. Subsequently, TEXTGCN learns embeddings for both words and documents, as supervised by ground-truth labels of the documents. In our study, we employed a GNN with an attention mechanism (GAT) for learning word representations. We provide more details of our approach (I-AID) in Chapter 5.

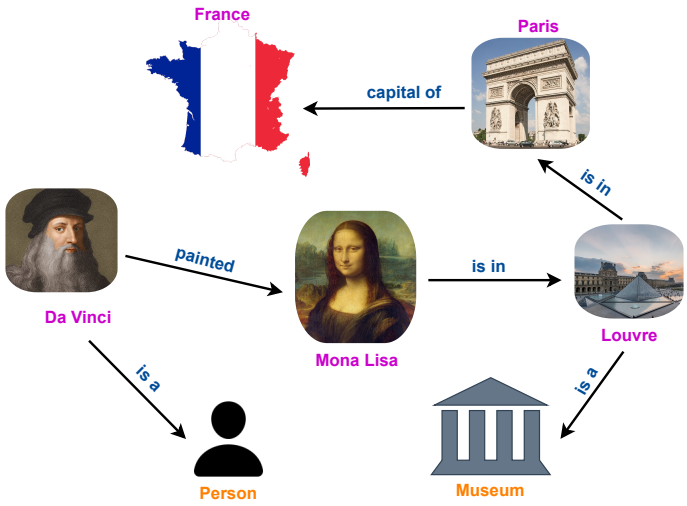


Figure 2.14: An example of *Mona Lisa* knowledge graph

2.5 Knowledge Graphs

In recent years, knowledge graphs (KGs) have become the foundation of many knowledge-based information systems (Hogan et al., 2021). In knowledge representation and reasoning, a knowledge graph serves as a knowledge base that employs a graph-structured model (i.e., topology) to integrate data from various sources. It describes information about entities of interest in a given domain (e.g., people, places, or events) and their relations. Figure 2.14 illustrates an example of a knowledge graph about "*Mona Lisa*". Each data instance is encoded with RDF[4] triples in the form (*subject, predicate, object*). For example, (*Da Vinci, painted, Mona Lisa*) encodes the relation *painted* between the entities *Da Vinci* and *Mona Lisa*.

There are many applications of knowledge graphs in both research and industry (Ji et al., 2021). In the machine learning domain, KGs are commonly employed to i) enrich data representation by adding structural information (e.g., from linked data), ii) understand the underlying semantics of data or on a broader scale, enable the development of intelligent systems, and iii) enable data integration and fusion from heterogeneous sources. In our studies, we used the ConceptNet knowledge graph (Speer et al., 2017) to enrich the representation of disaster-related tweets. We also used the DBpedia (Auer et al., 2007) and BabelNet (Navigli and Ponzetto, 2012) graphs to generate absent keyphrases for disaster summarization. We provide a brief overview of these graphs as follows:

[4] https://www.w3.org/RDF/

- **DBpedia**[5] is one of the most popular knowledge graphs in the Linked Open Data cloud (Auer et al., 2007). It is created by extracting structured information from Wikipedia, such as Wikipedia infoboxes. Specifically, the infoboxes types are mapped to DBpedia ontologies (i.e., DBpedia Classes), and the infobox attributes are assigned to the properties of a DBpedia ontology. According to its latest release in 2020, DBpedia contains 6 million entities, 9.5 billion facts (represented as RDF triples), and its ontology schema contains 760 classes. Since its release in 2007, DBpedia has been widely used in various semantic applications. In our studies, we exploited DBpedia to obtain relevant terms based on keyphrases in the input text (see Chapter 7 for more details).

- **BabelNet**[6] is a multilingual knowledge graph that combines Wikipedia and WordNet for cross-lingual entity disambiguation (Navigli and Ponzetto, 2012). BabelNet provides a large collection of encyclopedic dictionaries, covering both lexical and factual knowledge. It also includes a semantic network (ontology) that links concepts and entities within a comprehensive semantic network containing about 20 million synsets and around 1.4 billion word senses in 500 languages. The BabelNet system builds on the WordNet model to incorporate multilingual lexicalizations, based on the notion of synsets (for synonym sets). Similar to DBpedia, we exploited BabelNet in our studies to obtain relevant terms for generating absent keyphrases (see Chapter 7 for more details).

- **ConceptNet**[7] is a knowledge graph that connects words and phrases of a natural language with semantic relations. For instance, in the sentence "*Da Vinci painted the Mona Lisa*", a relation labelled "*painted*" links the words "*Da Vinci*" and "*Mona Lisa*". The ConceptNet knowledge is derived from various sources that include expert-created resources, crowdsourcing, and purpose-driven games. The primary goal of ConceptNet is to capture the commonsense knowledge involved in understanding language, enabling NLP applications to better interpret the meanings of words. In our studies (see Chapter 4), we used ConceptNet to obtain the embedding vectors of named entities in tweets, resulting in better performance of disaster prediction than conventional word embeddings (e.g., Skip-gram).

In the crisis informatics domain, knowledge graphs can help to improve disaster responses by providing a coherent, structured, and accessible source of information about entities and relations related to disasters (Purohit et al., 2019). This information can cover the types of disasters that can occur, the locations where they

[5] https://www.dbpedia.org
[6] https://babelnet.org
[7] https://conceptnet.io

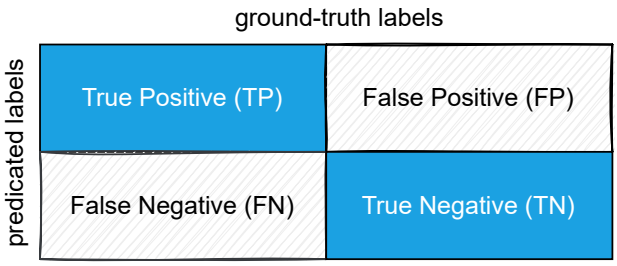


Figure 2.15: Confusion matrix for *Precision*, *Recall* and F_1

are most probable, the resources that can be used for response, and the potential impacts of disasters on communities and infrastructure. Accordingly, a knowledge graph can help disaster managers to better understand the risks and challenges they encounter and make informed decisions about how to deal with disasters. For example, a knowledge graph could help to locate the most vulnerable areas in a community or to plan evacuation routes and distribute resources according to the type and severity of a disaster.

2.6 Performance Evaluation

2.6.1 Accuracy, Precision, Recall, F_1

In this section, we introduce the main metrics used to evaluate the performance of our approaches in classifying social media data: *Precision*, *Recall*, and *F1-score* (Powers, 2020). These metrics measure how well a model can predict the correct information types for a given set of tweets, compared to their true types. Figure 2.15 illustrates the schema of a confusion matrix, which is used to represent the results of these comparisons. A confusion matrix consists of four counts, defined as follows:

- **TP (true positive)**: The number of samples that are correctly predicted and labelled as *positive*.
- **TN (true negative)**: The number of samples correctly predicted and labelled as *negative*.
- **FP (false positive)**: The number of samples that are incorrectly predicted as *positive*, but are actually labelled as *negative*.
- **FN (false negative)**: The number of samples that are incorrectly predicted as *negative*, but are actually labelled as *positive*.

Accuracy: The ratio of correctly predicted labels to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}. \tag{2.15}$$

Precision: The fraction of predicted positive labels that are correct.

$$Precision = \frac{TP}{TP + FP}. \tag{2.16}$$

Recall: The fraction of actual positive labels that are correctly predicted.

$$Recall = \frac{TP}{TP + FN}. \tag{2.17}$$

The F_1 metric is used to summarize the precision and recall of a model. It is computed as the harmonic mean of precision and recall, i.e., it gives more weight to low values.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{2.18}$$

Additionally, F_1 can be calculated in three distinct ways: i) *micro-average*, which derives scores by counting all true positives, false negatives, and false positives, ii) *macro-average*, which calculates metrics individually for each class and then computes their unweighted average. This does not take into account class imbalances, and iii) *weight-average*, which first calculates precision and recall for each class, then computes their weighted average by considering the number of true examples for each class.

2.6.2 Alert Accumulative Worth

Formerly, we used *Precision*, *Recall*, and F_1 score to measure the performance of detecting relevant information from social media data. However, these metrics are not suitable for identifying actionable information, for which alerts should be triggered. To overcome this limitation, McCreadie et al. (2019) proposed a new metric, Alert Accumulative Worth (AAW), which captures the effectiveness for alerting messages. Moreover, the authors introduced a component of AAW, called

highPriorityWorth, which focuses only on critical tweets. We calculate the scores of tweets that should generate alerts as follows:

$$hightPriorityWorth(t) = \begin{cases} \alpha + ((1-\alpha)).(ActCScore(t) + NActCScore(t))) & p_t^s >= 0.7 \\ -1 & otherwise. \end{cases} \tag{2.19}$$

$$ActCScore(t) = \gamma.\frac{|ActC_t^s| \cap ActC_t^a|}{|ActC_t^s \cup ActC_t^a|}. \tag{2.20}$$

$$NActCScore(t) = (1-\gamma).\frac{|NActC_t^s| \cap NActC_t^a|}{|NActC_t^s \cup NActC_t^a|}. \tag{2.21}$$

$$\gamma = \begin{cases} \lambda & |ActC_t^a| \\ 0 & otherwise. \end{cases} \tag{2.22}$$

For tweets that should not generate alerts, we use the following score:

$$lowPriorityWorth(t) = \begin{cases} argmax(1 - log(\frac{\sigma}{2} + 1), -1), & p_t^s >= 0.7 \\ ActCScore(t) + NActCScore(t) & otherwise. \end{cases} \tag{2.23}$$

The AAW metrics is defined as:

$$AAW = \frac{1}{2}\sum_{t \in T} \begin{cases} \frac{1}{|T_{high/critical}|}.highPriorityWorth(t) & t \in T_{high/critical} \\ \frac{1}{|T_{low/medium}|}.lowPriorityWorth(t) & otherwise. \end{cases} \tag{2.24}$$

where $ActC_t^s/ActC_t^s$ are the actionable/non-actionable categories assigned to a tweet t by the system s and p_t^s is its priority score. α is a hyperparameter (default value= 0.3) that gives a reward for a correct alert regardless of the tweets categories. T is the set of tweets to be evaluated, and $T_{high/critical}$ is the set of tweets that are labelled high or critical by human assessors. δ is the count of false alerts since the last true alert. A false alert occurs when a tweet that is not in $T_{high/critical}$ has a p_t^s score >= 0.7. δ is reset to 0 each time the system gives a p_t^s score >= 0.7 to a tweet in $T_{high/critical}$ (true alert). This reflects the user's trust in the system over time.

The AAW value ranges from -1 and $+1$, where the higher values indicate better performance in alerting critical tweets. This metric was proposed by TREC-IS[8] for detecting tweets with actionable information or alerting emergency responders to urgent situations.

[8] http://dcs.gla.ac.uk/~richardm/TREC_IS/

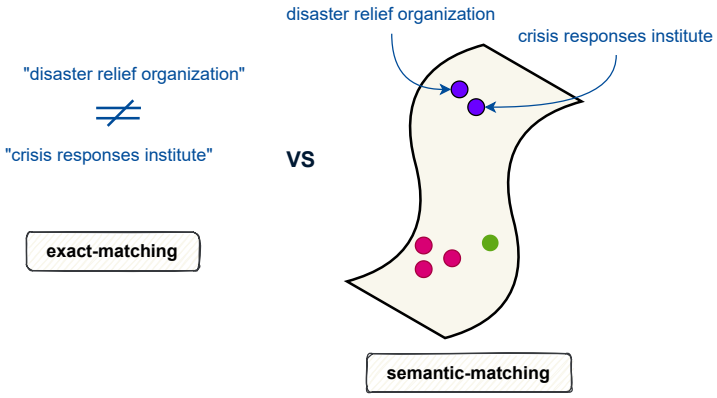


Figure 2.16: Exact matching vs Semantic matching keyphrases

2.6.3 Keyphrase Extraction Evaluation

Summarizing tweets has proven to be a useful technique to gain insights and facilitate awareness of emerging disasters (Imran et al., 2018). There are two main methods for tweet summarization: i) extractive summarization, which selects salient keyphrases from tweets and ii) abstractive summarization, which generates human-like summaries. In our study, we focus on extractive summarization by extracting relevant keyphrases from a collection of disaster-related tweets. An important aspect of this task is the evaluation of the quality of extracted or generated keyphrases performance of extracted keyphrases. The main criterion for this evaluation is comparing the number of correctly extracted keyphrases and the number of ground-truth keyphrases. Therefore, different strategies have been developed for matching keyphrases. In the following, we present the matching strategies, including our semantic match metric, that we use in the evaluation of extracted keyphrases:

- **Manual Match**: This approach involves domain experts to judge the accuracy of keyphrases returned by a system. However, this type of evaluation is not only expensive but also lacks subjectivity (Zesch and Gurevych, 2009). Accordingly, researchers have explored automatic metrics for comparing predicted keyphrases with ground-truth ones.
- **Exact Match**: This method uses string similarity to compare ground-truth and predicted keyphrases. In most cases, stemming is applied to determine if two keyphrases are the same or not (Papagiannopoulou and Tsoumakas, 2020).
- **Partial Match:** This is a more flexible evaluation that compares all ground-truth keyphrases with all extracted ones (Rousseau and Vazirgiannis, 2015). While this

assessment can evaluate syntactic correctness, it cannot handle more complex issues, such as the presence of overlapping keyphrases.

- **Semantic Match:** This is a novel approach that compares predicted and ground-truth keyphrases based on their embedding representation (Zahera et al., 2022). For example, if two keyphrases are semantically similar, such as *"Typhoon"* and *"Hurricane"*, these keyphrases are not considered as a *match* using the previous metrics.

2.7 Datasets

Collecting data from social media is essential for the development of disaster models. Over the past decade, researchers have collected, analyzed, and processed large amounts of disaster-related tweets. Using these datasets, they can build, evaluate, and deploy disaster management systems that are based on real-world data. In our study, we focused on disaster datasets in English since our approaches are designed for *monolingual* data. We also provide a description of the most used datasets in the literature for event detection, information extraction, and situational awareness improvement.

- **CrisisLexT26** (Olteanu et al., 2014): This dataset contains disaster-related tweets from 26 events, covering natural disasters such as *earthquakes*, *wildfires*, and *floods*, as well as human-induced disasters such as *shootings* and *train crashes*. The number of tweets per event varies from $1,1K$ to $157,5K$, resulting in approximately $285K$ tweets in total. Paid workers annotated these tweets using the CrowdFlower[9] platform based on three criteria: *informativeness*, *type of information*, and *tweet source*. This dataset can be used to benchmark the performance of disaster models in filtering informative tweets (Khare et al., 2018b).

- **TREC-IS** (McCreadie et al., 2019): This is the largest annotated disaster dataset, containing $17,6k$ tweets collected from various crisis events (e.g., *earthquakes*, *hurricanes*, or *public shootings*) using hashtags and keywords. Human annotators assigned labels to tweets according to a multi-layer ontology of information types. We used this dataset to benchmark the performance of classifying disaster-related tweets and identifying actionable information (Zahera et al., 2019a, 2021).

[9] http://faircrowd.work/platform/crowdflower/

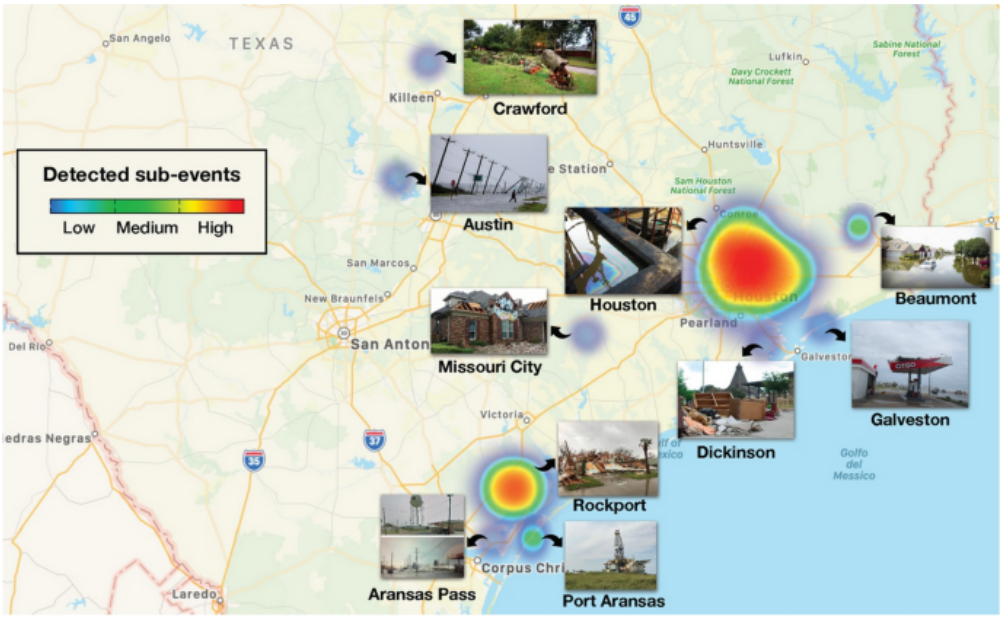


Figure 2.17: An example of crisis events detection using tweets during hurricane HARVEY. Source from (Belcastro et al., 2021)

- **CrisisMMD** (Alam et al., 2018): This is the first multimodal dataset that contains thousands of tweets and images collected during seven major disasters, including *earthquakes*, *hurricanes*, *wildfires*, and *floods*. Alam et al. (2018) collected this dataset for multimodal tasks such as natural language and image processing. This dataset can be leveraged to learn a joint embedding space of tweets' text and images, which can be then applied to text-to-text and image-to-text retrieval tasks. We also find this dataset useful for other disaster management tasks, such as estimating post-disaster damages.
- **Covid-19 Tweets**: Recently, Buntain et al. (2020) introduced a disaster dataset about the Covid-19 pandemic. The dataset contains $7,590$ tweets collected from different regions around the world. These tweets are annotated with one or more information types (e.g., *ServicesAvailable*, *Advice*, *GoodsServices - in total 12 types*) similar to the TREC-IS dataset.

2.8 Applications

Information and communication technologies (ICTs) play a vital role in disaster management, especially in facilitating the sharing of information and coordinating with authorities during crisis events. Various applications have been developed

to leverage the potential of ICTs for disaster response and recovery. For example, Figure 2.17 illustrates how different crisis events were detected during hurricane Harvey in 2017 using a social media analysis tool (Belcastro et al., 2021). Some of the most popular applications designed for use during disasters are:

- **TweetTracker:** This application analyzes tweets related to a crisis event from various perspectives, such as location, keywords, and sentiment. It provides situational updates for first responders and humanitarian organizations by filtering and extracting relevant information from the large volume of social media data. More information can be found on the official website at http://blogtrackers.fulton.asu.edu/#/.

- **Facebook Crisis Response:** This is a central hub for all of Facebook's safety-related tools. It enables users to share situational updates, mark themselves safe, give or find help, and raise money during crisis events. It also aggregates timely information from various sources, such as articles, photos, and videos from public posts. Further details can be found on the official website at https://www.facebook.com/about/crisisresponse/.

- **Emergency Situation Awareness (ESA):** This web-based tool enhances situational awareness, especially during earthquakes. ESA collects, filters, and analyzes tweets to extract valuable information for emergency managers. Its features include event detection, text classification, clustering, and geotagging. It can help identify tweets containing critical information, such as damage reports, requests for help, or offers of assistance. More information can be found at the official website at https://www.csiro.au/en/research/technology-space/ai/emergency-situation-awarenes.

- **Twitcident:** This situational awareness tool employs semantic methods to filter crisis-related tweets. Specifically, Twitcident recognizes named entities and uses external resources to retrieve attribute-value pairs for relevant tweets. Additional details can be found at the official website at https://wis.st.ewi.tudelft.nl/twitcident/.

- **Artificial Intelligence for Disaster (AIDR):** This is a web application that monitors disasters and analyzes relevant information shared on Twitter. The AIDR application collects tweets related to a crisis event and categorizes them using a crowdsourcing platform into different types, such as donations, damage, etc. It also generates reports about emerging events and classifies messages based on geographical information, i.e., crisis mapping. Further information can be found at the official website at http://aidr.qcri.org/.

- **Tweedr:** This application collects crisis-related data from Twitter using keywords and regional queries. It extracts actionable information using clustering and classification techniques. Several classification algorithms, including SVM, sLDA, and logistic regression are employed to identify tweets reporting losses or damage. More details can be found on the GitHub repository at https://github.com/dssg/tweedr.
- **CrisisTracker:** This application utilizes social media to track crisis-related keywords and cluster-related stories. Volunteers manually tag stories according to the categories of disaster reported in the associated stories. The system's success depends on the number and motivation of the volunteers who are assigned to accurately label each story. More information can be found on the official website at https://crisistracker.org/.

2.9 Summary

This chapter provided a comprehensive overview of the fundamental aspects of using social media in disaster situations. We started by introducing the main concepts and terminology of disaster management. Then, we explored how to collect and filter social media data, focusing on Twitter as a prominent source of disaster-related information. We discussed the challenges of processing tweets, which were often informal, limited, and noisy. For example, tweets might contain abbreviations, slang, and emojis that made them hard to understand. To address these challenges, we presented advanced techniques for processing tweets effectively. We emphasized the importance of feature representation for developing efficient models that could analyze social media data. We reviewed various approaches to represent features from social media. We pointed out the limitations of traditional methods such as TF-IDF and bag-of-words for dealing with unstructured and noisy data like tweets. We also showed that contextualized embeddings, such as BERT, can overcome the limitations of static embeddings by encoding semantic and syntactic information from text. BERT learns from the bidirectional context of a word in a tweet, which enables it to capture situational meanings of words and produces more rich representations of tweets than static embeddings.

We also described the deep learning models used in our experiments, such as Convolutional Neural Network, Long Short Term Memory, Bidirectional Long Short Term Memory, and Graph Attention Network. Furthermore, we gave a brief overview

of the datasets and knowledge graphs that we used in our studies, such as CONCEPTNET, DBPEDIA, and BABELNET. Moreover, we explained how to evaluate the performance of machine learning models in various tasks, such as multi-label tweet classification, detection of actionable tweets, and extractive summarization. Finally, we listed some real applications that had been developed for disaster communication and situational awareness.

3 State-of-the-Art

This chapter discusses state-of-the-art approaches that have been developed for leveraging social media in disaster situations. Most approaches focus on obtaining and processing disaster-related tweets. We organize this chapter into three sections: i) Early Detection of Crisis Events, ii) Filtering Informative Tweets, and iii) Summarization of Events for Situational Insights. In each section, we present the main approaches and discuss their limitations.

3.1 Early Event Detection

One of the key aspects of crisis management is to identify emerging events and provide timely warnings so that appropriate actions can be taken to reduce the impact and subsequent damage. Remarkably, social media contains all essential information for detecting events through shared text, images or both; thus, serving as a rapid event detector, or so-called *social-sensing* from the crowd (Aiello et al., 2013; Imran et al., 2018; Sakaki et al., 2012). Several studies have been conducted to detect events on social media, especially on Twitter. These approaches can be mainly categorized into i) *traditional approaches* which rely on statistical and linguistic features, and ii) *state-of-the-art approaches* which employ deep neural models for detecting events.

3.1.1 Traditional Approaches

Social media platforms generate a massive number of messages related to emerging events, which creates a burst of associated keywords (i.e., social media trends). Traditional approaches mainly rely on linguistic-based features (e.g., term frequencies, topic detection) or clustering techniques to identify the most relevant and salient keywords for each event. We summarize these approaches as follows:

- **Term-based approaches** utilize statistical features such as term frequency features (TF) peakiness and trending scores to detect events based on the most frequent

words (i.e., burst words). For example, Aiello et al. (2013) developed an event detection model that computes the frequency-inverse document frequency (TF-IDF) for bigrams tokens and ranks events according to the most frequent keywords. However, this method has some limitations, such as assuming a fixed number of events over a specific period and failing to detect emerging events. Moreover, Sakaki et al. (2012) analyzed a set of disaster tweets using keywords, word counts, and context. They developed a probabilistic spatial-temporal model that treats each Twitter user as a social sensor and applied Kalman and Practical filtering techniques (Chen et al., 2003) to estimate the location and trajectory of events. The experimental results showed that their approach can detect an earthquake with high probability and achieve an accuracy of 96% according to the Japan Meteorological Agency (JMA) intensity scales. Similarly, Mathioudakis and Koudas (2010) introduced the *TwitterMonitor* framework that detects events on Twitter in real time and provides informative analytics. They extracted trend keywords and grouped them based on their co-occurrences using a context extraction algorithm.

- **Topic modelling approaches** aims to extract latent topics by modelling a document as a generative process and inferring the topic distributions using optimization algorithms. One of the most popular and widely used topic modelling methods is Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which has been successfully applied in various applications, especially for detecting emerging topics on social media. LDA assumes that a document $\mathcal{D}$ consists of a mixture of k topics and randomly assigns $\mathcal{W}$ words to these topics. To determine the optimal value of k, LDA performs a series of iterations until it finds the best trade-off between perplexity and high log likelihood.

- **Clustering approaches** have also contributed to detecting events on social media. For instance, Li et al. (2012) proposed the *Tweetvent* approach, which extracts segments of tweets that show a burst of activity within a fixed time window and clusters them using the Jarvis-Patrick algorithm (Jarvis and Patrick, 1973). Then, each cluster is evaluated using Wikipedia articles to identify realistic events and their related keywords. Likewise, Parikh and Karlapalem (2013) detected events by extracting keywords based on bigrams tokenization and ranking them according to their frequency. Then, they clustered the keywords that belong to the same event based on similarity. Zhang et al. (2021) developed a real-time detection system that monitors nearby events. Their system consists of four components: text filtering, text representation, deep clustering, and event merging. It filters out irrelevant messages, represents event messages using entities and words and applies the DBSCAN clustering algorithm for event detection. Similarly, Dang et al. (2016) used the DBSCAN clustering for event detection and employed

Dynamic Bayesian Networks, which combine information between tweets and users to detect emerging keywords and rank them based on their co-occurrence.

3.1.2 State-of-the-art approaches

Deep learning models have recently achieved significant performance in different NLP tasks, including event detection from short text (e.g., tweets). We categorize the state-of-the-art deep models into two groups:

- **Standalone models:** One approach to event detection is using standalone models that only consider the text of the tweet. For example, Burel et al. (2017b) extended traditional bag-of-word models by incorporating bag-of-concepts extracted from knowledge graphs, such as*WordNet*, and *DBpedia*. However, the authors represented the presence of concepts as a vector of indices within a concept space, which may not capture the semantic relations among concepts. Another approach is to use convolutional neural networks (CNNs) which have demonstrated remarkable performance in classification tasks. Wang et al. (2017) proposed a CNN model to classify short text. The proposed approach represents text as a bag-of-concepts based on a taxonomy knowledge base. Then, the embeddings of words and concepts are combined and fed to the CNN model to infer text classes. In contrast, we leverage embeddings from the ConceptNet knowledge graph that captures semantic representations of concepts (i.e., entities) and their relationships. Additionally, Burel et al. (2017b) proposed a Dual-CNN model with an extra semantic layer that uses the conceptual semantics of words for fine-grained event detection. The authors also investigated how named entities in tweets can be extracted and utilized in conjunction with their corresponding semantic concepts. The experimental results demonstrated that the proposed approach outperforms the traditional models (i.e., without semantic layer) in detecting events from disaster-related tweets of 26 different events.

- **Joint models** have attained superior performances in various NLP tasks in recent years (Ouyang and Wang, 2013). For instance, Chowdhury et al. (2019) proposed a jointly trained model with two recurrent neural networks to extract keyphrases from disaster-related tweets. By training the recurrent networks jointly, the proposed approach achieved significant performance in detecting salient phrases compared to different baselines. Moreover, Zheng et al. (2017) demonstrated that the joint learning of two deep models not only learns feature representation from users and items data but also collaborates with each other to boost the rating prediction performance in a recommendation system task. Inspired by

these approaches, we propose a joint learning model that employs social media and environmental data for an early detection of disaster events.

3.2 Filtering Informative Tweets

The literature on crisis informatics has extensively explored the problem of finding relevant information from large-scale crisis data (Imran et al., 2013; Stowe et al., 2016). This problem poses significant challenges, as emergencies generate a massive volume of social media content that is often unstructured and noisy (Landwehr and Carley, 2014). As we discussed in Section 2.2, this content requires advanced analysis to extract meaningful information for crisis response. Several approaches have tackled this problem, ranging from *traditional machine learning* to *deep learning techniques*.

3.2.1 Traditional Approaches

These approaches typically use *handcrafted*, *statistical* or *linguistic* features to filter out irrelevant data. Keyword matching is a common technique to select relevant information based on the content and informativeness of tweets (Cobo et al., 2015; Olteanu et al., 2014). For example, Guan and Chen (2014) classified disaster-related tweets using pre-defined keywords and hashtags. They applied their method to a collection of tweets about hurricane SANDY and investigated the correlation between people's Twitter activities and the damages caused by the hurricane. Similarly, Avvenuti et al. (2014) developed a social media-based system for earthquake detection that monitors tweet interactions. They extracted features based on URLs, usernames, and bag-of-words to train their system. They also devised a method for detecting spikes in tweet volume within a specific time window. Sakaki et al. (2010) analyzed Twitter data for real-time earthquake detection using an SVM classifier to remove irrelevant tweets. They then built a probabilistic model based on the Poisson process for estimating the occurrence time of an earthquake using temporal analysis. However, these systems have some drawbacks, such as the need to define a set of features beforehand, which may affect the overall performance of the system. Although traditional approaches have achieved satisfactory results in classifying the relevance of crisis data, they have two main limitations: i) their performance deteriorates when applied to classifying "unseen" data (Khare et al., 2018a), *i.e., the terms that appear in the test data but not in the training data* and ii) the high diversity of social media data makes these approaches unsuitable for learning meaningful

features (Imran et al., 2018). Therefore, neural approaches have been proposed to address this challenge, inspired by the recent advances in deep learning. In the next section, we provide an overview of state-of-the-art approaches that have been recently proposed for classifying disaster-related data.

3.2.2 State-of-the-art Approaches

Different deep learning architectures (e.g., CNN, LSTM, GAT) have been designed to capture rich representation from various sources (see Section 2.4 for more details). One of these sources is social media, which provides valuable information during disasters. However, analyzing social media data poses several challenges, such as noise, informality, and diversity. Therefore, researchers have explored different neural architectures and semantic features to enhance the classification of disaster-related tweets. For instance, ALRashdi and O'Keefe (2019) compared the performance of two neural architectures (CNN and BiLSTM) with domain-specific and GloVe embeddings for tweet classification. They found that BiLSTM with domain-specific embeddings achieved the best results. Similarly, Khare et al. (2018a) and Burel et al. (2017a) leveraged semantic features extracted from knowledge graphs (e.g., CONCEPTNET, BABELNET, DBPEDIA) to enrich the representation of tweets. Khare et al. (2018a) used both statistical and semantic features as input to a support vector machine classifier, while Burel et al. (2017a) integrated semantic annotations into a wide and deep model that combines a CNN with a generalized linear model. Both studies reported improved performance compared to baseline models that rely only on textual features. However, most of the existing works focus on the binary classification of tweets (relevant vs. irrelevant), which limits the granularity and usefulness of the extracted information. To address this limitation, more recent studies (e.g., (McCreadie et al., 2019; Olteanu et al., 2015)) have proposed annotation schemes that assign multiple fine-grained labels to tweets, such as location, information source, and people's behaviours. These labels can provide more detailed and actionable insights for crisis management and response. To facilitate the development and evaluation of multi-label tweet classification, initiatives such as CrisisNLP (Imran et al., 2016) and TREC-IS (McCreadie et al., 2019) have been launched, which provide large-scale datasets and shared tasks for crisis monitoring on social media platforms.

On the other hand, previous works (Sriram et al., 2010) considered feature engineering and model training as separate subtasks, allowing for the use of pre-trained embeddings as features on the fly or fine-tuning models on domain-specific datasets. Recent advances in deep learning have enabled end-to-end training approaches

that can learn from raw text without manual feature extraction (Miyazaki et al., 2019). Moreover, attention mechanisms (Vaswani et al., 2017) have been shown to improve the representation and interpretation of a text by focusing on the most relevant parts of the input. One of the most successful applications of attention is BERT (Devlin et al., 2019), a pre-trained language model that can be fine-tuned for various text classification tasks. For example, Liu et al. (2021) proposed CRISISBERT, a fine-tuned BERT model for classifying crisis data, which achieved outperforming results on several benchmark datasets, outperforming the baselines by up to 8.2% and 25.0% for detection and recognition tasks, respectively. However, BERT embeddings are limited by their local context within sentences, and do not capture global relations between words across the whole vocabulary (Lu et al., 2020). To address this limitation, graph-based approaches such as graph convolution network (Kipf and Welling, 2017) and graph attention network (Yao et al., 2019) have been introduced to model the semantic and syntactic connections between words in a graph.

3.3 Summarizing Disaster-related Tweets

Summarizing disaster events and their impacts is essential for obtaining insights and informing decisions. However, the vast amount of social media data generated during such events poses a challenge for effective summarization. To address this challenge, summarization techniques can be applied to condense social media data into shorter, coherent, and query-specific reports that enhance the comprehension of disaster events (Rudra et al., 2016). Two main methods exist for text summarization: i) selecting relevant phrases (*extractive summarization*) and ii) producing human-like summaries (*abstractive summarization*).

3.3.1 Traditional Approaches

Previous studies have developed unsupervised methods for keyphrase extraction, which do not require labelled data. For example, statistical approaches such as TF-IDF and YAKE (Campos et al., 2020) use statistical features (e.g., word frequencies and co-occurrences) to identify important words as candidates for keyphrases. Moreover, graph-based approaches like TEXTRANK (Mihalcea and Tarau, 2004) construct a graph representation of text, wherein words are nodes, and their co-occurrences are edges. Then, a node ranking algorithm (e.g., PAGERANK) is applied to sort words, returning the top-k words as candidate keyphrases. Bougouin et al. (2013) introduced TOPICRANK, another graph-based approach similar to TEXTRANK. This

approach first clusters candidate phrases into topics and then ranks them based on their significance with respect to their documents.

On the other hand, supervised approaches have also significantly contributed to the development of keyphrase models. For example, Chowdhury et al. (2019) designed a joint-learning model comprising two BiLSTM models. The first BiLSTM model is trained to detect keywords; this task can be seen as a binary classification (a word is labelled with 1 if it is a keyword, and with 0 if it is not). Meanwhile, the second BiLSTM is trained to predict keyphrases using a sequence labelling scheme. In the BIOE tagging schema, for instance, 'B' denotes the beginning of a keyphrase, 'E' indicates the end of a keyphrase, 'I' represents words within a keyphrase (i.e., in-between 'B' and 'E'), and *'O'* indicates words outside a keyphrase. Although traditional approaches have shown considerable performances in extracting keyphrases that appear in the text (i.e., present keyphrases), they failed to generate keyphrases that do not appear in the text (i.e., absent keyphrases). In the next section, we discuss state-of-the-art approaches that have been recently proposed to address both present keyphrase extraction and absent keyphrase generation.

3.3.2 State-of-the-art Approaches

Recent studies have demonstrated that embedding-based models can achieve high performances in keyphrase extraction tasks. For example, the EMBEDRANK (Bennani-Smires et al., 2018) approach uses part-of-speech tags to identify candidate keyphrases from an input document. It relies on a pre-trained embedding model to represent both phrases and their corresponding document as high-dimensional vectors. Then, candidate keyphrases are ranked based on their Cosine similarity scores with respect to the embedding vector of the document. However, pre-trained language models are not capable of generating absent keyphrases that are not in their vocabulary. Moreover, Liang et al. (2021) observed that embedding-based models only capture local information about words within a sentence range. To overcome this limitation, they developed a jointly trained model that incorporates both the global and local contexts of a document. In the global view, their approach represents candidate keyphrases and an input document as high-dimensional vectors in the same semantic space. A similarity score between each candidate keyphrase and the document is then computed to identify relevant keyphrases. In terms of the local context, the authors constructed a graph structure based on the document context, where nodes are candidate keyphrases and edges are similarities between them. Finally, all keyphrases are ranked based on global and local information.

Most previous approaches have relied on *sequence-to-sequence* models —with an encoder-decoder architecture— to generate absent keyphrases (Chen et al., 2019). These models can decode not only keyphrases that appear in the text but also those that are absent, i.e., not explicitly mentioned. However, generating absent keyphrases requires additional mechanisms to improve the performance of these models. For example, Ye et al. (2021) applied a *graph neural network* to capture knowledge from related references in a scholarly dataset, while Wang et al. (2019) employed a *neural topic model* to expand the context of the decoder component for generating more diverse and relevant absent keyphrases. It is noteworthy that Zhao et al. (2021a) achieved outperforming results in extracting keyphrases by dividing this task into two sub-tasks: *present keyphrase extraction* and *absent keyphrase generation*. Specifically, the authors proposed a multi-task approach to *select*, *guide*, and *generate* keyphrases. In the *selector* module, a BiLSTM is used to predict whether a sentence contains a keyphrase or not. A *guider* network is then employed to capture information from the attention mechanism and memorize the predictions of the *selector* module. Finally, this information is fed to the *generator* network to generate absent keyphrases by selecting words from both the source text and a pre-defined vocabulary. Besides these supervised approaches, some unsupervised methods have also achieved promising results in generating keyphrases without the need for training data. For instance, Shen et al. (2022) observed that many keyphrases absent from a target document appear in other related documents. Therefore, they constructed a *phrase bank* of all keyphrases in a corpus and identified present keyphrases in relevant documents as candidates for absent keyphrases for the target document. They also used present keyphrases as *sliver labels* to train a sequence-to-sequence model. Finally, all keyphrases (both present and absent) are ranked based on their lexical and semantic similarity with respect to their corresponding document.

3.4 Summary

In this chapter, we reviewed traditional and state-of-the-art approaches that are relevant to our studies and organized them into three main sections: i) early detection of crisis events, ii) filtering informative social media data, and iii) summarizing social media data for situational insights. Each section corresponded to one of the studies conducted in this thesis. We also discussed how traditional and state-of-the-art approaches had been employed to process social media data during disaster situations. We observed that traditional methods relied on linguistic and statistical features (e.g., word frequency, co-occurrence, pre-defined terminology) to identify events,

categorize relevant tweets, and summarize crisis events. On the other hand, state-of-the-art approaches employed neural models (e.g., CNN, RNN, BiLSTM) as black boxes, reducing the effort for feature engineering. There was a trade-off between traditional and state-of-the-art approaches for processing social media data. While traditional approaches could process social media data rapidly, they required feature engineering to extract meaningful data representations. In contrast, deep learning methods often required sufficient resources (e.g., GPU memory) to efficiently train deep learning models and achieve superior results. In the following chapters, we discuss the details of our approaches and findings regarding the application of social media data to improve situational awareness during disasters.

4 Joint Learning from Environmental Data and Social Media

This chapter addresses the first and second research questions ($\mathbf{Q}_1$ and $\mathbf{Q}_2$, see Section 1.4) by investigating the potential impact of social media in improving the detection (i.e., prediction) of disaster events. In this context, we present our approach for detecting events using a combination of social media and environmental data. The main content is based on our publication work (Zahera et al., 2019b), which represents the first study to jointly learn from social media and environmental data for disaster prediction. The author designed, implemented, and evaluated the approach presented herein, and co-wrote the aforementioned paper. We provide a review of the studies related to the approach presented in this chapter in Section 3.1.

4.1 Overview

Accurate disaster prediction and early warnings are crucial in mitigating the impact of disasters and minimizing the resulting damage (Glade and Nadim, 2014). Despite significant improvements in forecasting and warning systems, there are several factors that continue to limit the accuracy of the current prediction algorithms. These factors include incomplete data from monitoring equipment, and the highly dynamic nature of natural hazards and their impacts (Reese, 2016). Social media has played an increasingly significant role in disaster management and communication (Reuter and Kaufhold, 2018). During disasters, people use social media platforms (e.g., Twitter) to express their feelings, ask for help, and contribute to disaster relief efforts. Consequently, a significant body of research has leveraged disaster-related information shared on social media to reduce disaster impact and facilitate faster responses (Houston et al., 2015). For instance, Sakaki et al. (2010) analyzed user tweets during 25 different earthquakes in Japan, demonstrating the reliability of social media users as a source of real-time situational updates during disasters. Additionally, decision-makers employ social media to rapidly engage

with the public. For example, during typhoon PABLO in 2012, local authorities in the Philippines encouraged people to use the hashtag `#pabloph` to obtain or share on-site updates about the typhoon (O'Glasser et al., 2020). Such correlations are valuable in supporting decision-makers in emergency response processes. Previous works utilized data mining techniques to extract correlations between social media data and crisis events. For instance, Anam et al. (2018) applied wavelet analysis to monitor disaster progression from social media data. Their findings showed that wavelet-based features can preserve text semantics and predict the total duration for localized, small-scale disasters.

In our studies, we propose an end-to-end learning model for classifying the intensity of typhoons, also known as typhoon category or class (Chen et al., 2012). Our approach leverages both environmental data and social media posts (*tweets*) as sources of information. We were inspired by previous works (Qin et al., 2016; Tompson et al., 2014) that demonstrate the benefits of joint learning of multiple models for various tasks. Our approach contains two models that are trained jointly: the first model (called *Feature Extractor*) analyzes typhoon-related tweets and computes statistical features, such as tweet volume and sentiment variances. To capture tweet sentiments, we use semantic-enriched word embeddings, in which *entities* are recognized and represented by semantic vectors from the CONCEPTNET[1] knowledge graph. The second model (called *Typhoon Classifier*) takes a concatenated vector of features extracted by the first model and environmental data as input. Both models share a common loss function and are optimized using the same gradient descent. Furthermore, we explored various architectures based on Deep Neural Networks (DNN), Deep Convolutional Networks (CNN) and Recurrent Neural Networks (e.g., RNN, LSTM and BiLSTM) as baselines.

To evaluate the performance of our approach, we used two real-world datasets: i) *Typhoon Environmental Data*, obtained from the *Joint Typhoon Warning Center (JTWC)*[2], which contains climate change measurements (e.g., wind speed and sea-level pressure) before, during and after typhoon landfall; and ii) *Typhoon Social Media Data*: we collected typhoon-related tweets using keyword-based queries during $2006 - 2018$. Our results show that jointly trained models outperform standalone baselines in disaster prediction. We summarize the main contributions as follows:

- To the best of our knowledge, this is the first study to leverage joint learning from social media and environmental data for classifying typhoon intensities.

[1] `https://conceptnet.io/`
[2] `https://www.metoc.navy.mil/jtwc/jtwc.html`

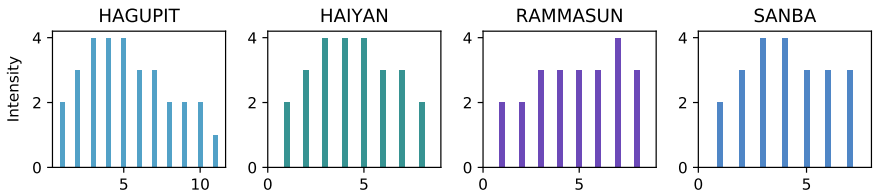
- We investigated the impact of incorporating semantic embeddings from knowledge graphs to enrich tweets representation. Our experiments demonstrate that representing named entities in tweets with their embedding vectors from ConceptNet improves disaster prediction.
- We provide a disaster dataset (named TED), which contains environmental data of different typhoons and their associated tweets up to 2018 (the last archived date by JTWC).
- We conducted extensive experiments on a real-world disaster dataset to evaluate the performance of disaster prediction. The evaluation results clearly indicate that our approach outperforms state-of-the-art baselines with a significant performance margin. The implementation of our approach is open-sourced and available at the GitHub repository.[3]

4.2 Data Analysis and Preliminaries

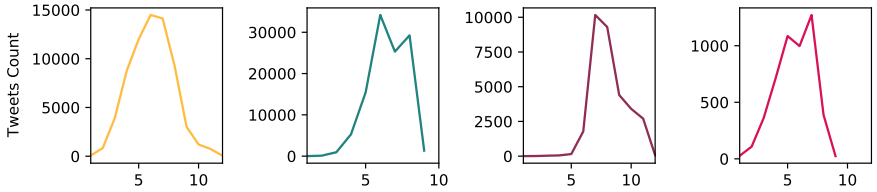
Typhoon environmental data are recorded periodically (i.e., at regular time intervals) before, during and after their occurrence. This data typically includes the typhoon's location, direction, and speed. Our research aims to detect typhoon intensity based not only on environmental data but also on human-generated data, such as social media posts, as complementary sources of information for typhoon intensity detection. Specifically, we collect and analyze tweets that are posted within the same time slots as the corresponding environmental data for four different typhoons. Inspired by previous works (He et al., 2013; Kryvasheyeu et al., 2016), we examine the volume and sentiment of tweets during different phases of the typhoons. As shown in Figure 4.1, the upper row of plots presents the variation of typhoon intensity over time, as measured by environmental data. The middle row of plots shows the tweet count within the same time slots. The lower row of plots displays the distribution of tweets with positive (in blue) and negative (in yellow) sentiment. By comparing the plots for each typhoon, we can observe a clear correlation between typhoon intensity, tweet volume and tweet sentiment. Based on these observations, we use tweet volume and sentiment as features for typhoon intensity detection.

Datasets: We used three datasets to evaluate the performance of our approach (see Table 6.2 for more details).

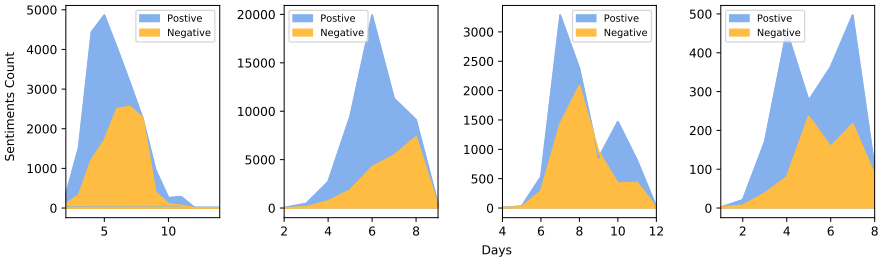
[3] https://github.com/dice-group/joint-model-disaster-prediction



(a) Typhoons intensity per day



(b) Tweets Count per day



(c) Tweets Count per day

Figure 4.1: Social media analysis during typhoons: HAGUIT, HAIYAN, RAMMASUN, and SANBA

- *JTWC Best-tracked (typhoons environmental data):* This dataset contains $3,162$ data points for 70 typhoons that occurred between $2006 - 2018$. Each data track is labelled with one of the 4 classes (*TD*: tropical depression, *TS*: tropical storm, *TY*: typhoon and *ST*: super typhoon). The dataset also provides timestamps, location, maximum wind speed (*VMAX*), wind intensity (*RAD*) and sea-level pressure (*MSLP*) for each typhoon. We preprocessed the dataset to remove noisy and corrupted data (e.g., missing values) and applied the SMOTE technique (Bowyer et al., 2011) to deal with the class imbalance problem.

- *Typhoon Tweets:* To collect typhoon-related tweets, we used keyword queries based on specific terms associated with typhoons, such as the word *typhoon* and typhoon names (e.g., HAIYAN). The official Twitter streaming API only allows free access[4] to tweets from only the past seven days. To overcome this limitation, we used the open-source library *GetOldTweets-python*[5] to obtain older tweets.

[4] https://developer.twitter.com/en/docs/tweets/search/overview
[5] https://github.com/Jefferson-Henrique/GetOldTweets-python

Table 4.1: Overview of the datasets.

Dataset	Training	Testing	Classes
JWTC Best-Track	2,529	633	4
Typhoon Tweets	1,052,599	270,364	unlabelled
Sentiment140	1,280,000	320,000	2

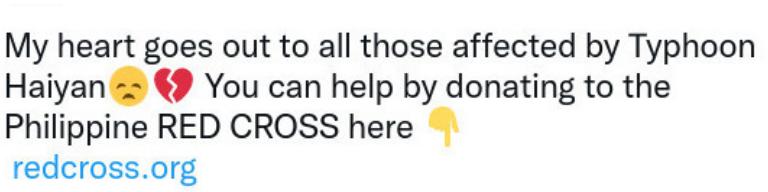


Figure 4.2: A tweet example during typhoon HAIYAN

- *Stanford NLP Sentiment140:*[6] This dataset contains 1.6 million tweets with binary sentiments (positive or negative). We used this dataset for training and evaluating our model in the sentiment analysis task.

Data Preprocessing: As we described in Section 2.2, we collected and preprocessed tweets to remove informal and noisy content. Tweet preprocessing is an essential step to ensure high-quality analysis. An example of a preprocessed tweet is shown in Figure 4.2: [*heart, goes, out, affected, [typhoon], [haiyan], can, help, donating, Philippine, [red], [cross]*], where the words in brackets are the detected named entities. Specifically, we performed the following preprocessing steps:

- *Tokenization:* We split tweets into words and converted them to lowercase letters.
- *Cleaning up:* We cleaned tweets by removing irrelevant and noisy data, such as stop words, URLs, non-ASCII characters, and usernames.
- *Entity recognition:* In this step, we detected named entities within tweets. We used the SPACY toolkit[7], which is free and open-source and provides multiple functions within a single pipeline (e.g., named entity recognition, POS tagging, dependency parsing, and pre-trained word embeddings)

[6] http://help.sentiment140.com/for-students
[7] https://spacy.io/

Table 4.2: A list of symbols used in this chapter.

Symbol	Description
$\mathcal{D}$	Labelled Typhoons Dataset
$\mathcal{X}$	Typhoons observations, i.e. measurements vector of sea level, wind speed etc.
y_i	Typhoon category (e.g., tropical depression, tropical storm, typhoon or super typhoon)
$H(x_i)$	The predicted typhoon category (final output)
$\mathcal{T}$	A collection of tweets collected during typhoons
$\mathcal{F}_\infty$	The first model in our approach (Feature Extractor)
$\mathcal{F}_\in$	The second model in our approach (Typhoon Classifier)
$\mathcal{L}$	The Cross-entropy loss function
ϕ_1	The learning coefficients (i.e., weights) of $\mathcal{F}_1$ model
ϕ_2	The learning coefficients (i.e., weights) of $\mathcal{F}_2$ model
λ	A hyperparameter that balance the joint loss functions between $\mathcal{F}_1$ and $\mathcal{F}_2$
d	The dimension of an embedding vector
w_i	A word in a tweet
s	Number of words in a tweet
B_t	a batch of input tweets
e_i	A named-entity recognized in a tweet
M	An embedding matrix of a tweet (rows correspond to words and columns are their embedding vectors)
$v-$	The variance of negative tweets
$v+$	The variance of positive tweets
c	The tweets count
$\mathcal{S}_i$	The predicted sentiment of a tweet
μ	The average of sentiments across all tweets

4.3 Our Approach

In this section, we first formulate the problem of typhoon prediction using social media and environmental data. Then, we describe the representation of tweets using semantic vectors from the ConceptNet knowledge graph. Finally, we discuss the details of our approach's components: i) *Feature Extractor* and ii) *Typhoon Classifier*. Figure 4.3 illustrates the architecture of our joint learning approach (BiLSTM+CNN).

4.3.1 Problem Formulation

Let $\mathcal{D} = \{\langle x_1, y_1\rangle \dots \langle x_n, y_n\rangle\}$ be a set of typhoon environmental data, where $\mathcal{X} = \{x_1 \dots x_n\}$ denotes the typhoon observations and $\mathcal{Y} = \{y_1 \dots y_j\}$ represents a set of typhoon categories (i.e., labels, classes). Each $x_i \in \mathcal{X}$ is an instance of a typhoon data with m features (e.g., *time-tamp, wind-speed, sea-level pressure* and *gust*), and each $y_i \in Y$ corresponds to its respective typhoon category (e.g., *tropical-depression*, *tropical-storm*, *typhoon* or *super-typhoon*). For each typhoon observation x_i, we collect related tweets shared within its time slot, referred to as $\mathcal{T}$. Furthermore,

we analyze these tweets to extract meaningful features that can indicate typhoon events. We observed that statistical features, such as *tweet volume* and *variances of tweet sentiments*, can be strong features for typhoon intensity. Finally, we combine these features with the typhoon's environmental data into a single vector. Our goal is to build a classification model that can learn features from $\mathcal{D}$ and $\mathcal{T}$ to predict the category ($y \in \mathcal{Y}$) of an emerging typhoon. We designed our classification model as a joint learning of two cascaded models: *Feature Extractor* ($\mathcal{F}_1(\mathcal{T})$) and *Typhoon Classifier* ($\mathcal{F}_2(\mathcal{D})$). To ensure joint training between $\mathcal{F}_1$ and $\mathcal{F}_2$, we combined the loss functions of both models (L_{F_1}, L_{F_2}) as follows:

$$\mathcal{L}_{joint} = \lambda_{F_1} \cdot L_{F_1} + \lambda_{F_2} \cdot L_{F_2}. \tag{4.1}$$

The $\lambda_{\mathcal{F}}$ parameter is used to balance the individual loss functions of both models. In our study, we set all λ parameters to 1. To compute the training losses, we used a Cross-entropy function as a loss function as follows:

$$\mathcal{L} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(H(x_i))) + (1 - y_i)\log(1 - H(x_i))\right], \tag{4.2}$$

where y_i and $H(x_i)$ denote the target and predicted typhoon categories, respectively, for a typhoon instance x_i.

Joint learning from social media and environmental data. Let ϕ_1 and ϕ_2 be the learning parameters (i.e., weights or coefficients) of *Feature Extractor* and *Typhoon Classifier* models, respectively. Both ϕ_1, ϕ_2 are optimized simultaneously as follows; assume two consecutive batches of training data B_t and B_{t+1}, the learning parameters in B_t are updated using the same gradient descent (*Adam* optimizer) by backpropagating the gradients to both models. In the subsequent batch (B_{t+1}), the computation of tweet features (e.g., the variance of tweet sentiments) is *adapted* based on the losses of both its outputs and the final output generated by the Typhoon Classifier.

4.3.2 Semantic-enriched Word Embeddings

By analyzing the corrected tweets (see Section 4.2) we showed that tweets volume and sentiment can be used as features for predicting typhoon intensities. We used the *Skip-gram* embedding method (Mikolov et al., 2013b) to learn embeddings based on the context of each word on two different datasets (typhoons-related tweets and Stanford-Sentiment140). However, pre-trained word embeddings models are

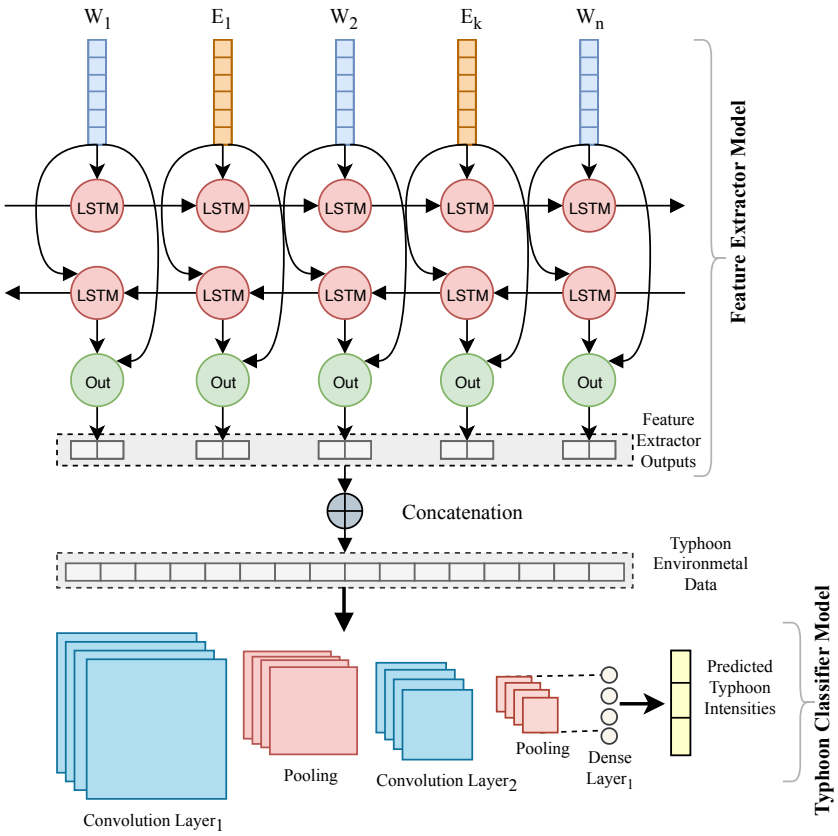


Figure 4.3: The architecture of BiLSTM+CNN. The entities (E_k) vectors (in orange) are from CONCEPTNET. The words (W_n) vectors (in blue) are from our word embeddings.

generic (trained on Wikipedia and Google News) and do not capture domain-specific knowledge. To overcome this limitation, we trained our word embeddings on domain-specific data that we collected from Twitter during different typhoons using relevant hashtags and keywords.

Given a preprocessed $t = (w_1, w_2, e_1, \ldots, e_j, w_s)$, each word is represented by its word embeddings vector in $w_i \in \mathbb{R}^d$ with d dimension. Unlike traditional word embeddings, we represent named entities $(e_1, \ldots, e_j)$, such as locations, organizations, or events, with their embedding vectors from CONCEPTNET graph, where entities and relationships are projected into the same embedding space. For each input tweet, we build an embedding matrix $M \in \mathbb{R}^{s \times |d|}$, where s is the number of words per tweet. Each row i of M represents the *Word2vec* embedding of word w_i at the corresponding position i in a tweet. Our *Word2vec* model has a dimension d of 200 and vocabulary size of $47,137$ words and $1,152$ detected entities. Since tweets have variable lengths, we fix the tweet length (s) to the average number of words per tweet in our dataset, to maintain a consistent embedding matrix. Since tweets have

variable lengths, we fix the tweet length (s) to the average number of words per tweet to maintain a consistent embedding matrix. As a result, we truncated longer tweets and padded shorter tweets with zeros.

4.3.3 Model I: Feature Extractor

The first model of our approach is designed to extract features from disaster-related tweets using a BiLSTM network. This network preserves the word sequences within the tweets and maps words to their embedding vectors using a look-up layer. The BiLSTM layer contains 64 units and a dropout rate of 0.25, followed by a dense layer with a Softmax function. For a given sequence of words $(w_1, w_2, \ldots, w_s)$, the BiLSTM network captures the context of a target word w_s from both directions (left-to-right $\overrightarrow{h}$ and right-to-left $\overleftarrow{h}$) and concatenated them into a vector $[\overrightarrow{h} \cdot \overleftarrow{h}]$. Moreover, BiLSTM associates each time-stamp with an input i_t, a memory cell m_t, a forget gate f_t and an output gate o_t. The output vector h_t is then obtained by applying the following equations:

$$
\begin{aligned}
\boldsymbol{i_{(t)}} &= \sigma(\theta_{xi}^T.x_{(t)} + \theta_{hi}^T.h_{(t-1)} + b_i). \\
\boldsymbol{f_{(t)}} &= \sigma(\theta_{xf}^T.x_{(t)} + \theta_{hf}^T.h_{(t-1)} + b_f) \\
\boldsymbol{o_{(t)}} &= \sigma(\theta_{xo}^T.x_{(t)} + \theta_{ho}^T.h_{(t-1)} + b_o) \\
\boldsymbol{g_{(t)}} &= \tanh(\theta_{xg}^T.x_{(t)} + \theta_{hg}^T.h_{(t-1)} + b_g) \\
\boldsymbol{m_{(t)}} &= f_{(t)} \otimes m_{(t-1)} + i_{(t)} \otimes g_{(t)} \\
\boldsymbol{h_{(t)}} &= o_{(t)} \otimes \tanh(m_{(t)}),
\end{aligned}
\tag{4.3}
$$

where $\theta_{xi}, \theta_{xf}, \theta_{xo}, \theta_{xg}$ are the weight vectors of the *input, forget, memory and output* gates for the input vector $x_{(t)}$. Similarly, $\theta_{hi}, \theta_{hf}, \theta_{ho}, \theta_{hg}$ are the weights vectors for the previous hidden vector $h_{(t-1)}$ and b_i, b_f, b_o, b_g are the bias terms for the four gates. σ and $\otimes$ represent the Sigmoid function and element-wise multiplication, respectively. The *Feature Extractor* outputs the probabilities of positive and negative sentiments. We then extract these features from the BiLSTM outputs and combine them with typhoon environmental data as $\mathcal{D} \in \mathbb{R}^{n \times [m+c, v_-, v_+]}$, where n is the number of typhoon observations and m is the number of features. c, v_-, v_+ represent the tweets-based features: *tweets count, variance of negative sentiments and variance of positive sentiments*. The variance of sentiments is computed as follows:

$$
v = \frac{1}{c} \sum_{i=1}^{c} (\mathcal{S}_i - \mu)^2, \tag{4.4}
$$

where $\mathcal{S}_i$ denotes the predicted sentiment of tweet i, μ is the average of sentiments, and c is the tweets count.

4.3.4 Model II: Typhoon Classifier

Typhoon Classifer is the second model in our approach that takes the features extracted by the *Feature Extractor* as input and predicts the typhoon intensities as output. The architecture of this model consists of two convolution layers with RELU activation function. The first convolution layer has 32 filters with a kernel size of 3. The output of this layer is passed to the second convolutional layer, which has 16 filters with a kernel size of 3, which further refine the features. The details of the convolutional layers are explained in Section 2.4.1. After each convolution operation, a max-pooling layer is applied to reduce the output dimension. In addition, dropout rates of 0.3 and 0.2 are used after the first and second convolutional layers, respectively, to prevent overfitting and enhance model robustness. The final layer is a dense layer with a Softmax function, which computes the probabilities of each typhoon intensity category and returns the category with the highest probability as the final output, as shown in Equation (6.2)

4.4 Experiments

This section describes the experimental setup, the datasets, the baselines, and the evaluation results of our study. We conduct our experiments to answer the two research questions (Q_1, Q_2) that address Challenge I in section 1.4. Specifically, we investigate how social media data can enhance the performance of models that rely solely on environmental data for disaster prediction. Further, we examine how semantic embeddings of tweet representation affect the performance of our approach.

4.4.1 Baselines

We compared our approach with various baselines, including traditional machine learning and deep learning models. We chose the SVM classifier as a traditional baseline since it has superior performance and outperforms several machine learning models (Burel et al., 2017b). We also used four deep models (DNN, RNN, CNN, and BiLSTM) as neural baselines. Unlike Burel et al. (2017b), who proposed

enriching data representation by detecting named entities and creating a bag-of-concepts feature, we detect named entities and obtain their representations from the CONCEPTNET knowledge graph, which captures not only the existence of an entity but also its context. We summarize the variations of our approach as follows:

- **LSTM+DNN**$_{(word2vec)}$: This is our first approach that combines LSTM and DNN models in a joint-learning setting. LSTM serves as the *Feature Extractor* and DNN as the *Typhoon Classifier*. We use classical word embeddings (*Skip-gram*) to represent tweets.
- **LSTM+DNN**$_{(semantic-emb.)}$: his model is similar to LSTM+DNN$_{(word2vec)}$, but we enrich word embeddings with semantic vectors from CONCEPTNET.
- **LSTM+RNN**$_{(word2vec)}$: In this model, we use LSTM as the *Feature Extractor* and RNN as the *Typhoon Classifier*. Both models are trained jointly and represent tweets with *Skip-gram* word embeddings.
- **LSTM+RNN**$_{(semantic-emb.)}$: This model is the same as LSTM+RNN$_{(word2vec)}$, but incorporates semantic vectors from CONCEPTNET in addition to word embeddings.
- **BiLSTM+CNN**$_{(word2vec)}$: This model investigates a BiLSTM model as the *Feature Extractor* and CNN as the *Typhoon Classifier*. Both models are trained jointly with combined features from typhoons' environmental data and word embeddings.
- **BiLSTM+CNN**$_{(semantic-emb.)}$: This model is the same as BiLSTM+CNN$_{(word2vec)}$, but includes semantic vectors from CONCEPTNET in addition to word embeddings.

We adopted the same experimental setup as the baseline models for a fair comparison. Moreover, we investigated the effect of using semantic vectors derived from knowledge graphs to represent tweets versus the *Skip-gram* word embeddings.

4.4.2 Evaluation Setup

We used various metrics to evaluate the performance of our approach. These metrics include: *Accuracy* (Acc), *Precision* (Pre), *Recall* (Rec), and F_1 score, which are defined and explained in Section 2.6.1. We randomly split the dataset (described in Section 5.3.1) into 80% training and 20% testing sets. To avoid training overfitting, we applied an early stopping technique during the training phase of the model.

Table 4.3: Performance evaluation on the test dataset using accuracy (Acc), precision (Pre), recall (Rec) and F_1. Best results are in Bold

Model	Acc	Pre	Rec	F_1
SVM	0.579	0.347	0.579	0.430
$DNN_{(word2vec)}$	0.756	0.809	0.756	0.781
$RNN_{(word2vec)}$	0.802	0.827	0.802	0.814
$CNN_{(word2vec)}$	0.702	0.918	0.702	0.796
$BiLSTM_{(word2vec)}$	0.840	0.880	0.840	0.859
$LSTM+DNN_{(word2vec)}$	0.873	0.892	0.873	0.882
$LSTM+DNN_{(wemantic-emb)}$	**0.917**	0.922	**0.925**	**0.917**
$LSTM+RNN_{(word2vec)}$	0.860	0.875	0.860	0.855
$LSTM+RNN_{(semantic-emb)}$	0.891	0.904	0.891	0.891
$BiLSTM+CNN_{(word2vec)}$	0.847	**0.938**	0.847	0.890
$BiLSTM+CNN_{(semantic-web)}$	0.902	0.933	0.902	**0.917**

4.4.3 Discussion and Result Analysis

To answer Q_1, we compared the performance of different baselines (SVM, DNN, RNN, CNN and BiLSTM) in predicting typhoon categories based on features extracted from typhoons data. The results are presented in the top section of Table 4.3. We also propose three jointly-learning models (LSTM+DNN, LSTM+RNN, BiLSTM+CNN) that leverage both typhoon data and tweet features. The tweet features included the count c, the variance of positive v_+, and negative v_- sentiments of relevant tweets. We found that deep learning classifiers outperformed the SVM classifier on this task. Moreover, our models achieve significant improvements when incorporating tweet features with typhoon data. The highest accuracy was obtained by the BiLSTM+CNN model, which improved the *micro-average* F_1 score by 12.1% over CNN and by 3.1% over BiLSTM. The other models (LSTM+DNN and LSTM+RNN) also show substantial gains over their respective baselines (by 11% for DNN and by 5.8% for RNN) on average. These results indicate that social media can provide valuable features and enhance the performance of disaster prediction models. To further understand the contribution of each feature, we used the RandomForest classifier to evaluate the feature's importance. As illustrated in Figure 4.5, tweet features were more influential than environmental features, which were derived from sensor devices and had incomplete measurements (Morton and Levy, 2011).

To answer Q_2, We evaluated the impact of embedding representations from ConceptNet on the performance of our models. Our results show that ConceptNet embedding vectors for named entities enhance the performance across different

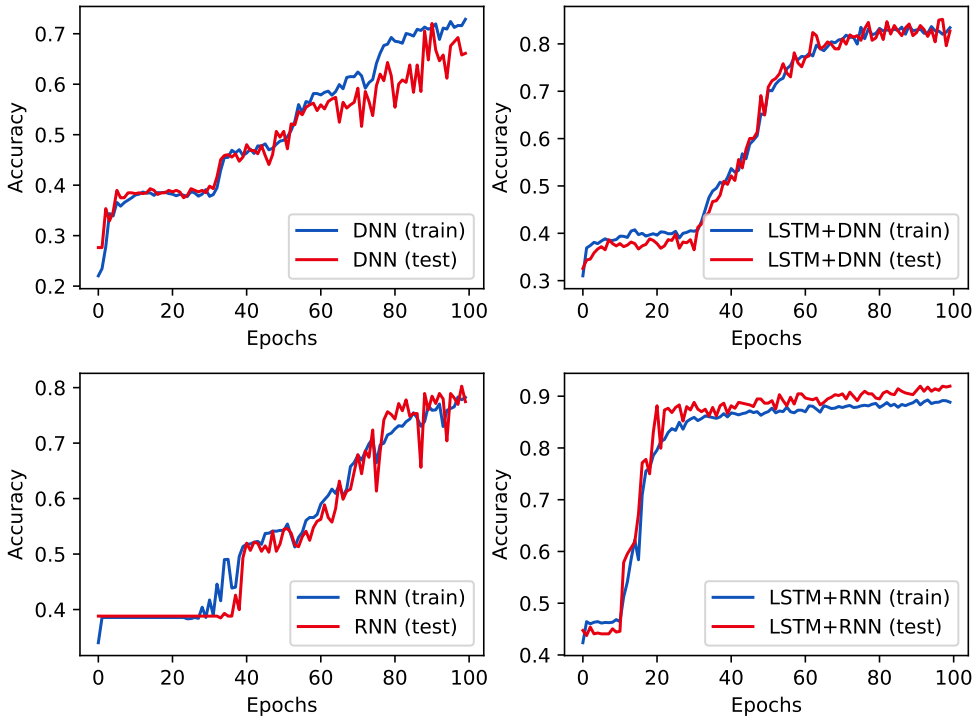


Figure 4.4: The evaluation of model's over-fitting

metrics, such as *Accuracy*, *Precision*, *Recall*, and F_1. Specifically, our approach obtains an accuracy gain of up to 3% in both LSTM+DNN and LSTM+RNN models compared to their baselines, which rely only on traditional word embeddings. These findings conclude that semantic embeddings from knowledge graphs can enhance the representation of entities and their relationships rather than traditional word embeddings.

Training Robustness. To ensure that our models are generalized well, we evaluated their robustness during the training phase. As depicted in Figure 4.4, all models were trained using an early stopping technique to ensure they did not overfit to the training data, and their robust performance in the testing set as well.

4.5 Summary and Conclusion

This chapter presented our approach to predicting typhoon intensities by jointly learning from social media and environmental data. We evaluated our approach on real-world datasets that included different typhoons and their associated tweets. Our approach differed from previous studies in that it extracted adaptive features

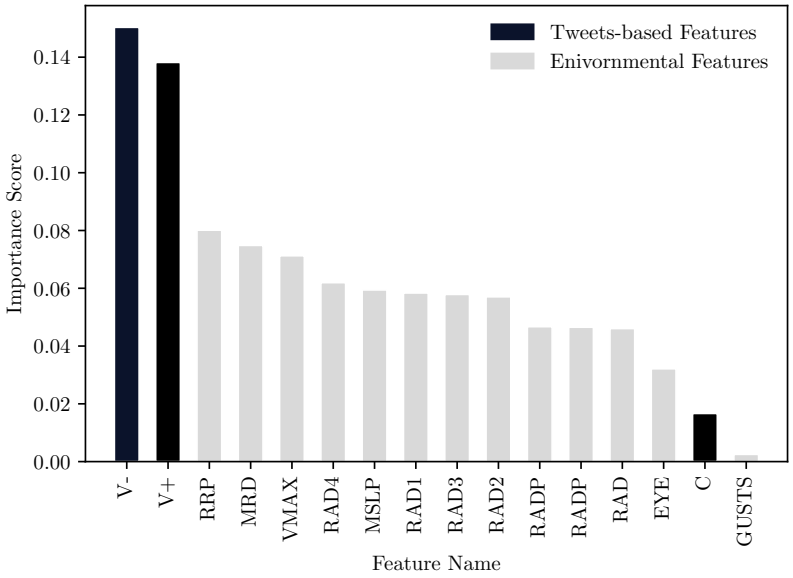


Figure 4.5: Importance of environmental and tweets-based features

by jointly learning from two models (e.g., BiLSTM+CNN). The BiLSTM model served as a *Feature Extractor* from social media, providing the CNN model with features derived from relevant tweets. Furthermore, we investigated the effect of using semantic embeddings to represent named entities on the performance of our approach. We detected named entities in tweets and obtained their embedding vectors from the ConceptNet graph. The evaluation results demonstrated that our approaches (LSTM+DNN: 87.3%, LSTM+RNN: 86.0% and BiLSTM+CNN: 0.90%) outperformed different baselines (DNN: 75.6%, RNN: 80.2%, CNN: 70% and BiLSTM: 84%). Remarkably, we observed that incorporating semantic vectors into our approach yielded improved F_1 scores (up to 3% in LSTM+DNN, up to 4% in LSTM+RNN and up to 2.7% in BiLSTM+CNN). In our future work, we plan to investigate methods for building knowledge graphs in disaster data. Our goal is to preserve semantic information and efficiently enable the integration of disaster data with their corresponding tweets.

5 Classifying Social Media into Multiple Information Types

This chapter addresses the research question $\mathbf{Q}_3$, (see Section 1.4), which explores how to classify social media data (tweets) into different information types simultaneously (i.e., multi-label classification). We present our approach UPB-BERT), which fine-tunes the BERT model for this task. The main content of this chapter is based on our publication work (Zahera et al., 2019a) where the author designed, implemented, and evaluated herein and co-wrote the aforementioned paper. Additionally, we provide a review of related works in Section 3.2.

5.1 Overview

In recent years, social media has emerged as a crucial information source during emergencies, enabling instant communication, and providing situational updates (Simon et al., 2015). As a result, several approaches (Houston et al., 2015; Landwehr and Carley, 2014) have been proposed to leverage social media data for mitigating disaster impacts and delivering faster relief responses. For instance, Sakaki et al. (2010) designed an earthquake-detection system that analyzes real-time tweets to locate affected regions and assist affected people. Furthermore, Stowe et al. (2016) developed a disaster information system for classifying social media posts (tweets) during and after disasters. Specifically, the authors investigated filtering tweets into fine-grained categories (e.g., *sentiment*, *reporting*, *action*) instead of filtering only relevant tweets. The evaluation results showed the effectiveness of the proposed approach in classifying disaster tweets, allowing emergency managers to access critical information and make faster disaster responses.

Most of the previous studies have focused on filtering informative social media data as either binary classification (i.e., relevant, or irrelevant) or a multi-class problem. For example, Caragea et al. (2016) proposed a CNN-based model to classify informative messages during disasters. The proposed approach demonstrated a significant improvement over traditional models, which use bag-of-words or n-grams features. Similarly, Burel et al. (2017a) developed an enhanced classification model

Table 5.1: The description of crisis information types (Olteanu et al., 2015)

Intent Type	Information Type	Description
REQUEST	GoodServices	Request for a particular service or physical good
	SearchAndRescue	The user is requesting a rescue for themselves or others
	InformationWanted	The user is requesting information
REPORT	Weather	Weather report
	FirstPartyObservation	The user is giving an eyewitness account
	ThirdPartyObservation	The user is reporting information from someone else
	EmergingThreats	Problems that cause loss or damage
	ServiceAvailable	Someone is providing a service
	SignificantEventChange	New occurrence to which officers need to respond
	MultimediaShare	Shared images or video
	Factoid	The user is reporting some facts, typically numerical
	Official	Report by a government or public representative
	CleanUp	Report of the cleanup after an event
	Hashtags	Report with hashtags correspond to each event
CALLTOACTION	Volunteer	Call for volunteers to help in response efforts
	Donations	Call for donations of goods or money
	MovePeople	Call to leave an area or go to another area
OTHER	PastNews	The post is reporting an event that has occurred
	ContinuingNews	The user is providing/linking to a continuous event
	Advice	Provide some advice to the public
	Sentiment	The post is expressing some sentiments about an event
	Discussion	Users are discussing an event
	Irrelevant	The post is irrelevant

with two CNN layers: i) a semantic layer that captures contextual information, and ii) a traditional CNN layer. The authors also incorporated semantic features (e.g., bag-of-entities) in tweet representation. The experimental results indicated superior performance when using semantic information within deep neural models compared to the traditional Word2vec embedding model. However, these approaches did not consider the possibility of classifying tweets into multiple information types (see Table 5.1) simultaneously. There is a lack of efficient tools that can categorize disaster tweets into more than one type. For example, a tweet may contain information about emerging threats and actionable information, such as moving people. In this case, a disaster model should assign both "*EmergingThreats*" and "*MovePeople*" types to the tweet. Olteanu et al. (2015) highlighted the importance of filtering disaster information into fine-grained types, which enable disaster organizations to quickly find relevant information. For example, humanitarian relief organizations may be particularly interested in tweets containing information about "*volunteers*" or "*donations*" information types, while local police might focus on information types such as "*MovePeople*". Toward this goal, TREC-IS[1] initiative was launched to help researchers to evaluate their systems in classifying real-world crisis tweets.

[1] http://dcs.gla.ac.uk/~richardm/TREC_IS/

Transfer learning from pre-trained models (especially BERT) has achieved impressive results in various NLP tasks without requiring training these models from scratch (Sun et al., 2019). Inspired by this success, we propose our approach (UPB-BERT) that fine-tunes the BERT model for multi-label classification of disaster tweets. In this context, we further train the BERT model on a dataset of tweets collected during different disaster events. By fine-tuning the BERT model, we achieve outperforming results in tweet classification using a small domain-specific dataset ($34k$ tweets) and reduce the computing time. We summarize the main contributions of our study as follows:

- To the best of our knowledge, this is the first that fine-tunes BERT for multi-label classification of disaster-related tweets.
- We conducted several experiments on real-world datasets provided by TREC-IS. Our evaluation showed that our fine-tuned BERT model can effectively classify tweets into multiple information types when fine-tuned on a crisis dataset.

5.2 Our Approach

This section presents our approach to categorizing disaster-related tweets into multiple information types. We first describe the preprocessing steps required for cleaning redundant data and noises from tweets. Then, we explain the fine-tuning of the BERT model for multi-label tweet classification.

5.2.1 Tweets Preprocessing

Processing tweets poses unique challenges compared to longer documents. Due to their nature, tweets are inherently noisy, full of informal abbreviations and emojis. As discussed in Section 2.2, specialized preprocessing is essential for generating better features from tweets. To accomplish this, we used the *tweettokenize* API[2], which offers ad-hoc preprocessing and creates a uniform representation of tweets. Figure 5.1 shows an example of preprocessing a tweet about hurricane SANDY into a vector of words. We applied the following steps for tweet preprocessing:

- We removed URLs, usernames, and Unicode characters from tweets. Additionally, we eliminated all extra white spaces, duplicated full stops, question marks, and punctuation points.

[2] https://www.nltk.org/api/nltk.tokenize.html

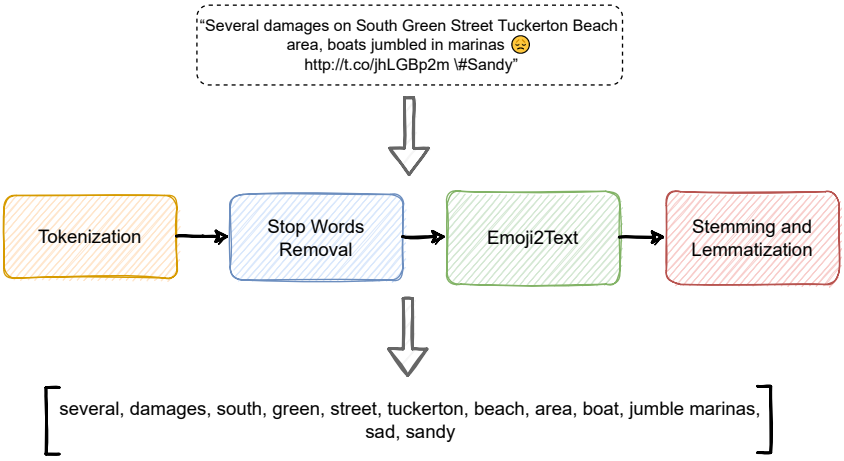


Figure 5.1: An example tweet shared during hurricane SANDY

- We retained stop words to provide sufficient contextual information for the BERT model. For example, such negative words (e.g., not, nor, and never) provide critical information on the meaning of the subsequent words (e.g., "not happy" implies "sad").
- For emojis symbols, we employed the *emoji*[3] library to convert them into text. For example, the emoji " : (" is converted into the word "`sad`".
- We applied stemming and lemmatization techniques to normalize words and restore general forms using *WordNetLemmatizer*[4].
- We converted all tweet tokens to lowercase letters.

5.2.2 Fine-tuning BERT Model

BERT, or Bidirectional Encoder Representation from Transformers, is a groundbreaking model proposed by (Devlin et al., 2019), which can be easily adapted to various downstream NLP tasks through fine-tuning. Since its release in 2018, BERT has become the flagship of pre-trained language models, demonstrating successful applications across numerous tasks, such as text classification (Munikar et al., 2019), natural language understanding (Jawahar et al., 2019), and question answering (Yang et al., 2019)). BERT enables faster development, reduces data requirements, and achieves superior results. Motivated by the success of fine-tuning BERT (Chang et al., 2020), we developed our own fined-tuned BERT model (named

[3] `https://pypi.org/project/emoji/`
[4] `https://www.nltk.org/_modules/nltk/stem/wordnet.html`

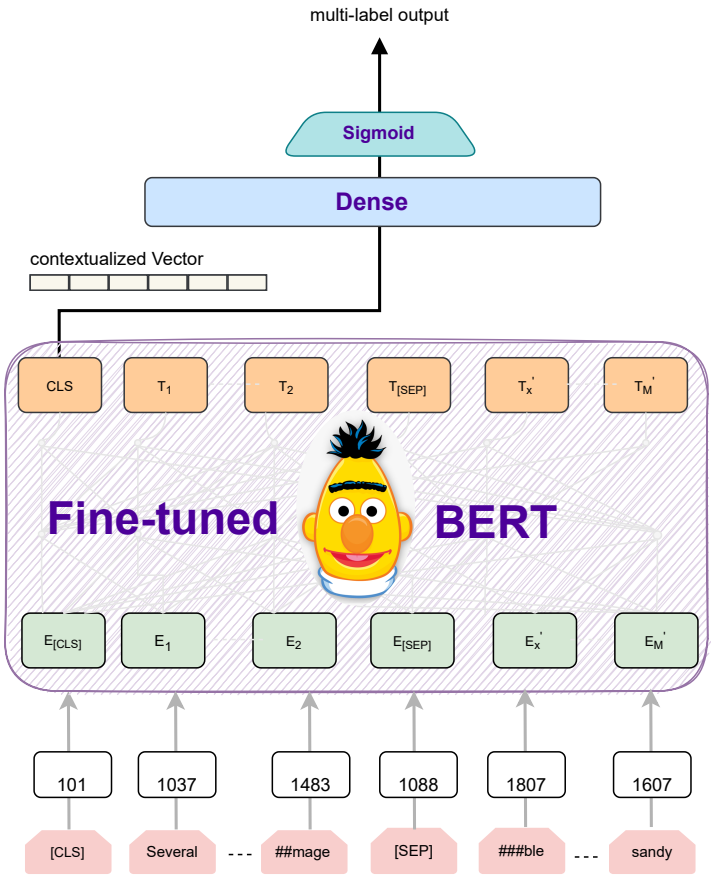


Figure 5.2: Our fine-tuned BERT model for multi-label tweets classification

UPB-BERT) to classify crisis-related tweets into multiple information types, as shown in Figure 5.2. The pre-trained contextualized embeddings from BERT have proved to be rich representations of words and sentences, eliminating the need for handcrafted features. As discussed in Section 2.3.2, BERT input should be transformed into a specific format. A $[CLS]$ token is inserted at the beginning of each sentence, and a $[SEP]$ token is used to specify the sentence end. The input is then tokenized by the *WordPiece* tokenizer (Song et al., 2021). Each token t_i is associated with three types of embeddings: *token embeddings* (E_{t_i}), which represents the vocabulary index of each token; *segmentation embeddings*, which distinguish between input sentences (E_A or E_B); and *position embeddings* (E_i), which indicate the position of each word.

In our study, we used the BERT$_{base}$ model, which consists of 12 transformer blocks, 12 self-attention heads, and a hidden size of 768. The maximum length input is truncated to no more than 512 tokens as required for the BERT input. The embedding vector of the input is obtained from the hidden state of the $[CLS]$ token

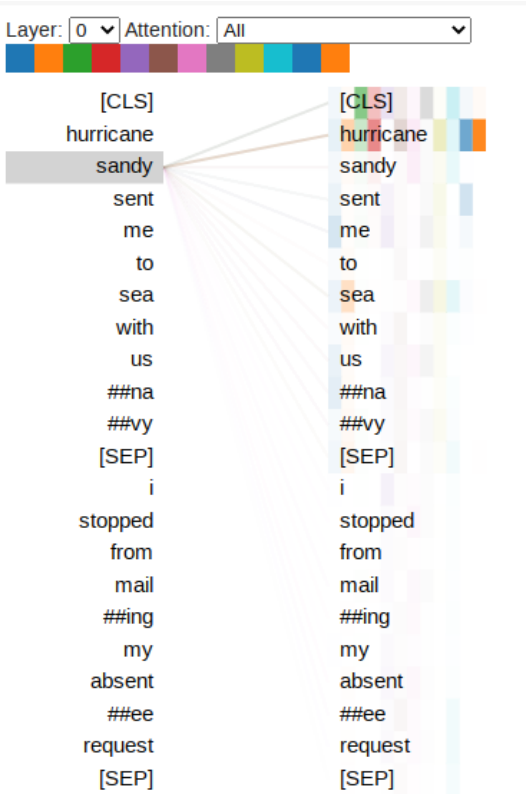


Figure 5.3: An example of attention visualization of BERT model

Table 5.2: Overview of the TREC-IS dataset

# Statistic	Train	Test
Total No. of events	15	6
Total No. of tweets	34,290	8,573
No. of Information Types	24	25

(i.e., the summary token). To predict the information types of an input tweet, we added an additional Dense layer (i.e., Fully connected) with a Sigmoid activation function. Furthermore, we minimized the classification errors during the training using a binary Cross-entropy loss function (see Equation (4.2)). By applying a threshold value (> 0.5), we filtered the most relevant information types to the input tweet.

$$\hat{\mathcal{Y}} = \sigma(\mathcal{W}_i \times x_i + b_i), \tag{5.1}$$

where x_i is the embedding vector of tweet i and $\mathcal{W}_i$ is the weight matrix of BERT that has been tuned. Additionally, σ denotes the *Sigmoid* function, and b represents the bias.

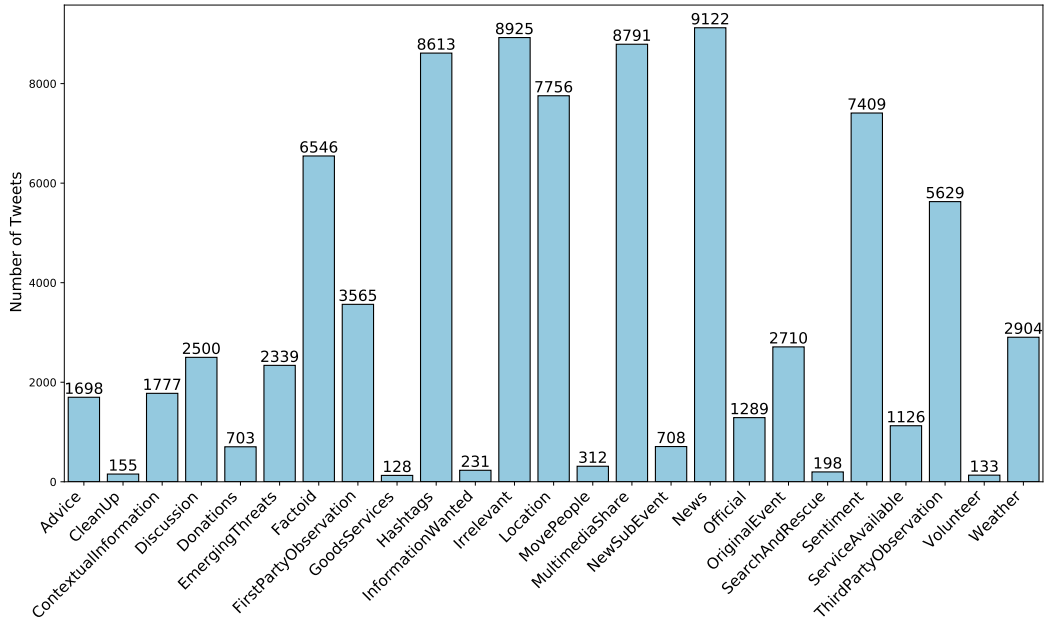


Figure 5.4: Information types per tweets in the TREC-IS dataset

5.3 Experiments

In this section, we describe the datasets, baselines, and evaluation metrics employed in our experiments to answer the research question (Q_3), which focuses on evaluating the performance of fine-tuning BERT model in multi-label tweets classification.

5.3.1 Dataset

We conducted our experiment on the TREC-IS dataset (Mccreadie, 2019), which was curated from various disaster events, such as earthquakes, hurricanes, and public shootings. The dataset comprises $34,2k$ tweets for training, $8,5k$ for testing, and an overall total of $42,7k$ tweets. A statistical overview of the train and test sets used in this study is presented in Table 7.3. For each event, tweets were collected using event-related hashtags and keywords through the Twitter search API. Furthermore, human annotators were recruited to label tweets according to a multi-layer ontology of information types, as shown in Table 5.1. According to this annotation, a tweet could be classified into one or more information types (i.e., classes). Figure 5.4 shows the distribution of tweets across different information categories.

We observed that most tweets are labelled with the "*Sentiment*" class (up to $7k$ tweets), while few tweets have labels such as "*MovePeople*". Our analysis also

Table 5.3: The evaluation results of our two variants of fine-tuned BERT (UPB-BERT and UPB-FOCAL) under metrics: AAW, F_1, Acc, and RMSE

System	$\mathbf{AWW}_{(high)}$	$\mathbf{AWW}_{(all)}$	$\mathbf{F}_{1(act)}$	$\mathbf{F}_{1(all)}$	Acc	$\mathbf{RMSE}_{(act)}$	$\mathbf{RMSE}_{(all)}$
UPB-BERT	-0.95	-0.47	**0.13**	0.14	0.81	0.15	0.09
UPB-FOCAL	-0.93	-0.47	0.12	**0.18**	0.81	**0.14**	**0.08**
Median	**-0.91**	**-0.46**	0.03	0.10	**0.85**	0.17	0.10

showed highly imbalanced distributions of the information types in the TREC-IS dataset. To address this issue, we used *macro-average* scores of evaluation metrics: *Precision*, *Recall*, and F_1. Additionally, we assigned weights to classes relative to the number of tweets they contained. Specifically, classes with a large number of instances were assigned lesser weights, while classes with few instances were given greater weights. In our future work, we plan to use recent language models, such as GPT-4, to generate a large number of tweets, thereby achieving a better balance within the dataset.

5.3.2 Baselines

We implemented two versions of our approach:

- UPB-BERT: This model uses a pre-trained BERT model which was fine-tuned for the task using a standard cross-entropy loss function.
- UPB-FOCAL: This model has the same architecture as UPB-BERT but employs the focal loss function during training to address the class imbalance problem (Mulyanto et al., 2020).

The two models were evaluated against different approaches that participated in the TREC-IS Challenge 2019.[5] The challenge, organized by the University of Glasgow, aimed to promote research in information retrieval technologies for emergency response situations. The participant approaches employed different techniques, ranging from traditional machine learning approaches (e.g., Support Vector Machine, RandomForest) to deep neural approaches (e.g., LSTM, CNN). A detailed description of the participant methods can be found in the TREC-IS evaluation report (McCreadie et al., 1970).

Our models, UPB-BERT and UPB-FOCAL, were evaluated against the baseline methods, and their performance is presented in Table 5.4. The table shows the full

[5] http://dcs.gla.ac.uk/~richardm/TREC_IS/2020/task.html

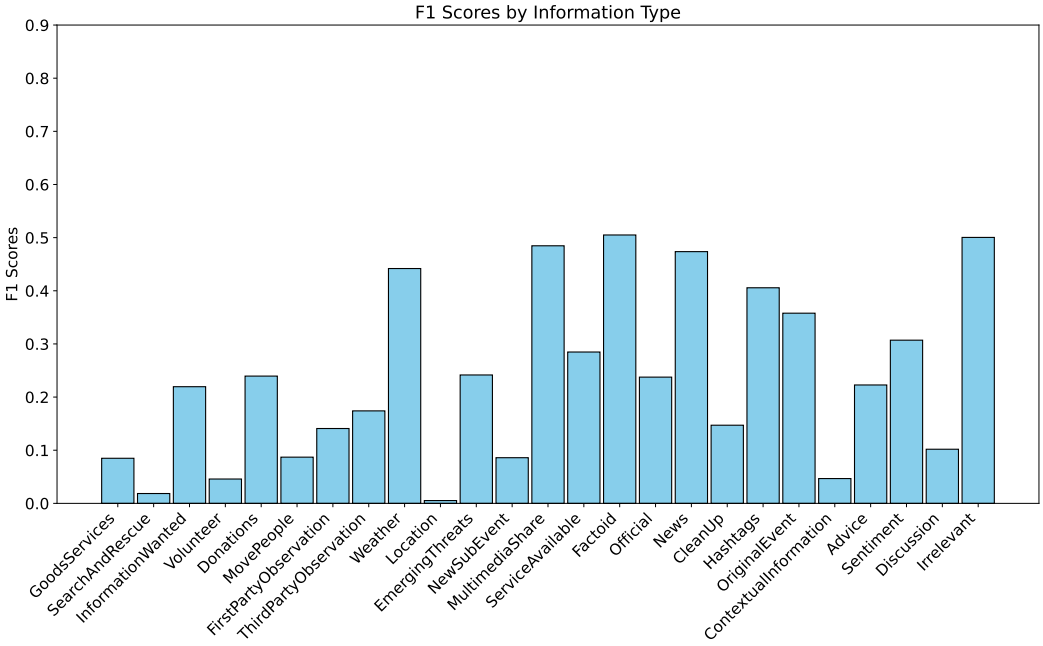


Figure 5.5: F_1 scores of the UPB-BERT across all information types

evaluation results from the TREC-IS challenge 2019 (run B), providing a comprehensive comparison of the various approaches.

5.3.3 Evaluation Metrics

We obtained the evaluation results from the TREC-IS evaluation system, in terms of *Accuracy* (Acc), F_1, and *Accumulated Alert Worth* (AWW), as detailed in Section 2.6.1. Additionally, the evaluation system computed *Root Squared Mean Error* (RMSE) to estimate the prediction errors of high-priority tweets compared to human-generated scores. This evaluation aspect aims to evaluate the ability of classification models in generating timely and accurate alerts for highly critical tweets.

5.3.4 Results and Discussion

Table 5.3 shows the results of the two variants of our approach (UPB-BERT and UPB-FOCAL), in comparison with the median scores of all participant approaches in the TREC-IS evaluation in 2019. The evaluations are reported in two divisions: *high* priority information types and *all* types. Apparently, our approach (UPB-FOCAL) demonstrates enhanced performance under the F_1 metric with an improvement of

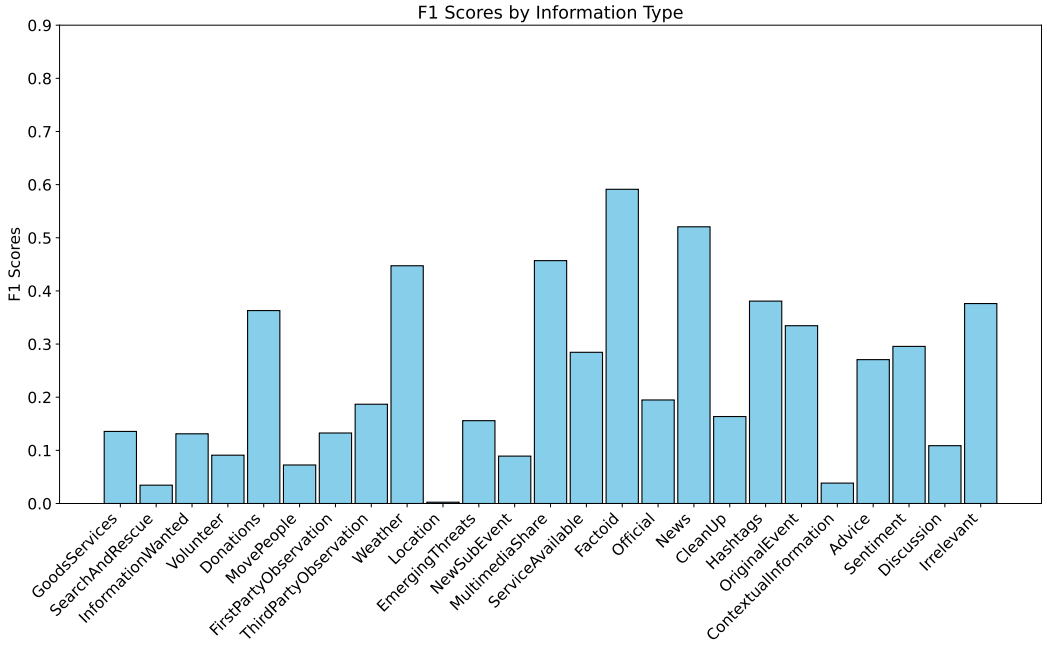


Figure 5.6: F_1 scores of the UPB-FOCAL across all information types

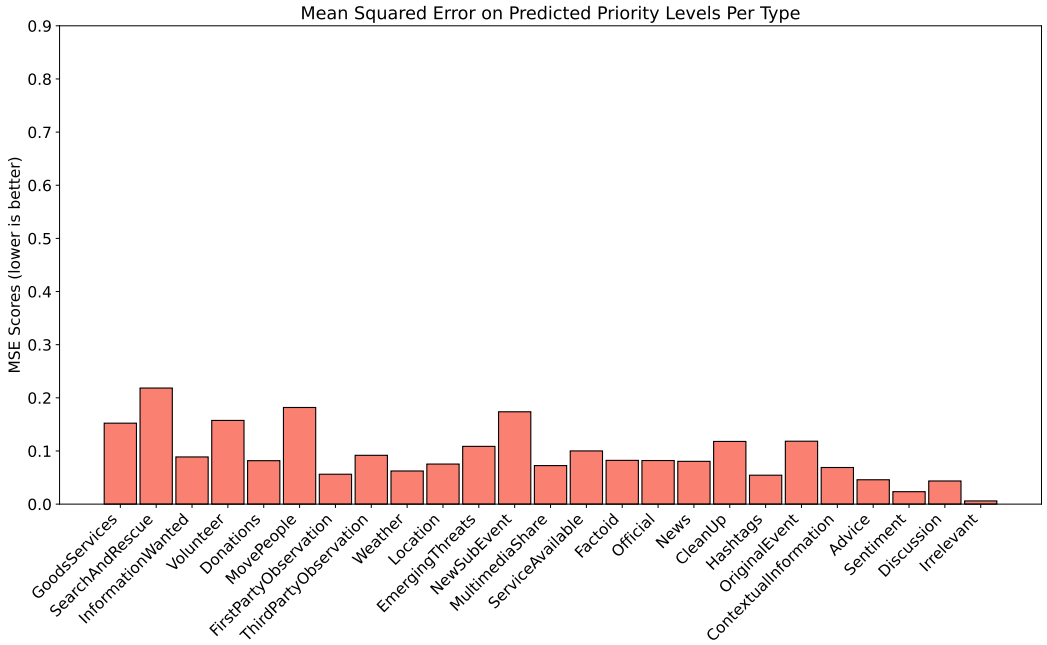


Figure 5.7: RMSE scores of the UPB-BERT across all information types

Table 5.4: The results of participant systems in the TREC-IS 2019 challenge (McCreadie et al., 1970). The best results are in bold

Systems	Information Feed			Prioritization	
	Info. Type Positive		Info. Type Accuracy	Prioritization RMSE	
	Actionable	All	All	Actionable	All
CS-UCD$_{basline}$	**0.1355**	0.223	0.7495	0.0859	0.0859
UPB-BERT	01338	**0.2343**	0.8139	0.1558	0.0938
CMUInformedia	0.1321	0.2167	0.8605	**0.0788**	0.0544
BERT-FOCAL	0.1287	**0.2343**	0.8159	0.1416	0.0829
CS-UCD$_{bilstmbeta}$	0.1269	0.1676	0.8378	0.1004	0.0822
CS-UCS$_{bcelmo}$	0.1099	0.1721	0.8452	0.1036	0.0769
DLR$_{BERT-R}$	0.0998	0.1989	0.0856	0.1834	0.1019
NYU$_{fast.multi}$	0.0854	0.2256	**0.8808**	0.2153	0.1185
CMUInformedia$_{rf2}$	0.0642	0.1382	0.8624	0.1025	0.0683
CS-UCD$_{bilstmalpha}$	0.0614	0.171	0.8600	0.1521	0.0893
NYU$_{base.sing}$	0.0606	0.1373	0.8658	0.1836	0.1104
CMUInformedia$_{rf3}$	0.0592	0.0813	0.8434	0.1660	0.2063
NYU$_{fast.sing}$	00431	0.1228	0.8739	0.2085	0.1169
UAGPLSI$_{baseline}$	0.0386	0.0302	0.8753	0.2067	0.1169
UAGPLSI$_{irn}$	0.0386	0.0302	0.8753	0.2138	0.1175
UAGPLSI$_{negative}$	0.0377	0.0278	0.8758	0.2075	0.1154
UAGPLSI$_{all}$	0.0377	0.0278	0.8758	0.2138	0.1177
ICTNET$_{dl}$	0.0347	0.0871	0.7285	0.1254	0.1451
CMUInformedia$_{rf1}$	0.0300	0.1361	0.8638	0.0815	0.0551
IITBHU$_{run2}$	0.0275	0.0548	0.7892	NA	NA
DLR$_{Fusion}$	0.0249	0.0939	0.8689	0.1916	0.1077
IRIT$_{run2}$	0.0248	0.1725	0.8534	0.1175	0.0659
IITBHU$_{run1}$	0.0191	0.0893	0.8139	0.1879	0.1128
DLR$_{SIF-R}$	0.016	0.1004	0.8605	0.2093	0.1129
IRIT$_{run1}$	0.0151	0.1677	0.8418	0.1316	0.0911
DLR$_{Mean}$	0.0071	0.0922	0.8635	0.2111	0.1153
IRIT$_{run4}$	0	0.1317	0.7576	0.1461	0.0775
IRIT$_{run3}$	0	0.131	0.8565	0.1771	**0.01028**
CBNU$_{C1}$	0	0	0.8788	NA	NA
IRIT$_{S1}$	0	0	0.8788	NA	NA

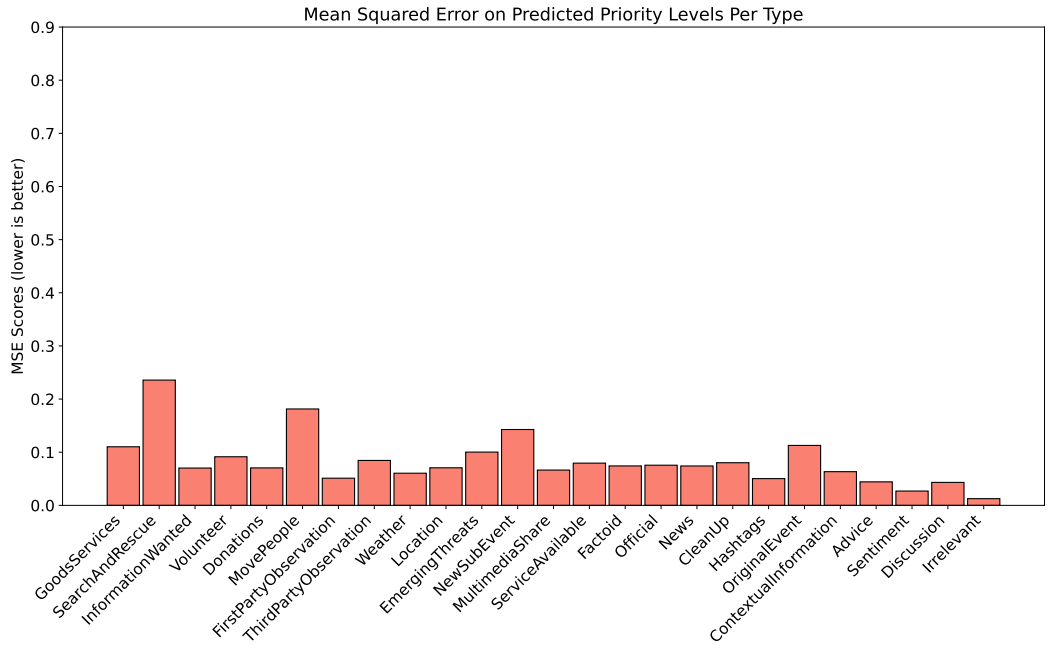


Figure 5.8: RMSE scores of the UPB-FOCAL across all information types

8% and a reduced error of 3%. However, the AAW metric indicates that our approach does not perform effectively in detecting tweets containing actionable information. To address this gap, we developed an improved architecture with an additional graph neural component, which is described in the next Chapter.

We analyzed the evaluation results (in terms of F_1 and RMSE) in Figure 5.5, Figure 5.6, Figure 5.7, and Figure 5.8 of our approaches (UPB-BERT and UPB-FOCAL) for all information types. The results show that we obtained higher precision and recall for categories (*News, Sentiment, MultimediaShare, Factoid*) that were more represented in the training dataset. In contrast, our approaches were unable to generalize well across information types with fewer tweets, such as *GoodServices, SearchAndRescue, CleanUp,* and *Volunteer*.

Limitation of our approach: Despite the significant results that our approach achieved in classifying tweets into multiple types, we observed that our models (UPB-BERT and BERT-FOCAL) were not efficient in detecting actionable information compared to other methods in Table 5.4. We attributed this sub-optimal performance to our approach, which depends on the contextualized representation of the BERT model. This representation ignores global relationships between words across the tweet corpus and only considers local information within sentences. Therefore, we propose an improved approach called I-AID in Chapter 6 to address these challenges.

5.4 Summary and Conclusion

This chapter described our studies for fine-tuning of BERT language model in classifying disaster-related tweets. Specifically, we proposed two fined-tuned BERT models: the first model (UPB-BERT) minimizes training errors using a binary Cross-entropy loss function, while the second model (UPB-FOCAL) employs the Focal loss function to handle imbalances in the TREC-IS dataset. Additionally, our approach leveraged contextualized embeddings from a pre-trained BERT model to represent tweets. The experimental results showed that the BERT model can effectively classify tweets into multiple labels with appropriate fine-tuning. However, the BERT model showed insufficient performance in detecting tweets with high-priority information. In the next Chapter, we describe our study to categorize disaster-related tweets with more focus on detecting actionable information.

6 Identifying Actionable Information From Social Media

This chapter addresses the research questions $\mathbf{Q}_4$ and $\mathbf{Q}_5$ (see Section 1.4) for identifying actionable information from social media. The main content of this chapter is based on our publication work (Zahera et al., 2021), where the author developed, evaluated, and co-wrote the said paper. Furthermore, we provide a review of related work in Section 3.2.

6.1 Overview

Social media has become an essential medium for disseminating information in emergency situations (Zade et al., 2018). A significant difference between social media and traditional news sources is the real-time feedback from the affected people. This two-way communication channel can assist disaster relief organizations in both informing the public and gaining insight into disaster situations. Consequently, extracting crisis information from social media posts (e.g., tweets) can enhance situational awareness and accelerate relief responses. Previous works (Stowe et al., 2018; To et al., 2017) primarily addressed information extraction from social media as a binary classification task, i.e., filtering tweets into *Relevant* or *Irrelevant* categories. However, there is a lack of efficient systems that can categorize relevant posts into fine-grained labels, as defined in (McCreadie et al., 2019) (see Figure 6.1). Fine-grained labels are invaluable for crisis responders, since they facilitate the filtering of critical data by more informative types, enabling faster disaster responses. In particular, labelling disaster-related tweets with multiple labels allows the rapid detection of tweets containing *actionable* information. Table 5.1 shows the list of information types (used as labels) defined by McCreadie et al. (2019). We adopt the definition of *actionable* tweets as formalized in (Zade et al., 2018). *Actionable* tweets are those that require immediate attention from emergency managers who are seeking critical information such as *SearchAndRescue*, and *MovePeople*), in con-

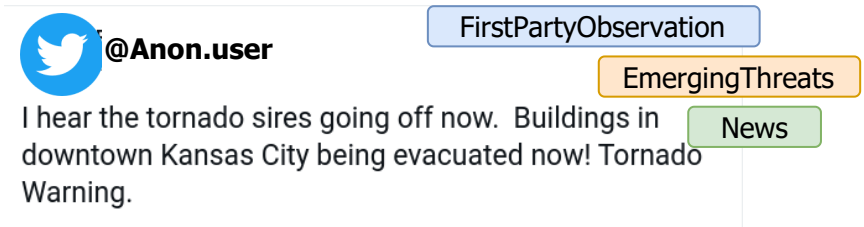


Figure 6.1: An example of a multi-label tweet classification

trast to *non-actionable* tweets that are labelled with categories such as *Hashtags* or *FirstPartyObservation* (refer to Table 5.1).

Tweet classification is a well-known challenging NLP task (Song et al., 2014), since tweets often lack contextual information, are inherently noisy (i.e., contain misspellings, acronyms, emojis, etc.), and have insufficient contextual information. Moreover, multi-label classification becomes even more challenging, as a tweet can simultaneously belong to multiple labels. In our study, we aim to i) categorize disaster-related tweets with fine-grained information types and ii) identify *actionable* or *critical* tweets that may be relevant for disaster relief and support disaster mitigation. Our approach consists of three components: i) BERT-ENCODER, which utilizes BERT as a sentence encoder to capture the semantics of tweets and represent them as contextualized embedding vectors, ii) TEXTGAT, which employs a graph attention network (GAT) to capture correlations between words and entities in tweets and the labels of these tweets, and iii) RELATION NETWORK (Sung et al., 2018), which is used as a learnable distance metric to compute the similarity between tweets vectors (obtained from the BERT-ENCODER) and labels vectors (obtained from the TEXTGAT) in a supervised way. This integration allows us to incorporate a contextualized representation of tweets from BERT and structural information between tweets and their labels from the TEXTGAT. Our main contributions can be summarized as follows:

- We propose a multi-model approach (named I-AID) to categorize disaster-related tweets into multiple information types.
- Our approach leverages a contextualized representation from the pre-trained BERT model to capture the tweet's semantics. Additionally, our approach employs a GAT component to capture the structural information between words and entities in tweets and their labels.
- We employ a *learnable* distance metric, in a supervised way, to determine the similarity between a tweet vector and label vectors.

Table 6.1: A list of symbols used in this chapter

Symbol	Description
S	Number of tweets in the dataset.
w	Tweet tokens (e.g., word or entity).
$y^{(i)}$	Ground-truth multi-label assigned to a tweet i.
$\hat{y}^{(i)}$	Predicted multi-label assigned to a tweet i.
λ_i	A single label/information type for a tweet.
N	Number of nodes in a graph
$\mathcal{V}$	Nodes of a graph
$\mathcal{E}$	Edges between nodes in a graph
A	Adjacency matrix of a graph
$\tau^{(i)}$	Embedding vector of tweet i learned by BERT
$h^{(i)}$	Embedding vector of node $v^{(i)}$
F	Dimension of node vector
$\iota^{(i)}$	Embedding vector for label $\lambda^{(i)}$
Z	The concatenated vector of $\tau^{(i)}$ and ι
$\mathcal{L}$	Binary cross-entropy loss function
α_{ij}	Attention score between nodes $v^{(i)}$ and $v^{(j)}$
$hPW(t)$	Scoring function for high-priority tweets.
$hPW(l)$	Scoring function for low-priority tweets.

- We conducted various experiments to evaluate the performance of our approach and the state-of-the-art baselines in multi-label text classification. The implementation of our approach and the datasets are available at the Github repository.[1]

6.2 Our Approach

In this section, we begin by providing a formal definition of multi-label tweet classification. Subsequently, the details of each component in our approach are discussed in Section 6.2.2. Figure 6.2 gives an overview of our approach and how its components work together.

6.2.1 Problem Formulation

Let $\mathcal{T}$ denotes a set of tweets and $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ be a set of k predefined labels, also referred to as information types. We formulate the problem of identifying

[1] https://github.com/dice-group/I-AID

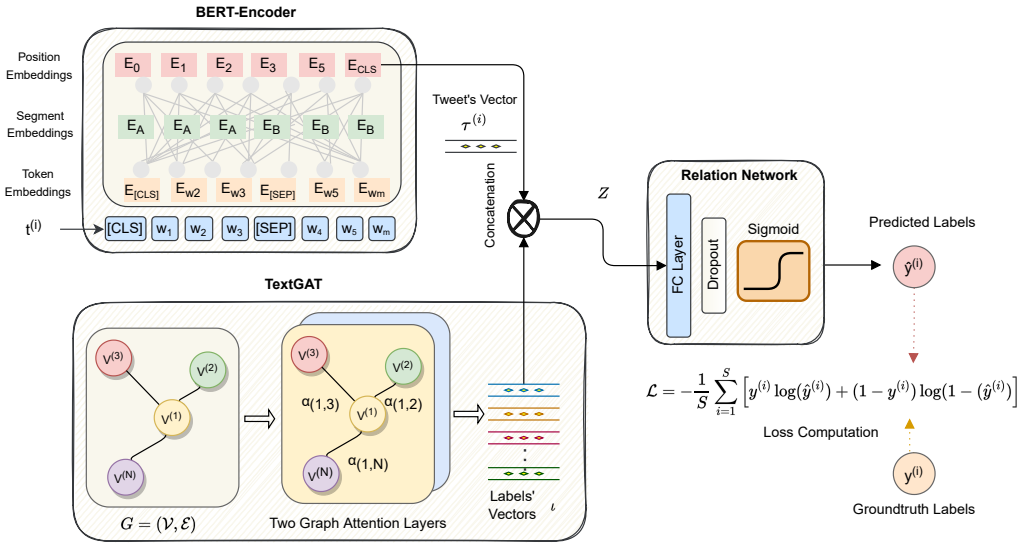


Figure 6.2: The I-AID architecture: BERT-ENCODER embeds a tweet $t^{(i)}$ into a feature vector $\tau^{(i)}$. TEXTGAT builds a graph G from our dataset and employs graph attention layers and output labels vectors ι. RELATION NETWORK learns a distance metric between $\tau^{(i)}$ and ι, then predicted labels $\hat{y}^{(i)}$ for $t^{(i)}$.

crisis information from tweets as a multi-label classification task, where a tweet t can be assigned one or more labels from Λ simultaneously. Our objective is to build a multi-label model $\mathbf{M} : \mathcal{T} \rightarrow (0,1)^k$ that maps tweets $\mathcal{T}$ to their relevant labels from Λ. Given a labelled dataset $\mathcal{D} = \{(t^{(i)}, y^{(i)}) \times \{0,1\}_j^k\}_{i=1}^S$ consisting of S tweets, each tweet $t^{(i)}$ is labelled with a set of relevant classes $y^{(i)}$. Here, $y_j^{(i)} = 1$ indicates that $t^{(i)}$ belongs to the class λ_j, while $y_j^{(i)} = 0$ means that $t^{(i)}$ does not belong to the class λ_j. Our approach aims to learn the function $\mathbf{M}$ using *three neural components*. First, we obtained a contextualized embedding vector ($\tau^{(i)}$) to represent tweet $t^{(i)}$ using a pre-trained BERT model. Concurrently, our approach acquires the embeddings vectors of labels ι through a graph attention network. Both vectors ($\tau^{(i)} \otimes \iota$) are then concatenated and fed to the last component (RELATION NETWORK) to match the most relevant labels to the input tweet.

6.2.2 The I-AID Architecture

Component I (BERT-ENCODER): This is the first component in our framework that transforms an input tweet into a vector representation τ that captures its contextual meaning. As illustrated in Figure 6.2, the BERT-ENCODER takes a tweet $t^{(i)}$ with m tokens $[w_1^{(i)}, w_2^{(i)}, \ldots, w_m^{(i)}]$ and generates its embedding vector $\tau^{(i)}$. We employ the BERT-base architecture comprising 12 encoder blocks, 768 hidden dimensions, and 12 attention heads. We refer readers to the original BERT paper (Devlin et al., 2019)

for a detailed description of its architecture and input representation. As discussed earlier in Section 2.3.2, a special preprocessing is performed for BERT input. Specifically, a `[CLS]` token is appended to the beginning of a tweet, and another token `[SEP]` is inserted after each sentence as an indicator for sentence boundaries. Each token $w^{(i)}$ is assigned three types of embeddings (*token*, *segmentation*, and *position*). These three embeddings are combined into a single output vector $\tau^{(i)}$ that captures the meaning of the input tweet.

Component II (TEXTGAT): Traditional methods (e.g., Word2vec (Mikolov et al., 2013a)) are capable of adequately representing words as embedding vectors based on the local context of a target word (within a window size of n words). However, these methods ignore the structural information and relationships between words in a text corpus (Peng et al., 2018). Recently, graph neural networks (Yao et al., 2019) have successfully addressed this challenge by modelling text as a graph, where words are considered as nodes and relations between them are edges. In our study, we construct a graph $G = (\mathcal{V}, \mathcal{E})$ from the dataset $\mathcal{D}$ with $\mathcal{V}$ and $\mathcal{E}$ representing sets of nodes and their edges, respectively. Each node $v^{(i)} \in \mathcal{V}$ can be a *word, named entity*[2] or *tweet label* (see Table 5.1). Nodes are represented using a feature matrix $\mathbf{H} = \{h^{(1)}, h^{(2)}, \cdots, h^{(N)}\}$ where $h^{(i)} \in \mathbb{R}^F$ is the feature vector of a node $v^{(i)}$ with F dimension, and N denotes the number of nodes. First, we initialize the nodes' representation $\mathbf{H}$ with pre-trained embeddings from the GloVe model (Socher and Manning, 2014). Additionally, relations between nodes are modelled using an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. As depicted in Figure 6.2, the TEXTGAT component consists of two graph attention layers. Each layer takes nodes' features $\mathbf{H}$ as input and performs an *attention* operation (Velickovic et al., 2018) to learn a new feature $\hat{\mathbf{H}} = \{\hat{h}^{(1)}, \hat{h}^{(2)}, \cdots, \hat{h}^{(N)}\}$ for each node based on the importance of its neighbours (i.e., *attention from its neighbours*). Consequently, we employ the shared attention mechanism $\mathbf{att} : \mathbb{R}^{\hat{F}} \times \mathbb{R}^{\hat{F}} \longrightarrow \mathbb{R}$ over all nodes. The graph attention operated on the node representation can be expressed as:

$$\alpha_{ij} = att\ (\mathbf{W}v^{(i)}, \mathbf{W}v^{(j)}), \tag{6.1}$$

where att is a single-layer feedforward network, parameterized by a weight matrix $\mathbf{W} \in \mathbb{R}^{\hat{F} \times F}$, which is applied to every node. Finally, a Softmax function is used to normalize the attention scores, as shown in Equation (6.2).

$$\alpha_{ij} = \frac{exp(\alpha_{ij})}{\sum_{k \in N_i} exp(\alpha_{ik})}. \tag{6.2}$$

[2]We detected named entities in tweets using SPACY entity recognizer https://spacy.io/api/entityrecognizer

To this end, TEXTGAT learns the structural information between nodes based on the importance of neighbour nodes. The label vectors are then concatenated with the tweet vector and fed to the next component, as shown in Figure 6.2.

Component III (RELATION NETWORK): This component aims to learn the similarity between the tweet's vector $\tau^{(i)}$ and label vectors ι in a supervised manner (also known as learning-to-learn or meta-learning). We employ a neural network as a learnable (i.e., non-linear distance) function to identify patterns of similarity. The RELATION NETWORK takes as inputs the concatenated matrix $Z = \tau^{(i)} \otimes \iota$ from the BERT-ENCODER and the label vectors. Since our task is a multi-label classification, we use a binary Cross-entropy as a loss function in Equation (6.3). Afterwards, we compute the probability of each label independently over all possible labels using a Sigmoid function in the output layer. Finally, a set of relevant labels is returned as the final result.

$$\mathcal{L} = -\frac{1}{S}\sum_{i=1}^{S}\left[y^{(i)}\log(\hat{y}^{(i)}) + (1-y^{(i)})\log(1-(\hat{y}^{(i)})\right], \tag{6.3}$$

where $y^{(i)}$ and $\hat{y}^{(i)}$ are the predicted and ground-truth labels of a tweet i, respectively. S is the size of tweets in the training dataset.

6.3 Experiments

This section presents the experimental setup, the datasets, the baselines, and the evaluation metrics that we employed in our experiments. We aim to address the research questions Q_4, and Q_5 (see Section 1.4), which focus on i) assessing the performance of our approach to classifying disaster-related tweets into multiple information types, ii) evaluating the performance of our approach in identifying actionable information, and iii) finally benchmark the impact of each component in our approach on overall performance (i.e., an ablation study).

6.3.1 Datasets

We used two benchmark datasets, which contain crisis-related tweets collected by TREC (McCreadie et al., 2019). Table 6.2 provides an overview of each dataset, including the number of tweets in the training set (**#Train**), validating (**#Valid**), and testing (**#Test**), as well as the total number of classes (**#Classes**). We split

Table 6.2: Overview of the Datasets.

Datasets	#Train	#Valid	#Test	#Classes
TREC-IS	27,467	6,867	8,584	25
COVID-19 Tweets	4,844	1,211	1,514	12

each dataset with an $80\% - 20\%$ ratio, where we used 80% of the tweets for training and 20% for testing. During the training phase, we used 20% of the training data to validate the model. We briefly summarize each dataset as follows:

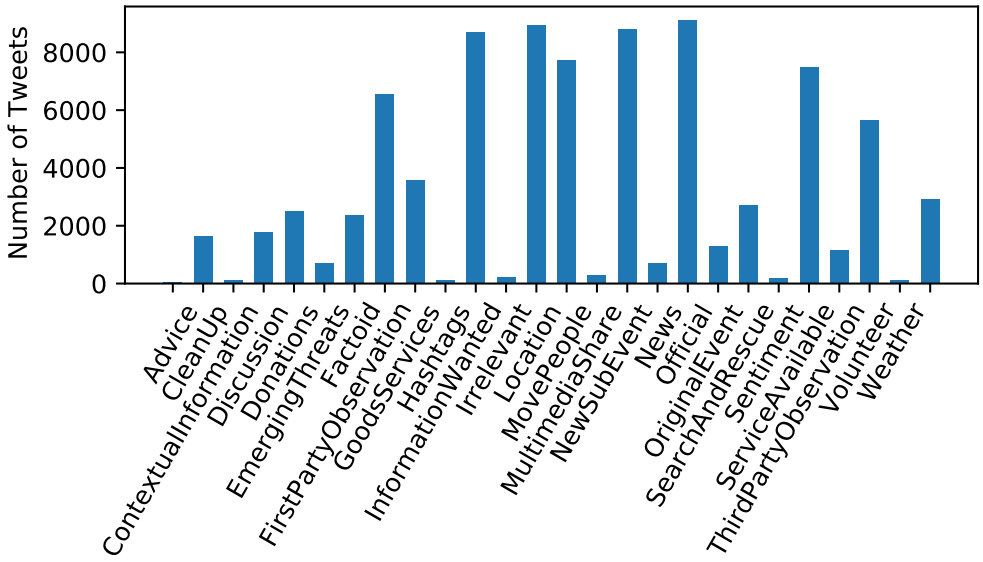
- **TREC-IS:** This dataset contains approximately 35K tweets collected during 33 different disasters between 2012 and 2019 (e.g., *wildfires, earthquakes, hurricanes, bombings,* and *floods*). Human experts and volunteers were employed to label the tweets with 25 information types.
- **COVID-19 Tweets:** This dataset contains a collection of tweets about the COVID-19 pandemic. In total, the dataset has $7,590$ tweets labelled with one or more information types (the same as for the TREC-IS dataset).

Figure 6.3 shows the distribution of tweets per information type in both datasets. As we can see, the datasets are highly imbalanced with respect to the distribution of tweets across different information types. For example, in the TREC-IS dataset, there are more than $6k$ tweets that are categorized into the information types *Hashtags*, *News*, *MultimediaShare*, and *Location*. In contrast, the information types *CleanUP, InformationWanted*, and *MovePeople* have significantly fewer tweets. Similarly, in COVID-19 Tweets, the tweets' distribution is extremely imbalanced; most tweets are categorized into *Irrelevant, ContextualInformation, Advice,* or *News*. Because of this skewing in the tweets' distribution, the multi-label classification of tweets becomes a more challenging task. This is because the model is more likely to learn to predict the more common information types and will be less likely to learn to predict the less common information types.

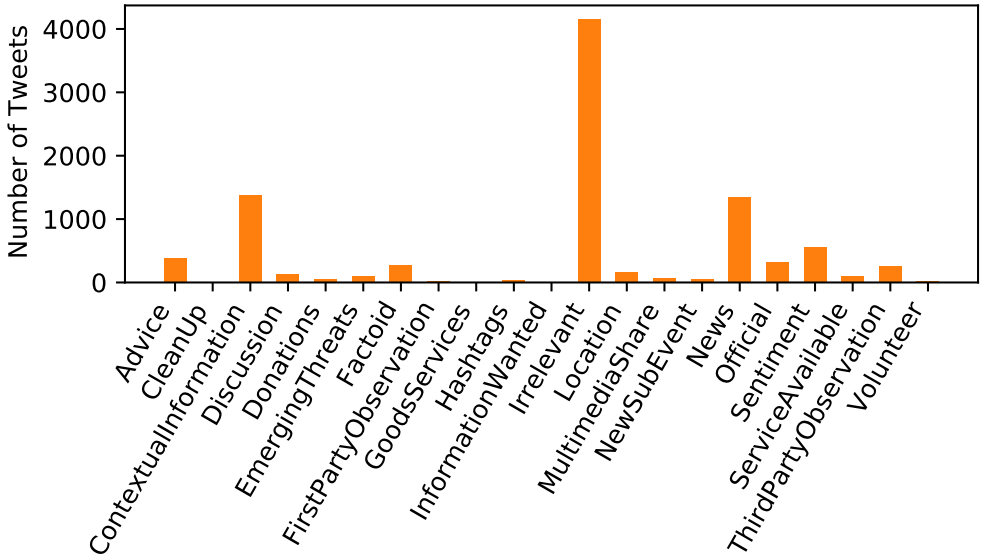
6.3.2 Baselines

We considered different approaches for multi-label classification as baselines in our evaluation. We briefly describe each baseline as follows:

- **TextCNN** (Kim, 2014) utilizes a convolutional neural network to construct text representation. First, it applies multiple convolution filters, followed by a max-



(a) Tweets distribution in TREC-IS dataset



(b) Tweets distribution in COVID-19 dataset

Figure 6.3: Tweets distribution of all information types in both datasets (TREC-IS and COVID-19 Tweets)

pooling layer to construct an embedding vector representation. At the final layer, the authors employ a dense with a Softmax function to predict the tweets classes.

- **HAN** (Yang et al., 2016) uses a hierarchical attention neural network to represent tweets. Two levels of attention mechanisms are applied at the word and sentence levels, allowing differential attention to be paid to more and less important content during constructing the tweet's representation.

- **BiLSTM** (Zhou et al., 2016) is a bidirectional LSTM model that parses an input text from left to right and right to left, then uses the final hidden state as a feature representation of the input text. Finally, a dense layer is added on top of BiLSTM layers with a Softmax function to compute final outputs.
- **MAGNET** (Pal et al., 2020) employs a bidirectional LSTM with BERT embeddings to represent tweets and GAT for labels classifiers. Then it uses a dot-product function to compute similarities between the tweet's vector and the labels' vectors. The most similar labels are returned as multi-label output.

6.3.3 Implementation and Preprocessing

We obtained the implementation of baseline methods from their GitHub repositories: TextCNN[3], HAN[4], and BiLSTM[5]. Moreover, we implemented the code for the MAGNET model since it has not been open-sourced to date. We followed the hyperparameter settings reported in the original papers for the baseline models. We performed a grid search method to optimize the hyperparameters, seeking optimal values that yield the best performances. Specifically, we obtained the best results with the following values: *training-epochs* of 200, *batch-size* of 128 and *Adam* optimizer (Kingma and Ba, 2015) with a *learning-rate* of $2e^{-5}$. To avoid overfitting, we added a `dropout` layer with a rate of 0.25 and applied an *early stopping* technique during the model's training. The implementation of our approach (I-AID) is open-source and available on the GitHub repository.[6]

Data Preprocessing: We performed the following steps to process tweets:

- We used the *NLTK's TweetTokenize*[7] API to tokenize tweets.
- We removed stop words, URLs, usernames, and Unicode characters.
- We eliminated extra white spaces, repeated full stops, question marks, and exclamation marks.
- We converted emojis to text using the *emoji*[8] python library.
- We used the SpaCy[9] library to extract *named-entities* from tweets.

[3] https://github.com/delldu/TextCNN
[4] https://github.com/tqtg/hierarchical-attention-networks
[5] https://github.com/yezhejack/bidirectional-LSTM-for-text-classification
[6] https://github.com/dice-group/I-AID
[7] https://www.nltk.org/api/nltk.tokenize.html
[8] https://pypi.org/project/emoji/
[9] https://github.com/explosion/spaCy

6.3.4 Evaluation Metrics

We adopted the standard metrics for evaluating multi-label predictions. Specifically, we used a *weighted-average* F_1 score, *Hamming loss* and *Jaccard index* to evaluate the performance:

- **Weighted-average F_1:** This metric computes the harmonic mean of Precision and Recall scores (Takahashi et al., 2022). We calculated F_1 scores for each label independently using a *weighted-average* approach, then added them together and applied a weight relative to the total number of tweets in each label.

$$\text{F1}_{w.avg.} = 2\sum_{i=1}^{k} \frac{|T_{\lambda_i}|}{|T|} \frac{Precision_{\lambda_i} \times Recall_{\lambda_i}}{Precision_{\lambda_i} + Recall_{\lambda_i}}, \tag{6.4}$$

  where $|T_{\lambda_i}|$ denotes the number of tweets with label λ_i and $|T|$ is the total number of tweets.

- **Hamming Loss:** To estimate the error rate in the predicted labels, we used the *Hamming loss* function (Schapire and Singer, 1999), which computes the fraction of incorrectly predicted labels out of all predicted labels. Lower *Hamming loss* scores indicate better performance.

$$hamming\ Loss\ (y^{(i)}, \hat{y}^{(i)}) = \frac{1}{S}\sum_{i=1}^{S} \frac{1}{k} |y^{(i)} \oplus \hat{y}^{(i)}|, \tag{6.5}$$

  where S denotes the dataset size, k represents the total number of labels (i.e., $|\Lambda|$), $\oplus$ signifies the XOR operator, and $y^{(i)}$ and $\hat{y}^{(i)}$ correspond to the ground-truth and predicted labels of tweet i, respectively.

- **Jaccard Index:** To evaluate the accuracy, we used the Jaccard index, which measures the similarity between predicted $\hat{y}^{(i)}$ and ground-truth labels $y^{(i)}$ (Gouk et al., 2016). Jaccard index computes the ratio of common labels in two sets of all labels as follows:

$$jaccard\ (y^{(i)}, \hat{y}^{(i)}) = \frac{|y^{(i)} \cap \hat{y}^{(i)}|}{|y^{(i)} \cup \hat{y}^{(i)}|}, \tag{6.6}$$

  where y_i and $\hat{y}_i$ denote the ground-truth and predicted labels for a tweet i. $\cap$ and $\cup$ represent intersection and union set operations, respectively.

Table 6.3: The results of our approach (I-AID) and baselines under metrics: weighted-average F_1, Hamming Loss and Jaccard Index. The best results are in bold.

Datasets	Metrics	Baselines				I-AID
		TextCNN	HAN	BiLSTM	MAGNET	
TREC-IS	$F1_{w.avg.}$	0.25	0.37	0.31	0.53	**0.59**
	Jaccard Index	0.18	0.28	0.19	0.38	**0.43**
	Hamming Loss	0.24	0.15	0.26	0.09	**0.07**
COVID-19 Tweets	$F1_{w.avg.}$	0.47	0.40	0.43	0.51	**0.55**
	Jaccard Index	0.33	0.28	0.21	0.40	**0.43**
	Hamming Loss	0.11	**0.04**	0.07	0.12	0.08

6.3.5 Evaluating Actionable Information

Our main goal is to evaluate the effectiveness of our approach in identifying actionable information in tweets. Actionable information refers to any information that requires an immediate response or intervention, such as requests for search and rescue or reports of emerging threats. For this purpose, we adopted the *Accumulated Alert Worth* (AAW) metric, which was specifically designed for evaluating systems that identify actionable information in the context of TREC-IS (McCreadie et al., 1970).

The AAW metric assigns a score between -1 and $+1$ to each tweet, where a positive score indicates a high-priority tweet that should trigger an alert, and a negative score indicates a low-priority tweet that should not trigger an alert. We provide a summary of the AAW metric (see Section 2.6.2 for more details) as follows:

$$AAW = \frac{1}{2} \sum_{t \in T} \begin{cases} \frac{1}{|T_h|} \cdot hPW(t) & \text{if } t \in T_h \\ \frac{1}{|T_l|} \cdot lPW(t) & \text{otherwise} \end{cases}, \tag{6.7}$$

where $\mathcal{T}_h$ and $\mathcal{T}_l$ are the sets of high and low-priority tweets, respectively. $hPW(t)$ function assigns a score for each tweet that should generate, while $lPW(t)$ function assigns a score to each tweet that should not trigger an alert.

6.3.6 Results and Discussion

Performance Comparison (Q_4): We evaluated the performance of our approach (I-AID) compared to the baseline methods using different metrics. All approaches were trained on the same training set and tested on the same test set to ensure a

fair comparison. Table 6.3 presents the evaluation results for each approach on the TREC-IS and COVID-19 Tweets datasets. We used the weighted-average F_1 score as the main criterion to rank all approaches. The weighted-average F_1 reflects the average performance across all information types. Overall, our approach, I-AID, achieves superior performance compared to the other baselines on several metrics. Specifically, our approach outperforms the MAGNET model –*the state-of-the-art baseline in multi-label tweets classification* – by $+6\%$, and $+4\%$ improved F_1 scores on the TREC-IS and COVID-19 Tweets datasets, respectively. We also used the Jaccard index and Hamming loss to evaluate the accuracy and error rate of the models. The Jaccard index indicates that our approach achieves higher accuracy than all baseline methods. Specifically, I-AID obtains 43% Jaccard index on both datasets, while MAGNET's score obtains 38% on the TREC-IS and 40% on the COVID-19 Tweets dataset. On the other hand, our approach yields sub-optimal results using Hamming loss. On the TREC-IS dataset, I-AID has a Hamming loss with 0.07%, which is the best among all models. On the Covid-19 Tweets, our approach achieves 0.08%, which is slightly higher than some baselines (HAN and BiLSTM).

To summarize our findings, our evaluation shows that I-AID can effectively classify disaster-related tweets into different types. This performance can be attributed to three factors: i) we use a multi-model framework that leverages contextualized embeddings from the BERT model and capture contextual information in tweets, ii) we incorporate label information into tweet representations and as well as structural information from the graph attention network and iii) we employ a RELATION NETWORK to detect similarities between tweets and labels using a learnable distance function.

Actionable Information in Tweets (Q_5): To identify tweets that contain *actionable* information, we used the AAW metric (Equation (6.7)). There are two ways to define *actionable* tweet (Zade et al., 2018): i) as the level of priority assigned by human assessors, or ii) as the type of information conveyed in the tweet. In our studies, we adopted the latter definition and evaluated our approach based on the information types. The evaluation results are presented in Table 6.4, where the top six rows correspond to the baselines, the middle rows show the AAW scores of the best-performing systems from the TREC-IS challenge 2019 (run B) (McCreadie et al., 2019), and the last row presents the result of our approach (I-AID). As shown in Table 6.4, I-AID achieved significant performance overall baseline methods; for high-priority tweets, I-AID obtained an absolute improvement of $+26\%$ AAW score higher than the MAGNET model and $+32\%$ higher than `nyu-smap`, the best-achieved result in TREC-IS 2019. It is noteworthy that our approach is the first to achieve a positive AAW score for high-priority tweets. Despite these promising results, we

Table 6.4: The evaluation results using the AAW metric on the TREC-IS test dataset (run B). A higher AAW value indicates better prediction

Systems	Accumulated Alert Worth (AAW)	
	High Priority	All
TextCNN	-0.9764	-0.4884
HAN	-0.7816	-0.4600
Bi-LSTM	-0.8760	-0.4482
BERT (UPB_BERT)	-0.9680	-0.4882
TEXTGAT	-0.9794	-0.4897
MAGNET	-0.9436	-0.4726
Median	-0.9197	-0.4609
BJUTDMS-run2	-0.9942	-0.4971
IRIT	-0.9942	-0.4971
irlabISIBase	-0.2337	-0.4935
UCDbaseline	-0.7856	-0.4131
nyu-smap	-0.1213	-0.1973
SC-KRun28482low	-0.9905	-0.4955
xgboost	-0.9942	-0.4972
UCDrunEL2	-0.8556	-0.4382
cmu-rf-autothre	-0.8481	-0.4456
I-AID	**0.2044**	**-0.1509**

believe that further research is still needed to reliably detect high-priority tweets that acquire urgent actions in real-case scenarios.

6.3.7 Ablation Study

Our approach, as described in Section 6.2, combines three main components (BERT-ENCODER, TEXTGAT, and RELATION NETWORK). To assess the contribution of the main components for tweet representations (BERT-ENCODER, and TEXTGAT), we conducted an ablation study where used them as standalone models for tweets classification: i) BERT-ENCODER, which only uses the BERT model for classifying tweets, and ii) TEXTGAT, which only relies on the graph attention network. Table 6.5 reports the evaluation results for each component. The system with BERT-ENCODER performed better than the one with TEXTGAT component. On the TREC-IS dataset, BERT-ENCODER obtains an F_1 score of 50%, while TEXTGAT reaches only 26%. This indicates that BERT-ENCODER is more effective in learning rich representations from short texts than TEXTGAT.

Table 6.5: The ablation study of I-AID Model

Datasets	Metrics	BERT-ENCODER	TEXTGAT	I-AID
TREC-IS	$F1_{w.avg.}$	0.50	0.26	0.59
	Jaccard Index	0.34	0.18	0.43
	Hamming Loss	0.11	0.24	0.07
COVID-19 Tweets	$F1_{w.avg.}$	0.47	0.36	0.55
	Jaccard Index	0.37	0.15	0.43
	Hamming Loss	0.10	0.17	0.05

On the same dataset, our approach (I-AID) achieved superior results with $+9\%$ improved F_1 score compared to the BERT-ENCODER and $+33\%$ compared to the TEXTGAT. We also observed these models achieved similar performances on the COVID-19 Tweet datasets. Our approach outperforms BERT-based and GAT-based baselines in F_1 scores by $+8\%$, $+19\%$, respectively. Noteworthy, the TEXTGAT model achieves better performance when predicting fewer labels as output. On COVID-19 Tweets with 12 labels, TEXTGAT achieved an F_1 score that is $+10\%$ higher than its performance on the TREC-IS dataset with 25 labels.

6.4 Summary and Conclusion

In this chapter, we presented our approach (I-AID), a multi-model approach for classifying tweets with multiple labels. Our approach consists of three components: BERT-ENCODER, TEXT-GAT, and RELATION NETWORK. The BERT-ENCODER captured local information, while the TEXTGAT component learned correlations between the tokens (words or named entities) of tweets and their potential labels. Finally, the RELATION NETWORK was used to determine the relevance of each label concerning the tweet content. The main contributions of our study could be summarized as follows: i) we showed that the combination of BERT-ENCODER and TEXTGAT enhanced the representation of short texts and significantly improved multi-label classification. ii) Transfer learning from pre-trained language models could effectively handle sparsity and noise in social media data (tweets), and iii) we highlighted the challenges of evaluating the multi-label classification task, which required appropriate metrics for fine-grained analysis. On the TREC-IS dataset, I-AID achieved its best-weighted average F_1 score of 0.59. The results clearly demonstrated the limitation of our approach in dealing with imbalanced datasets, which is a direction of our future work.

We plan to use data augmentation techniques (e.g., GPT-4 text generation) to generate synthetic tweets for under-represented classes. Furthermore, there is a lack of semantic resources that can be leveraged in disaster management. A possible extension of our work is to construct a disaster ontology of actionable information extracted from tweets. Using this semantic resource, critical information could be efficiently linked with their relevant crisis responses.

7 Keyphrases Extraction from Disaster-related Tweets

This chapter addresses the research questions $\mathbf{Q}_6$ and $\mathbf{Q}_7$ (see Section 1.4) for extracting keyphrases using pre-trained language models and knowledge graphs. The main content of this chapter is based on our publication work (Zahera et al., 2022), the author co-designed, evaluated and co-wrote the said paper. We also provide a review of the related in Section 3.3.

7.1 Overview

Keyphrase extraction process aims to identify a set of phrases that best describe a document. This technique has been used in various downstream applications, such as text summarizing, organizing, and indexing (Merrouni et al., 2020). Keyphrase extraction can be divided into: i) extracting present keyphrases (PKE) that appear in a document, and ii) generating absent keyphrases (AKG) that do not present in the original document but are essential for downstream applications (e.g., text summarization, indexing). Table 7.1 provides an example of extracting present and absent keyphrases.

In the context of disaster events, many tweets lack hashtags, which makes it difficult to filter and analyze them. For instance, emergency responders may not be able to find relevant tweets without hashtags and gain valuable insights. Keyphrase extraction from disaster-related tweets can improve disaster management by quickly and efficiently identifying essential trends, as shown in Figure 7.1). Additionally, many keyphrases are absent from tweets due to their informality, noise, and short length (tweet length is limited to 280 characters). Our study aims to not only *extract* present keyphrases from disaster-related tweets but also to *generate* absent keyphrases that are relevant and do not appear in a tweet. Previous works primarily focus on extracting *present* keyphrases from text using supervised learning (e.g., sequence labelling (Sahrawat et al., 2019)) and unsupervised learning (e.g., TextRank (Mihalcea and Tarau, 2004), YAKE (Campos et al., 2020)), however, generating *absent* keyphrases is a challenging task. A statistical study by Ye et al.

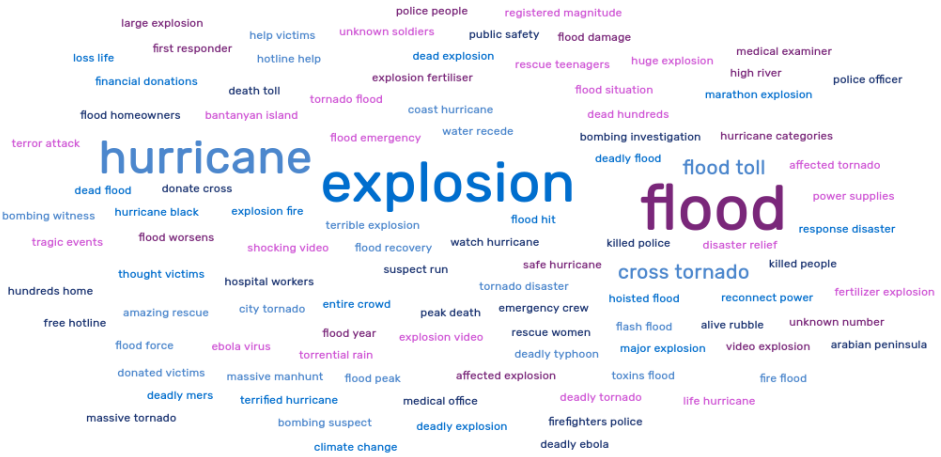


Figure 7.1: Wordcloud of top 100 keyphrases from tweets collect during tornado JOPLIN and hurricane SANDY

(2021) shows that some benchmark datasets, such as Inspec (hul, 2003)), miss up to 37.7% of *absent* keyphrases. Few studies have been proposed to cope with this challenge, one method is to use a supervised sequence-to-sequence with a *copy mechanism* that can copy relevant words from the source text instead of generating new ones Meng et al. (2017). However, this method requires a lot of labelled data for training. Further, the copy mechanism can only generate one word at a time without considering word dependencies (Zhao et al., 2021b). Another method is to use external knowledge sources to generate absent keyphrases. For example, Shen et al. (2022) creates a phrase bank with all keyphrases from a text corpus, assuming that keyphrases not present in one document could be in other related documents. However, this approach requires creating a domain-specific phrase bank for each dataset.

We propose an unsupervised multitask framework, called MULTPAX that uses *pre-trained language models* and *knowledge graphs* to reduce the effort of developing a keyphrase model. Our framework has the following pipeline: i) tokenizing and embedding the input document and its n-gram phrases as vectors in a shared semantic space, ii) *extracting* the phrases closest to the document's vector as present keyphrase candidates, and iii) *linking* the present keyphrases to knowledge graphs to find related terms (e.g., synonyms, hypernyms). For this purpose, we developed a new version of the MAG framework (Moussallem et al., 2017), optimized for linking keywords and extracting related terms; and iv) *ranking* all keyphrases (i.e., *present* and *absent*) based on their similarity to the input document, and returning the top-k phrases as the output.

Table 7.1: An example of present and absent keyphrase extraction from `Inspec` dataset. The predicted keyphrases are highlighted in green, and the absent ones are in red

Input Text	*"This paper shows the importance that management plays in the protection of information and in the planning to handle a security breach when a theft of information happens. Recent thefts of information that have hit major companies have caused concern. These thefts were caused by companies' inability to determine risks associated with the protection of their data and these companies' lack of planning to properly manage a security breach when it occurs." quoted from (Polstra III, 2005)*
Groundtruth Keyphrases	security breach, risk analysis, management issue, theft of information
Predicted Keyphrases	security breach, theft of information security management, security risk, data management

Additionally, we propose an improved metric for evaluating predicted keyphrases based on their *semantic matching* with ground-truth keyphrases. Existing studies (Liang et al., 2021; Meng et al., 2017; Zhao et al., 2021a) consider *Precision, Recall*, and F_1 based on the *exact matching* between predicted and ground-truth keyphrases, which works well for present keyphrases. However, this metric fails to capture the semantic similarity of absent keyphrases that have different words (Chowdhury et al., 2019). For instance, if *"Cryptocurrency"* is a ground-truth keyphrase, and a model generates *"Bitcoin"* as a predicted keyphrase, the exact matching metric considers them unrelated, even though they have semantic relatedness. By means of word embeddings, those words are similar and close in the embedding space. Therefore, we propose an embedding-based F_1-score for a more accurate evaluation of absent keyphrases. We evaluated MULTPAX's performance on four benchmark datasets and compared it with different baselines. The evaluation results show that our approach significantly outperformed state-of-the-art baselines with a significance t-test $p < 0.041$ and F_1 score up to 0.535. The main contributions of our study can be summarized as follows:

- We propose an *unsupervised* multi-task framework for extracting present keyphrases and generating absent ones.
- To the best of our knowledge, our approach is the first study to leverage *knowledge graphs* for keyphrase generation without the need for keyphrases vocabularies or phrase banks.

Table 7.2: A list of symbols used in this chapter

Symbol	Description
$\mathcal{D}$	A document of input text (e.g., disaster tweets)
S	A sentence in a document $\mathcal{D}$, it can also represent a tweet.
$\mathcal{T}$	Tokens of a sentence S
$\mathcal{Y}^p$	A set of extracted present keyphrases
$\mathcal{Y}^a$	A set of generated absent keyphrases
$\mathcal{Y}^{gold}$	A set of ground-truth keyphrases
$\mathcal{H}_i$	A contextualized embedding of a sentence i
$\mathcal{H}_\mathcal{D}$	A contextualized embedding of a document $\mathcal{D}$
k	A number of top relevant keyphrases to a document
C_i	A candidate link for a pre-marked entity i in the search index
a_p	The authority score computed by the Hits algorithm
h_p	The hub score for a node p
$P@k$	The precision scores of top-k ranked keyphrases
$R@k$	The recall scores of top-k ranked keyphrases
$F_1@k$	The F_1 scores of top-k ranked keyphrases

- We propose an *embedding-based* F_1 that considers the semantic similarity between generated and ground-truth keyphrases for precise evaluation of absent keyphrases.
- We conducted several experiments on four benchmark datasets, and the evaluation results demonstrate the efficacy of our approach keyphrase extraction compared to several baseline methods. The implementation of our approach and the datasets are available at the Github repository.[1]

7.2 Our Approach

This section describes our approach for extracting and generating *present* and *absent* keyphrases. Figure 7.2 depicts the architecture of our MULTPAX framework, which consists of three components: i) *present keyphrase extraction* (PKE), ii) *absent keyphrase generation* (AKG), and iii) *Keyphrases Semantic Matching*.

7.2.1 Problem Formulation

Given an input $\mathcal{D}$ with $|S|$ sentences; each sentence $s \in S$ is a sequence of $|s|$ tokens $\mathcal{T} = \{t_1, t_2, \cdots, t_{|s|}\}$. Our goal is to build a keyphrase model that can extract

[1] https://github.com/dice-group/MultPAX

present keyphrases $\mathcal{Y}^p = \{y_1^p, y_2^p, \cdots, y_{|\mathcal{Y}^p|}^p\}$ and also generate *absent keyphrases* $\mathcal{Y}^a = \{y_1^a, y_2^a, \cdots, y_{|\mathcal{Y}^a|}^a\}$ relevant to $\mathcal{D}$ using knowledge graphs. Following previous works (Gollapalli et al., 2017; Sahrawat et al., 2020), we divide keyphrase extraction into three sub-tasks: *Present Keyphrase Extraction* (PKE), *Absent Keyphrase Generation* (AKG) and *Semantic Matching*. First, we consider PKE as a *ranking* problem, where we extract and rank candidate keyphrases based on their similarities to the input document (see Section 7.2.2). Second, we formulate AKE as a *linking* problem to infer relevant information from knowledge graphs. We use an unsupervised *entity-linker* (Shen et al., 2014) that maps a present keyphrase ($\mathcal{Y}^p$) to its corresponding entity (i.e., resource) in a knowledge graph (e.g., DBPEDIA, BABELNET) and then get relevant terms (e.g., from `dct:subject`, `gold:hypernym` properties) as absent keyphrases candidates. Finally, we rank all keyphrases $\mathcal{Y}^p \cup \mathcal{Y}^a$ based on their similarities to $\mathcal{D}$ and return the top-k keyphrases as the output.

7.2.2 Present Keyphrase Extraction

We employ the BERT model (Devlin et al., 2019) to extract present keyphrases based on their semantic similarity to a document. The main steps are as follows: i) We tokenize an input document $\mathcal{D}$ into *n-gram phrases* and annotate each token with part-of-speech tags (e.g., ADJ: adjectives, NOUN: nouns, VERB: verbs). We remove stop words and keep noun phrases comprising zero or more adjectives followed by one or multiple nouns (Wan and Xiao, 2008). ii) We encode the candidate keyphrases and the input document as embedding vectors using the pre-trained language model (`BERT-Encoder`) in a shared semantic space. As discussed earlier in Section 2.3.2, a special preprocessing is applied to the input text of the `BERT-Encoder`. A `[CLS]` token is added at the beginning of each sentence, which is used to obtain the contextualized embedding vector of that particular sentence. An additional token `[SEP]` is inserted to mark the end of a sentence. Further, the input is tokenized by the *WordPiece* tokenizer (Song et al., 2021); each token t_i has three types of embeddings: *token embeddings* (E_{t_i}) for the vocabulary index, *segmentation embeddings* for the input sentence (E_A or E_B), and *position embeddings* (E_i) for the word position. The output of the `BERT-Encoder` is the sentence's representation matrix $\mathcal{H} = [h_0, h_1, \cdots h_{|s|}]$, where h_i is the embedding vector of token t_i. Formally, the embedding vector of a sentence s_j is

$$\mathcal{H}_j = \texttt{BERT-Encoder}(\{t_1, t_2, \cdots t_{|s|}\}). \tag{7.1}$$

Pooling is an essential operation for creating sentence and document embeddings (Chen et al., 2018). It is commonly used to aggregate (e.g., mean, max)

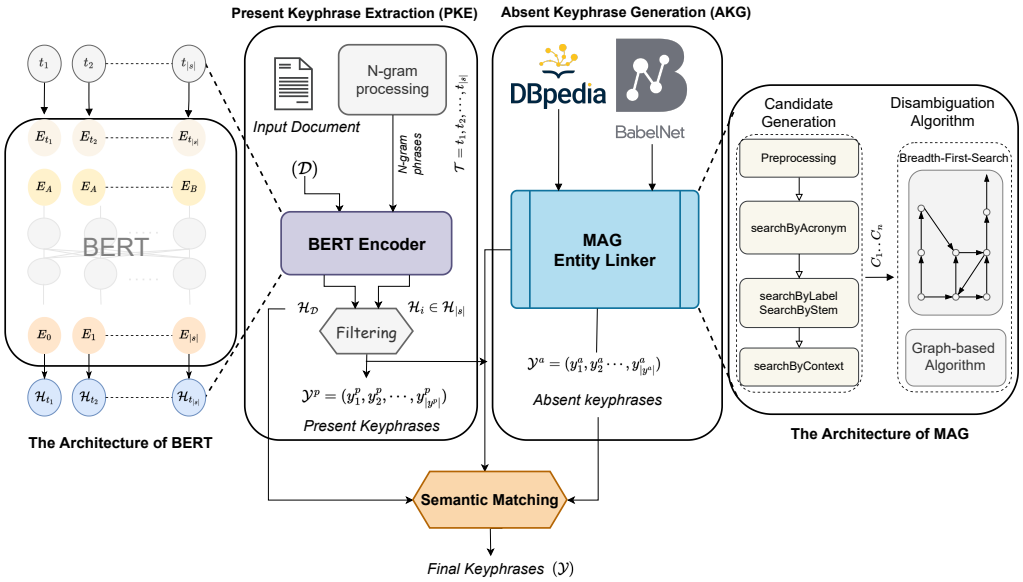


Figure 7.2: The architecture of MULTPAX framework with components: Present Keyphrase Extraction, Absent Keyphrase Generation and Semantic Matching

multiple representations (e.g., sentences) into one embedding vector. We use a `Max-Pooling` to aggregate all sentences' vectors into one for the document representation $\mathcal{H}_\mathcal{D}$. Formally,

$$\mathcal{H}_\mathcal{D} = \texttt{MaxPooling}(\{\mathcal{H}_1, \mathcal{H}_2, \cdots \mathcal{H}_{|S|}\}). \tag{7.2}$$

Finally, we use Cosine distance to measure similarities between the embedding vectors of candidate keyphrases $\mathcal{H}_i \in H_{|S|}$ and the document vector $\mathcal{H}_\mathcal{D}$. We select the top-k keyphrases as present keyphrases candidates.

7.2.3 Absent Keyphrase Generation

To obtain absent keyphrases, we first link all present keyphrases $\mathcal{Y}^p$ to a knowledge graph and get additional surface forms (i.e., strings that could be synonyms or alternative names). We use DBPEDIA knowledge graph, which covers a wide range of common entities and provides their surface forms. For *entity linking*, we adopt a similar approach to the MAG framework (Moussallem et al., 2017).

The MAG framework includes two stages: *candidate generation* and *candidate disambiguation*, to extract entity links. In the *candidate generation* stage, MAG identifies candidate links $(C_1, \ldots, C_n)$ for pre-marked entities in the search index. To achieve this, MAG uses acronyms and labels from a knowledge graph to map pre-marked

entity spans from the input text to candidate entities. Furthermore, MAG also relies on the Concise Bounded Description (CBD)[2] of the entities in a knowledge graph, comparing the context of entity spans in the input document with the CBD of an entity in a knowledge graph. We apply the candidate generation step from MAG to the present keyphrases extracted by the PKE component. In the *candidate disambiguation* stage, MAG creates a local graph using a breadth-first-search method for all candidate entities on a knowledge graph. Then, MAG applies the HITS ranking algorithm (Kleinberg, 1999) to jointly rank candidate links for all entities in the local graph. HITS ranks nodes in a directed graph based on incoming and outgoing edges. Authorities are the nodes that contain essential information, while hubs are the nodes that point to numerous authority nodes. Therefore, the authority score a_p of a node p is computed based on the hub score of the nodes that have directed edges to p. Further, the hub score h_p of p is calculated based on the authority score of the nodes linked by p (Kleinberg, 1999). Formally, HITS calculates the authority score a_p for the node p as

$$a_p = \sum_{q:(q,p)\in G} h_q, \tag{7.3}$$

where h_q is the hub score for a node q, given that there is an a directed edge from q to p in the graph G. The hub score h_p for a node p is calculated as

$$h_p = \sum_{q:(q,p)\in G} a_q, \tag{7.4}$$

where a_q is the authority score for a node q, which is linked by node p (Kleinberg, 1999). a_q and h_p are initialized randomly and updated iteratively until convergence. Unlike MAG, our approach not only links present keyphrases but also extracts related terms for each linked keyphrase from a knowledge graph. Moreover, we extract the top-ranked candidates for each entity and n nodes with the highest authority scores in the local graph, as their surface forms could be candidates for absent keyphrases. In our approach, we also use BABELNET to find *hypernyms* for the present keyphrases, besides the surface forms from DBPEDIA.

7.2.4 Keyphrases Semantic Matching

The last component in our approach aims to identify the top-k relevant keyphrases (*present* and *absent*), where we set $k = \{5, 10, 20\}$ in our experiments. We regard this task as a semantic textual similarity problem (Majumder et al., 2016). To measure the semantic relatedness between a document $\mathcal{D}$ and candidate keyphrases, we

[2] https://www.w3.org/Submission/CBD/

Table 7.3: Overview of the datasets (#Doc: number of documents, #Test: size of test set, #Avg. KP: average keyphrase per document, #Ratio%: percentage of absent keyphrase per dataset).

Dataset	#Doc	#Test	Avg. KP	Ratio%
Inspec	2k	500	7.65	37.7%
Krapivin	2.3k	460	3.03	15.3%
SemEval2010	144	100	7.15	11.3%
NUS	211	211	2.71	17.8%

embed them into a common semantic space using the pre-trained BERT model. We then use Cosine distance to rank the top-k closest keyphrases ($\mathcal{H}_i$) to the document's vector $\mathcal{H}_D$ and return them as the final keyphrase predictions. Formally,

$$\text{Cos}\ (\mathcal{H}_i, \mathcal{H}_\mathcal{D}) = \frac{\mathcal{H}_i \cdot \mathcal{H}_\mathcal{D}}{||\mathcal{H}_i|| \times ||\mathcal{H}_\mathcal{D}||}, \tag{7.5}$$

where $\mathcal{H}_i$ denotes the embedding vector of a candidate keyphrase (*present* y_i^p or *absent* y_i^a), and $\mathcal{H}_\mathcal{D}$ represents the embedding vector of the input document.

7.3 Experiments

This section describes the setup of our experiments, including the datasets, the baselines, and evaluation metrics. We aim to answer the research questions Q_6 and Q_7 (see Section 1.4), which examine how well our approach in extracting present keyphrases from disaster-related tweets. We also investigated the suitability of existing exact-matching metrics (i.e., Precision, Recall, and F_1-score) for evaluating absent keyphrases. Furthermore, we conducted an ablation study to measure the impact of each component on the overall performance.

7.3.1 Experimental Setup

Datasets. We used four benchmark datasets (English corpus) in our experiments, namely, *Inspec* (hul, 2003), *SemEval2010* (Kim et al., 2010), *Krapivin* (Krapivin et al., 2009), and *NUS* (Vijayakumar et al., 2018). Table 7.3 provides an overview of each dataset, including the total number of documents (#Doc.), the number of documents in the evaluation set (#Test), the average keyphrases per document (Avg. KP) and the ratio of absent keyphrases in each dataset (Ratio%).

Baselines. We compared our approach with the following baseline methods:

- **TextRank:** (Mihalcea and Tarau, 2004) This is an unsupervised approach that constructs a graph representation of a document, where nodes are phrases and edges are based on lexical similarities. Then, TextRank uses the *PageRank* algorithm to extract present keyphrases.
- **YAKE** (Campos et al., 2020) This is a simple unsupervised method that extracts keywords by leveraging statistical features such as word co-occurrence and frequency.
- **EmbedRank** (Bennani-Smires et al., 2018) This is an unsupervised method that leverages word embeddings to identify words relevant to a document as candidate keyphrases. Additionally, EmbedRank applies the *Maximum Marginal Relevance* algorithm to increase the diversity of the extracted keyphrases.
- **CopyRNN** (Meng et al., 2017) This is a supervised baseline that trains a sequence-to-sequence framework with a *copy mechanism* on the KP20K dataset (Meng et al., 2017). This approach serves as a baseline for both present keyphrase extraction and absent keyphrase generation, allowing us to compare the performance of copy mechanism in generating keyphrases.
- **AutoKeyGen** (Shen et al., 2022) This is an unsupervised approach that constructs a *phrase bank* by aggregating keyphrases from all documents in a corpus. AutoKeyGen uses lexical- and semantic-level similarities to select the top candidate keyphrases (present and absent) for each document.

Evaluation Metrics. We evaluated the performance of our approach and the baseline methods using different metrics. In the following, we describe how we computed the *Precision*, *Recall*, and F_1 scores for keyphrase extraction task. *Precision* measures the proportion of correctly matched keyphrases among all predicted keyphrases. Given a list of predicted keyphrases $\mathcal{Y} = (y_1, \ldots, y_{|\mathcal{Y}|})$, we select the top-$k$ ranked keyphrases $\mathcal{Y}_{:k} = (y_1, \ldots, y_{\min(k,|\mathcal{Y}|)})$ and compare them with the top-k ranked keyphrases in the ground-truth set. We set $k = \{5, 10\}$ for present keyphrases and $k = \{10, 20\}$ for absent ones in our experiments. Following previous works (Shen et al., 2022; Ye and Wang, 2018), we used the *Porter Stemmer* from the `NLTK library`[3] `v3.7` to determine exact matches between the top-k predicted ($\mathcal{Y}_{:k}$) and the ground-truth ($\mathcal{Y}^{gold}$) keyphrases. The Precision of the top-k predicted keyphrases is defined as:

[3] https://www.nltk.org/index.html

$$P@k = \frac{|\mathcal{Y}_{:k} \cap \mathcal{Y}^{gold}|}{|\mathcal{Y}_{:k}|}. \tag{7.6}$$

Recall measures the proportion of correctly matched keyphrases among all ground-truth keyphrases. Formally, Recall is defined as

$$R@k = \frac{|\mathcal{Y}_{:k} \cap \mathcal{Y}^{gold}|}{|\mathcal{Y}^{gold}_{:k}|}, \tag{7.7}$$

and the $F_1@k$-score is defined as the harmonic mean of $P@k$ and $R@k$

$$F_1@k = 2 \times \frac{P@k \times R@k}{P@k + R@k}. \tag{7.8}$$

Although the *exact-matching* metric has been used widely in the literature (Liang et al., 2021), there remains potential for improvement in evaluating absent keyphrases evaluation based on semantic similarity. Consequently, we propose in Section 7.3.3 a semantic matching between the predicted and ground-truth keyphrases based on their semantic relatedness.

Hyperparameters. We used a grid search method to optimize the hyperparameters of our approach. We obtained the best F_1 scores with the following values. For the PKE component, we tokenized the input text into phrases of $2-4$ grams and selected the top-10 ranked phrases as candidates for present keyphrases. For the baseline methods, we set the hyperparameters according to their original papers. In the MAG framework, we modified the extraction of common entities to cover a boarder range of entity types. Moreover, we used the standard configuration[4] of the MAG framework for the other hyperparameter values.

7.3.2 Present Keyphrase Evaluation

To answer Q_6, we compared the effectiveness of our approach (MULTPAX) with different baselines in extracting *present* keyphrases. As shown in Table 7.4, MULTPAX significantly surpasses all baselines by a large margin on most of the datasets, with a significant *t-test* $p < 0.041$. This performance can be attributed that MULTPAX employs semantic similarity between candidate keyphrases and the input document using a start-of-the-art pre-trained model in semantic textual matching (Xia et al.,

[4] https://github.com/dice-group/AGDISTIS/blob/master/src/main/resources/config/agdistis.properties

Table 7.4: The evaluation results of present keyphrases extraction on Inspec, SemEval2010, Krapivin, and NUS datasets. $F_1@k$ scores are reported based on exact-matching between the predicted and ground-truth keyphrases.

Model	Inspec		SemEval2010		Krapivin		NUS	
	$F_1@5$	$F_1@10$	$F_1@5$	$F_1@10$	$F_1@5$	$F_1@10$	$F_1@5$	$F_1@10$
TextRank	0.263	0.279	0.183	0.181	0.148	0.139	0.187	0.195
YAKE	0.027	0.038	0.050	0.242	0.013	0.020	0.013	0.020
EmbedRank	0.295	0.344	0.108	0.145	0.131	0.138	0.103	0.134
CopyRNN	0.292	0.336	0.291	**0.296**	0.302	0.252	0.342	0.317
AutoKeyGen	0.303	**0.345**	0.187	0.240	0.171	0.155	0.218	0.233
MULTPAX	**0.371**	0.210	**0.449**	0.255	**0.384**	**0.334**	**0.535**	**0.344**

Table 7.5: The evaluation results of absent keyphrases generation (in terms of R@10, R@20). All results are reported based on exact-matching between the predicted and ground-truth keyphrases, except the last row shows Recall results based on semantic-matching

Model	Inspec		SemEval2010		Krapivin		NUS	
	R@10	R@20	R@10	R@20	R@10	R@20	R@10	R@20
CopyRNN	0.051	0.068	0.049	0.057	0.116	0.142	0.078	0.10
AutoKeyGen-Bank	0.015	0.017	0.007	0.009	0.031	0.041	0.021	0.026
AutoKeyGen-Full	0.017	0.021	0.010	0.011	0.033	0.054	0.024	0.032
$\text{MULTPAX}_{\text{exact}}$	0.079	0.080	–	–	–	–	0.017	0.017
$\text{MULTPAX}_{\text{semantic}}$	0.696	0.584	–	–	–	–	0.608	0.669

2021). In contrast, COPYRNN (Meng et al., 2017) and AUTOKEYGEN (Shen et al., 2022) employ sequence-to-sequence models to encode an input document as low-dimensional vectors and decode it back into sequences of predicted keyphrases.

We also observed that YAKE exhibits suboptimal performance in detecting present keyphrases from short texts, such as paper abstracts. This is because YAKE relies on statistical features like word co-occurrence and frequency, which are only efficiently computed in long texts, such as full papers or news articles. Remarkably, the embedding-based baseline, (EMBEDRANK), achieves comparable results; however, it fails to generate absent keyphrases. In our approach, we extract present keyphrases from text using contextualized embeddings and semantic matching. We conclude that pre-trained language models are not only efficient at identifying present keyphrases without labelled data but also outperform the state-of-the-art approach (AUTOKEYGEN).

7.3.3 Absent Keyphrase Evaluation

To address Q_7 that investigates the performance of our approach in generating absent keyphrases, we conducted additional experiments comparing our approach to two baselines, namely, CopyRNN and AutoKeyGen. Following previous work (Shen et al., 2022), we employed the Recall metric (R@10, R@20) based on *exact-matching* for performance evaluation, as shown in Table 7.5. Since we used the same experimental setup as the CopyRNN and AutoKeyGen approaches, we obtained the evaluation results from their respective papers (Meng et al., 2017; Shen et al., 2022).

Regarding the research question Q_7, we can clearly see that all approaches achieve poor performances when considering exact matches between predicted and ground-truth keyphrases. For example, if two keyphrases are semantically similar, such as *"disaster relief organization"* and *"crisis responses institute"*, these keyphrases are not be considered as a *match* using existing metrics. Consequently, we found that such metrics are unsuitable for evaluating absent keyphrases and propose an improved evaluation metric based on the *semantic-matching*. Formally, let $\mathcal{Y}^a$ represents predicted keyphrases, and $\mathcal{Y}^{gold}$ denotes ground-truth keyphrases. We first embed each keyphrase in $\mathcal{Y}^a$ and $\mathcal{Y}^{gold}$. Then, we use Cosine distance to compute similarities between the embeddings of each keyphrase in $\mathcal{Y}$ and $\mathcal{Y}^{gold}$. We set a threshold (> 0.5) for similarity scores to consider semantic matching between $\mathcal{Y}$ and $\mathcal{Y}^{gold}$. The two last rows in Table 7.5 present the evaluation results of R@10 and R@20 based on semantic-matching compared to exact-matching in absent keyphrase extraction. The AutoKeyGen baseline demonstrates competitive performance in generating absent keyphrases on the `NUS` dataset. However, the keyphrases generated by AutoKeyGen are limited to those from the phrase bank of each dataset. In contrast, our approach leverages public knowledge graphs, such as DBpedia and BabelNet, to obtain relevant phrases as candidates for absent keyphrases.

Limitation of our work. In our experiment, we used the MAG framework to link present keyphrases to the DBpedia knowledge graph (see Section 7.2.3). In the `SemEval2010` and `Krapivin` datasets, we were unable to link the present keyphrases due to the insufficient coverage of these keyphrases in the DBpedia knowledge graph. This limitation accounts for the missing values shown in the last two rows of Table 7.5 for these datasets. In our future work, we plan to integrate additional knowledge graphs, such as YAGO and Wikidata, to extend the coverage of entity linking in the MAG framework.

Table 7.6: The ablation Study of MULTPAX framework on Inspec dataset. F_1@K-scores are reported based on semantic-matching between the predicted and ground-truth keyphrases

MULTPAX-variant	**F_1@5**	**F_1@10**
MULTPAX-PKE	0.892	0.686
MULTPAX-AKE$_{\text{BabelNet}}$	0.907	0.701
MULTPAX-AKE$_{\text{DBpedia}}$	0.911	0.727
MULTPAX$_{\text{Full}}$	**0.911**	**0.763**

7.3.4 Ablation Study

We analyzed the impact of each component of our framework on the overall performance. For this purpose, we developed four variants of our framework. The first variant, MULTPAX-PKE, focused only on extracting present keyphrases, without generating absent keyphrase generation or linking with knowledge graphs. We also created two variants of MULTPAX to assess the generation of absent absent keyphrases, namely MULTPAX-AKE$_{\text{DBpedia}}$ and MULTPAX-AKE$_{\text{BabelNet}}$. Moreover, we configured the MAG framework to link present keyphrases only with DBPEDIA in case of MULTPAX-AKE$_{\text{DBpedia}}$, and exclusively with BABELNET for MULTPAX-AKE$_{\text{BabelNet}}$. Finally, we evaluated the complete framework, MULTPAX$_{\text{Full}}$, as our fourth variant. Table 7.6 presents the evaluation results of each component in terms of *semantic-matching* F_1@5, and F_1@10 on the `Inspec` dataset, as it contains the highest ratio of absent keyphrases among the benchmark datasets. We observed that the performance of MULTPAX-PKE improves when it is linked with knowledge graphs; for instance, MULTPAX-AKE$_{\text{DBpedia}}$ outperforms MULTPAX-PKE by $+0.41$ in F_1@10. Additionally, we noted that our approach retrieves more terms from DBPEDIA than BABELNET, since DBPEDIA contains more semantic ontologies with approximately 3.5 million instances extracted from Wikipedia information boxes. Our MULTPAX-$_{\text{Full}}$ demonstrates improved performance, achieving F_1-scores of 0.911 in F_1@5, 0.763 in F_1@10 when incorporating both DBPEDIA and BABELNET knowledge graphs, compared to individual variants.

7.3.5 Use Case: Keyphrase Extraction from Crisis Tweets

In disaster situations, keyphrases can be particularly useful for finding relevant information that can improve situational awareness. To evaluate the effectiveness of our approach in the context of crisis data, we conducted a use-case experiment

Table 7.7: Keyphrase extraction from disaster-related tweets

Tweet:	"Severe flooding causing road closures and evacuations, please follow evacuation orders #flood #evacuation"
Keyphrases:	Flood, Road closures, Evacuation orders
Tweet:	"Breaking: Earthquake with magnitude 7.1 just struck the region. If you are safe, please check on your neighbours and report any injuries or damage to authorities."
Keyphrases:	Earthquake, magnitude 7.1, region, safe, neighbours, injuries, damage, authorities
Tweet:	"Devastating tornado just hit the town. Emergency services are overwhelmed with calls. If you are able, please consider donating blood to help those injured in the storm."
Keyphrases:	Tornado, town, emergency services, calls, donate blood, injured, storm

on a disaster-related dataset (Zhang et al., 2016), which contains $110K$ labelled tweets from different disaster events. Table 7.7 shows examples of keyphrases extracted from tweets during different disasters. We applied our approach (MULTPAX) to extract keyphrases, treating each tweet as a separate document. Unlike long documents, extracting keyphrases from tweets is more challenging due to their shortness. The experimental results reveal that our approach outperforms baseline methods in identifying present keyphrases, achieving in F_1@1 and F_1@3 of 0.58 and 0.67, respectively, in contrast to YAKE (0.047, 0.023), TEXTRANK (0.35, 0.37), and Embrank (0.27, 0.29). Nevertheless, we were unable to evaluate the performance of the keyphrases generated because there are no ground-truth phrases in this dataset. Overall, our findings suggest that keyphrase extracting can significantly improve situational awareness during disasters, particularly when dealing with short texts like tweets.

7.4 Summary and Conclusion

In this chapter, we presented the MULTPAX framework, a multi-task approach for extracting present and generating absent ones. The framework consisted of three components: i) Present Keyphrase Extraction, ii) Absent Keyphrases Generation, and iii) Semantic Matching. In our approach, we used a pre-trained language model (BERT) and knowledge graphs (DBPEDIA and BABELNET) to extract keyphrase from

documents. Our experiments demonstrated that pre-trained language models could effectively extract present keyphrases. Furthermore, knowledge graphs proved to be valuable resources for generating keyphrases that were absent, especially in a short text.

In our future work, we plan to apply a bootstrapped approach for extracting keyphrases from DBPEDIA abstracts to find more relevant terms. Specifically, we aim to apply MULTPAX iteratively on the abstracts of DBPEDIA entities. We will experiment with other knowledge graphs (e.g., YAGO and WIKIDATA) to extend the entity linking coverage in the MAG framework.

8 Conclusion

In this chapter, we present a summary of the insights gained through our studies into leveraging social media for disaster management. We summarize the main contributions of this thesis and conclude with potential research directions and the broader impact of our studies.

8.1 Summary

In this thesis, we studied the impact of using social media in disaster management. We conducted several experiments on real-world datasets, with the aim of answering the research question "***To what extent can we leverage social media in disaster situations?***". We also tackled various challenges involved in processing social media data, including i) collecting relevant data from Twitter (tweets), ii) applying specialized preprocessing steps to filter out noise and irrelevant information from tweets, and iii) extracting meaningful features. Our studies demonstrated that social media is a valuable source of information during crises. People often use platforms such as Twitter to share and obtain situational updates during crises (see examples in Figures 1.1 and 4.2). Moreover, authorities can also obtain valuable reports regarding affected individuals and consequent damages from social media data. Overall, our goal was to improve disaster management by developing efficient approaches for processing crisis data on social media. Concretely, we designed four novel approaches that facilitated the following:

- ***Early detection of disaster events***: In Chapter 4, we presented a novel approach for improving disaster prediction by joint learning from social media and environmental data. Our approach addressed the challenges posed by the noise and incompleteness of environmental data. We conducted our experiments on real-world environmental data about typhoons and their corresponding tweets. The evaluation results demonstrated the value of social media as a complementary information source to environmental data. By extracting meaningful features from social media, such as tweet volume and sentiment variances, we achieved significant improvements in typhoon prediction (up to 12.1% in F_1 scores).

- ***Multi-label classification of crisis-related tweets***: Extracting meaningful features from tweets is a challenging task due to their limited content, informal nature, and noise. Additionally, multi-label classification requires assigning one or more labels to an input tweet simultaneously, which requires an efficient feature representation. To cope with these challenges, we fine-tuned the BERT language model on domain-specific data curated from different crisis events. In Chapter 5, we provide the details of our approach and experiments on real-world disaster tweets collected from 22 crisis events. The evaluation results demonstrate that fine-tuning the BERT model is an efficient solution for classifying tweets into multiple information types. By categorizing disaster tweets into fine-grained types, disaster relief organizations can find relevant information and make proper decisions quickly. Furthermore, we conducted additional experiments to benchmark the performance of detecting *"actionable"* tweets, i.e., which contain such information as *"Move People"* or *"Ask for help"*. In this regard, we designed an improved approach called I-AID, which consists of three components (BERT-ENCODER, TEXTGAT) to detect actionable information, and RELATION NETWORK to match similarities between the input tweet and one or more information types. In Chapter 6, we describe our approach and experiments on two real-world datasets: i) TREC-IS, which contains thousands of related tweets collected from different disasters, and ii) COVID-19 tweets with pandemic-related content. To evaluate our approach, we used the *Accumulated Alert Worth* (AAW), which is designed to estimate the performance of algorithms for the detection of alerting tweets, i.e., tweets which contain highly critical information. The evaluation results indicate that our approach achieves superior performance with an absolute $+26\%$ in AAW compared with state-of-the-art baselines.

- ***Disaster event summarization***: Several studies have been conducted in the aftermath of disasters to conclude lessons learned and develop effective disaster management strategies. According to Imran et al. (2018), obtaining timely and accurate situational information from emerging disaster events is essential for rapid and effective disaster response. One approach to capturing a "big picture" of crisis events is to summarize related data using keyphrase extractive techniques. In our study, we employed extractive summarization using keyphrase extraction from disaster-related tweets. However, keyphrase extraction from tweets is more challenging compared to long documents. This is due to tweet characteristics such as shortness and noisiness. In addition, absent keyphrase generation is essential for categorizing and retrieving relevant tweets. Previous research has shown that many disaster-related tweets do not have user-provided hashtags, which makes it difficult to identify relevant information. For example, Chowdhury et al. (2020)

found that approximately $5,200$ rescue requests were shared on social media but missed by emergency responders due to the lack of relevant search terms or hashtags. In Chapter 7, we describe our approach, MULTPAX, for extracting present and absent keyphrases. Specifically, we used the BERT language model to extract present keyphrases and generate absent ones from knowledge graphs such as DBPEDIA and BABLENET. Our study showed that pre-computed resources, such as pre-trained language models and knowledge graphs, can significantly reduce the effort required to build an efficient keyphrase model compared to state-of-the-art baselines, which required building a domain-specific phrase bank.

8.1.1 Research Contributions

This thesis presents four significant contributions to processing social media data for disaster management. First, our study shows that multiple models trained in a *"joint or shared"* space outperform *"standalone"* models. In Chapter 4, we demonstrate that jointly learning from two sources of data leads to improved performance in detecting typhoons. Second, an end-to-end training between the pre-trained BERT model and a graph attention network in Chapter 5 leads to efficient identification of actionable information from crisis-related tweets. Third, our study also highlights the role of social media as a valuable source of information during emergencies. Although applications such as Facebook Crisis Response offer effective communication channels, there is still a need for advanced tools capable of detecting crisis events, identifying actionable information, and summarizing situational insights. To address this gap, this thesis provides four different approaches, namely JOINT-MODEL (Zahera et al., 2019b), UPB-BERT (Zahera et al., 2019a), I-AID (Zahera et al., 2021) and MULTPAX (Zahera et al., 2022), which can improve situational awareness and enable faster crisis responses. The thesis also demonstrates the significance of semantic information from knowledge graphs. For instance, embeddings representation from CONCEPTNET (i.e., semantic embeddings) improves the performance of event detection compared to traditional word embeddings. Additionally, knowledge graphs such as DBPEDIA, BABELNET prove to be rich resources for generating absent keyphrases without the need for creating phrase banks for each dataset.

To this end, our works provide a foundation for further research directions to enhance the efficiency of disaster management. In the next section, we present some potential research directions that arose while carrying out this work.

8.2 Open Challenges and Future Work

Our studies in processing crisis data for improving disaster management have raised a number of interesting (*open*) research questions.

- **Towards a crisis recommender system**: A central question regarding the future work of this thesis is how to design an action-based recommender system that supports crisis responders in taking timely actions. While existing disaster management systems monitor and track emerging events efficiently, disaster responses still heavily rely on human expertise. We believe disaster management systems should learn historical data from previous disasters and automatically recommend appropriate actions. Therefore, our open research question is: "*Can we develop a recommender system that not only monitors, and tracks disasters, but also recommends proper actions?*" Such a disaster response system could use machine learning to analyze data from various sources, such as weather forecasts, satellite imagery, and social media posts, to estimate the potential impact of a disaster. Based on this analysis, the recommender system could suggest how to prepare for and respond to a disaster. For example, it could suggest deploying specific resources, such as emergency shelters or medical supplies, to certain locations to reduce disaster impact.

- **Damage estimation from social media data**:

  Estimating damages after disasters is a time-critical process. Usually, human experts conduct field surveys, which can take weeks, to assess the damage in affected areas such as roads, bridges, and buildings. One approach to estimate the damages is using natural language processing to analyze social media content about the disaster. This analysis helps to identify common keywords and phrases (e.g., *damage*, *injured*, or *evacuated*), which enables the estimation of a disaster's overall impact, including the number of affected people and resulting damages. While our study focused on textual information from social media data, recent research has demonstrated promising results in rapidly estimating disaster damages using satellite images (Imran et al., 2022; Nguyen et al., 2017). We observe that shared images can also provide valuable information about damage and can be used to implement disaster recovery plans as early as possible. It is noteworthy that Alam et al. (2018) provides a benchmark dataset called *CrisisMMD* that contains thousands of annotated tweets and images collected during different crisis events. We believe that developing a multi-modal system capable of processing both text and images is a promising research direction for improving situational awareness and damage estimation.

- **Towards a crisis knowledge graph**:

  One of the most challenging aspects of processing crisis data is the lack of reliable resources that can model data semantics (i.e., linked data) for efficient linking and reasonable tasks. In other domains, such as criminology and health, researchers have developed knowledge graphs to enable semantically interoperable access to heterogeneous data sources. In the crisis informatics domain, Purohit et al. (2019) presented the design and requirements for creating knowledge graphs to support disaster management functions. Purohit et al. (2019) found that knowledge graphs can facilitate crisis management systems by querying critical resources to improve disaster preparedness, response, and recovery decisions. We believe this is a crucial research direction for semantifying the crisis data to develop rapid, comprehensible, and effective approaches. For instance, assume a hurricane is approaching a coastal city. Disaster managers can access a knowledge graph containing extensive information about the hurricane's location and intensity, as well as available resources for disaster response. This knowledge graph could be employed to identify high-risk areas for hurricanes, such as flood-prone and low-lying regions. In this way, a knowledge graph can offer disaster managers a comprehensive and accessible information source, assisting them in making more informed decisions about disaster response and community protection.

- **Crisis data augmentation**: One of the primary challenges in constructing effective disaster models is the lack of benchmark datasets. While there are a few available crisis-related datasets that we used in our experiments, these datasets suffer from class imbalance and require extensive hyperparameter tuning to extract meaningful features. We believe that data generation is now achievable through the use of advanced large-scale language models, such as GPT-4, which can be employed to augment tweets tailored to a particular disaster type. By providing augmented data, a machine learning model could be trained efficiently, avoid overfitting, and enhance its generalization to unseen data.

Bibliography

2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, Sapporo, Japan, July 11-12, 2003*.

Luca Maria Aiello, Georgios Petkos, Carlos J. Martín, David P. A. Corney, Symeon Papadopoulos, Ryan Skraba, Ayse Göker, Ioannis Kompatsiaris, and Alejandro Jaimes. 2013. Sensing trending topics in twitter. *IEEE Trans. Multim.*, 15(6):1268–1282.

Firoj Alam, Ferda Ofli, and Muhammad Imran. 2018. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018*, pages 465–473. AAAI Press.

Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. IEEE.

Reem ALRashdi and Simon O'Keefe. 2019. Deep learning and word embeddings for tweet classification for crisis response. *arXiv preprint arXiv:1903.11024*.

Amrita Anam, Aryya Gangopadhyay, and Nirmalya Roy. 2018. Evaluating disaster time-line from social media with wavelet analysis. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 41–48. IEEE.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Marco Avvenuti, Stefano Cresci, Mariantonietta N La Polla, Andrea Marchetti, and Maurizio Tesconi. 2014. Earthquake emergency management by social sensing. In *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, pages 587–592. IEEE.

Vimala Balakrishnan and Ethel Lloyd-Yemoh. 2014. Stemming and lemmatization: A comparison of retrieval performances. *Lecture Notes on Software Engineering*, 2(3).

Loris Belcastro, Fabrizio Marozzo, Domenico Talia, Paolo Trunfio, Francesco Branda, Themis Palpanas, and Muhammad Imran. 2021. Using social media for sub-event detection during disasters. volume 8, pages 1–22. Springer.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium. Association for Computational Linguistics.

Gilles Bernard and Georges Lebboss. 2017. Methods for word encoding: A survey. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. IEEE.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics*, 21(4):543–565.

Cody Buntain, Richard McCreadie, and Ian Soboroff. 2020. Incident streams 2020: TRECIS in the time of COVID-19. In *Proceedings of the Twenty-Ninth Text REtrieval Conference, TREC 2020, Virtual Event [Gaithersburg, Maryland, USA], November 16-20, 2020*, volume 1266 of *NIST Special Publication*. National Institute of Standards and Technology (NIST).

Grégoire Burel, Hassan Saif, and Harith Alani. 2017a. Semantic wide and deep learning for detecting crisis-information categories on social media. In *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, volume 10587 of *Lecture Notes in Computer Science*, pages 138–155. Springer.

Grégoire Burel, Hassan Saif, Miriam Fernandez, and Harith Alani. 2017b. On semantics and deep learning for event detection in crisis situations.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Cornelia Caragea, Adrian Silvescu, and Andrea H Tapia. 2016. Identifying informative messages in disaster events using convolutional neural networks. In *International Conference on Information Systems for Crisis Response and Management*, pages 137–147.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3163–3171.

Qian Chen, Zhen-Hua Ling, and Xiaodan Zhu. 2018. Enhancing sentence embedding with generalized pooling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1815–1826. Association for Computational Linguistics.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019. Title-guided encoding for keyphrase generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6268–6275. AAAI Press.

Xiaoyan Chen, Delu Pan, Xianqiang He, Yan Bai, and Difeng Wang. 2012. Upper ocean responses to category 5 typhoon megi in the western north pacific. *Acta Oceanologica Sinica*, 31(1):51–58.

Zhe Chen et al. 2003. Bayesian filtering: From kalman filters to particle filters, and beyond. *Statistics*, 182(1):1–69.

Mark Cheung and José MF Moura. 2020. Graph neural networks for covid-19 drug discovery. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5646–5648. IEEE.

Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. 2019. Keyphrase extraction from disaster-related tweets. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1555–1566. ACM.

Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. 2020. On identifying hashtags in disaster twitter data. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial*

Intelligence Conference, New York, NY, USA, February 7-12, 2020, pages 498–506. AAAI Press.

Abu Nowshed Chy, Umme Aymun Siddiqua, and Masaki Aono. 2021. Exploiting transfer learning and hand-crafted features in a unified neural model for identifying actionable informative tweets. *Journal of Information Processing*, 29:16–29.

Alfredo Cobo, Denis Parra, and Jaime Navón. 2015. Identifying relevant messages in a twitter-based citizen channel for natural disaster situations. In *Proceedings of the 24th international conference on world wide web*, pages 1189–1194.

Qi Dang, Feng Gao, and Yadong Zhou. 2016. Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks. *Expert Systems with Applications*, 57:285–295.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Han Dong, Milton Halem, and Shujia Zhou. 2013. Social media data analytics applied to hurricane sandy. In *Social computing (SocialCom), 2013 international conference on*, pages 963–966. IEEE.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 55–65. Association for Computational Linguistics.

Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. 2022. Graph neural networks for recommender system. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1623–1625.

Thomas Glade and Farrokh Nadim. 2014. Early warning systems for natural hazards and risks.

Sujatha Das Gollapalli, Xiaoli Li, and Peng Yang. 2017. Incorporating expert knowledge into keyphrase extraction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3180–3187. AAAI Press.

Henry Gouk, Bernhard Pfahringer, and Michael Cree. 2016. Learning distance metrics for multi-label classification. In *Asian Conference on Machine Learning*, pages 318–333. PMLR.

Xiangyang Guan and Cynthia Chen. 2014. Using social media data to understand and assess disasters. *Natural hazards*, 74(2):837–850.

Jingrui He, Wei Shen, Phani Divakaruni, Laura Wynter, and Rick Lawrence. 2013. Improving traffic prediction with tweet semantics. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1387–1393. AAAI Press.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 6(2):107–116.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Dynamic contextualized word embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6970–6984, Online. Association for Computational Linguistics.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (CSUR)*, 54(4):1–37.

J Brian Houston, Joshua Hawthorne, Mildred F Perreault, Eun Hae Park, Marlo Goldstein Hode, Michael R Halliwell, Sarah E Turner McGowen, Rachel Davis, Shivani Vaid, Jonathan A McElderry, et al. 2015. Social media and disasters: a functional framework for social media use in disaster planning, response, and research. *Disasters*, 39(1):1–22.

Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2018. Processing social media messages in mass emergency: Survey summary. pages 507–511.

Muhammad Imran, Shady Elbassuoni, Carlos Castillo, Fernando Diaz, and Patrick Meier. 2013. Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd international conference on world wide web*, pages 1021–1024.

Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 2016. Twitter as a lifeline: Human-annotated twitter corpora for NLP of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Muhammad Imran, Umair Qazi, Ferda Ofli, Steve Peterson, and Firoj Alam. 2022. AI for disaster rapid damage assessment from microblogs. pages 12517–12523.

Raymond Austin Jarvis and Edward A Patrick. 1973. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on computers*, 100(11):1025–1034.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2):494–514.

Prashant Khare, Grégoire Burel, and Harith Alani. 2018a. Classifying crises-information relevancy with semantics. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 367–383. Springer.

Prashant Khare, Grégoire Burel, Diana Maynard, and Harith Alani. 2018b. Cross-lingual classification of crisis data. In *International Semantic Web Conference*, pages 617–633. Springer.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632.

Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. Large dataset for keyphrases extraction.

Yury Kryvasheyeu, Haohui Chen, Nick Obradovich, Esteban Moro, Pascal Van Hentenryck, James Fowler, and Manuel Cebrian. 2016. Rapid assessment of disaster damage using social media activity. *Science advances*, 2(3):e1500779.

Peter M Landwehr and Kathleen M Carley. 2014. Social media in disaster relief. In *Data mining and knowledge discovery for big data*, pages 225–257. Springer.

Chenliang Li, Aixin Sun, and Anwitaman Datta. 2012. Twevent: segment-based event detection from tweets. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 155–164. ACM.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Unsupervised keyphrase extraction by jointly modeling local and global context. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 155–164. Association for Computational Linguistics.

Junhua Liu, Trisha Singhal, Lucienne TM Blessing, Kristin L Wood, and Kwan Hui Lim. 2021. Crisisbert: a robust transformer for crisis classification and contextual crisis embedding. In *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, pages 133–141.

Zhibin Lu, Pan Du, and Jian-Yun Nie. 2020. VGCN-BERT: augmenting BERT with graph embedding for text classification. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*, volume 12035 of *Lecture Notes in Computer Science*, pages 369–382. Springer.

Goutam Majumder, Partha Pakray, Alexander Gelbukh, and David Pinto. 2016. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas*, 20(4):647–665.

Sajid A Marhon, Christopher JF Cameron, and Stefan C Kremer. 2013. Recurrent neural networks. In *Handbook on Neural Information Processing*, pages 29–65. Springer.

Michael Mathioudakis and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158.

Richard Mccreadie. 2019. Trec-is v2 metrics.

Richard McCreadie, Cody Buntain, and Ian Soboroff. 1970. Incident streams 2019: Actionable insights and how to find them. *Incident Streams 2019: Actionable Insights and How to Find Them*.

Richard McCreadie, Cody Buntain, and Ian Soboroff. 2019. TREC incident streams: Finding actionable information on social media. In *Proceedings of the 16th International Conference on Information Systems for Crisis Response and Management, València, Spain, May 19-22, 2019*. ISCRAM Association.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 582–592. Association for Computational Linguistics.

Zakariae Alami Merrouni, Bouchra Frikh, and Brahim Ouhbi. 2020. Automatic keyphrase extraction: a survey and trends. volume 54, pages 391–424. Springer.

Stuart E Middleton, Lee Middleton, and Stefano Modafferi. 2013. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, 29(2):9–17.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual*

Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 3111–3119.

Taro Miyazaki, Kiminobu Makino, Yuka Takei, Hiroki Okamoto, and Jun Goto. 2019. Label embedding using hierarchical structure of labels for twitter classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6316–6321. Association for Computational Linguistics.

Melinda Morton and J Lee Levy. 2011. Challenges in disaster data collection during recent disasters. *Prehospital and disaster medicine*, 26(3):196–201.

Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. 2017. MAG: A multilingual, knowledge-base agnostic and deterministic entity linking approach. In *Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017*, pages 9:1–9:8. ACM.

Mulyanto Mulyanto, Muhamad Faisal, Setya Widyawan Prakosa, and Jenq-Shiou Leu. 2020. Effectiveness of focal loss for minority classification in network intrusion detection systems. *Symmetry*, 13(1):4.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5. IEEE.

Roberto Navigli and Simone Paolo Ponzetto. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.

Tahora H Nazer, Fred Morstatter, Harsh Dani, and Huan Liu. 2016. Finding requests in social media for disaster relief. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1410–1413. IEEE.

Tahora H Nazer, Guoliang Xue, Yusheng Ji, and Huan Liu. 2017. Intelligent disaster response via social media analysis a survey. *ACM SIGKDD Explorations Newsletter*, 19(1):46–59.

Dat T Nguyen, Ferda Ofli, Muhammad Imran, and Prasenjit Mitra. 2017. Damage assessment from social media imagery data during disasters. In *Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017*, pages 569–576.

RI Ogie, S James, A Moore, T Dilworth, M Amirghasemi, and J Whittaker. 2022. Social media use in disaster recovery: A systematic literature review. *International Journal of Disaster Risk Reduction*, page 102783.

Avital Y O'Glasser, Rebecca C Jaffe, and Michelle Brooks. 2020. To tweet or not to tweet, that is the question. In *Seminars in nephrology*, volume 40, pages 249–263. Elsevier.

Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2014. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014*. The AAAI Press.

Alexandra Olteanu, Sarah Vieweg, and Carlos Castillo. 2015. What to expect when the unexpected happens: Social media communications across crises. In *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*, pages 994–1009. ACM.

Wanli Ouyang and Xiaogang Wang. 2013. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*.

Ankit Pal, Muru Selvakumar, and Malaikannan Sankarasubbu. 2020. MAGNET: multi-label text classification using attention-based graph neural network. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020*, pages 494–505. SCITEPRESS.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2020. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1339.

Ruchi Parikh and Kamalakar Karlapalem. 2013. Et: events from tweets. In *Proceedings of the 22nd international conference on world wide web*, pages 613–620.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, pages 1063–1072.

Robert M Polstra III. 2005. A case study on how to manage the theft of information. In *Proceedings of the 2nd annual conference on Information security curriculum development*, pages 135–138.

David MW Powers. 2020. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Hemant Purohit, Rajaraman Kanagasabai, and Nikhil Deshpande. 2019. Towards next generation knowledge graphs for disaster management. In *2019 IEEE 13th international conference on semantic computing (ICSC)*, pages 474–477. IEEE.

Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu. 2016. Joint training of cascaded cnn for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3456–3465.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

A Reese. 2016. How we'll predict the next natural disaster: Advances in natural hazard forecasting could help keep more people out of harm's way. *Discover Magazine, Sep*.

Christian Reuter and Marc-André Kaufhold. 2018. Fifteen years of social media in emergencies: a retrospective review and future directions for crisis informatics. *Journal of contingencies and crisis management*, 26(1):41–57.

François Rousseau and Michalis Vazirgiannis. 2015. Main core retention on graph-of-words for single-document keyword extraction. In *European Conference on Information Retrieval*, pages 382–393. Springer.

Koustav Rudra, Siddhartha Banerjee, Niloy Ganguly, Pawan Goyal, Muhammad Imran, and Prasenjit Mitra. 2016. Summarizing situational and topical information during crises. *arXiv preprint arXiv:1610.01561*.

Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2019. Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. *arXiv preprint arXiv:1910.08840*.

Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. In *European Conference on Information Retrieval*, pages 328–335. Springer.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM.

Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2012. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):919–931.

Anita Saroj and Sukomal Pal. 2020. Use of social media in crisis management: A survey. *International Journal of Disaster Risk Reduction*, 48:101584.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

Robert E Schapire and Yoram Singer. 1999. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336.

Deepika Sharma and ME Cse. 2012. Stemming algorithms: a comparative study and their analysis. *International Journal of Applied Information Systems*, 4(3):7–12.

Yashvardhan Sharma and Sahil Gupta. 2018. Deep learning approaches for question answering system. *Procedia computer science*, 132:785–794.

Wei Shen, Jianyong Wang, and Jiawei Han. 2014. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Xianjie Shen, Yinghan Wang, Rui Meng, and Jingbo Shang. 2022. Unsupervised deep keyphrase generation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Virtual Event, February 22 - March 1, 2022*, pages 11303–11311. AAAI Press.

Tomer Simon, Avishay Goldberg, and Bruria Adini. 2015. Socializing in emergencies—a review of the use of social media in emergency situations. *International Journal of Information Management*, 35(5):609–619.

Richard Socher and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.

Ge Song, Yunming Ye, Xiaolin Du, Xiaohui Huang, and Shifu Bie. 2014. Short text classification: A survey. *Journal of multimedia*, 9(5):635.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 841–842. ACM.

Kevin Stowe, Jennings Anderson, Martha Palmer, Leysia Palen, and Ken Anderson. 2018. Improving classification of Twitter behavior during hurricane events. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 67–75, Melbourne, Australia. Association for Computational Linguistics.

Kevin Stowe, Michael J. Paul, Martha Palmer, Leysia Palen, and Kenneth Anderson. 2016. Identifying and categorizing disaster-related tweets. In *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media*, pages 1–6, Austin, TX, USA. Association for Computational Linguistics.

Stoyan Stoyanov. 2017. Crisis and disaster management terminology. In *Implications of Climate Change and Disasters on Military Activities*, pages 3–10. Springer.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Kanae Takahashi, Kouji Yamamoto, Aya Kuchiba, and Tatsuki Koyama. 2022. Confidence interval for micro-averaged f1 and macro-averaged f1 scores. *Applied Intelligence*, 52(5):4961–4972.

Hien To, Sumeet Agrawal, Seon Ho Kim, and Cyrus Shahabi. 2017. On identifying disaster-related tweets: Matching-based or learning-based? In *Third IEEE International Conference on Multimedia Big Data, BigMM 2017, Laguna Hills, CA, USA, April 19-21, 2017*, pages 330–337. IEEE Computer Society.

Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. 2014. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 1799–1807.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks.

Sarah Vieweg, Amanda Lee Hughes, Kate Starbird, and Leysia Palen. 2010. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*, pages 1079–1088. ACM.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. pages 7371–7379.

Carlos Villegas, Matthew Martinez, and Matthew Krause. 2018. Lessons from harvey: Crisis informatics for urban resilience.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pages 855–860. AAAI Press.

Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. 2017. Combining knowledge with deep convolutional neural networks for short text classification. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2915–2921.

Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. 2019. Topic-aware neural keyphrase generation for social media language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2516–2526, Florence, Italy. Association for Computational Linguistics.

Tingyu Xia, Yue Wang, Yuan Tian, and Yi Chang. 2021. Using prior knowledge to guide bert's attention in semantic textual matching tasks. In *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, pages 2466–2475. ACM / IW3C2.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with bertserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 72–77. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489. The Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7370–7377. AAAI Press.

Hai Ye and Lu Wang. 2018. Semi-supervised learning for neural keyphrase generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4142–4153. Association for Computational Linguistics.

Jiacheng Ye, Ruijian Cai, Tao Gui, and Qi Zhang. 2021. Heterogeneous graph neural networks for keyphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2705–2715. Association for Computational Linguistics.

Joosung Yoon and Hyeoncheol Kim. 2017. Multi-channel lexicon integrated CNN-BiLSTM models for sentiment analysis. In *Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017)*, pages 244–253, Taipei, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Manzhu Yu, Chaowei Yang, and Yun Li. 2018. Big data in natural disaster management: a review. *Geosciences*, 8(5):165.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.*, 31(7):1235–1270.

Himanshu Zade, Kushal Shah, Vaibhavi Rangarajan, Priyanka Kshirsagar, Muhammad Imran, and Kate Starbird. 2018. From situational awareness to actionability: Towards improving the utility of social media data for crisis response. *PACMHCI*, 2(CSCW):195:1–195:18.

Hamada M. Zahera, Ibrahim A. Elgendy, Rricha Jalota, and Mohamed Ahmed Sherif. 2019a. Fine-tuned BERT model for multi-label tweets classification. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019*, volume 1250. National Institute of Standards and Technology (NIST).

Hamada M Zahera, Rricha Jalota, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. 2021. I-aid: Identifying actionable information from disaster-related tweets. *IEEE Access*, 9:118861–118870.

Hamada M Zahera, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. 2019b. Jointly learning from social media and environmental data for typhoon intensity prediction. In *Proceedings of the 10th International Conference on Knowledge Capture*, pages 231–234. ACM.

Hamada M. Zahera, Daniel Vollmers, Mohamed Ahmed Sherif, and Axel-Cyrille Ngonga Ngomo. 2022. Multpax: Keyphrase extraction using language models and knowledge graphs. In *The Semantic Web - ISWC 2022 - 21st International Semantic Web Conference, Virtual Event, October 23-27, 2022, Proceedings*, volume 13489 of *Lecture Notes in Computer Science*, pages 303–318. Springer.

Torsten Zesch and Iryna Gurevych. 2009. Approximate matching for evaluating keyphrase extraction. In *Recent Advances in Natural Language Processing, RANLP 2009, 14-16 September, 2009, Borovets, Bulgaria*, pages 484–489. RANLP 2009 Organising Committee / ACL.

Jiaofu Zhang, Lianzhong Liu, Zihang Huang, Lihua Han, Shuhai Wang, Tongge Xu, Jingyi Zhang, Yangyang Li, Yifeng Liu, and Md Zakirul Alam Bhuiyan. 2021. Robust social event detection via deep clustering. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 814–819. IEEE.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuan-Jing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 836–845.

Yin Zhang, Rong Jin, and Zhi-Hua Zhou. 2010. Understanding bag-of-words model: a statistical framework. *International journal of machine learning and cybernetics*, 1(1):43–52.

Jing Zhao, Junwei Bao, Yifan Wang, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2021a. SGG: learning to select, guide, and generate for keyphrase generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5717–5726. Association for Computational Linguistics.

Yu Zhao, Jia Song, Huali Feng, Fuzhen Zhuang, Qing Li, Xiaojie Wang, and Ji Liu. 2021b. Deep keyphrase completion. *CoRR*, abs/2111.01910.

Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 425–434. ACM.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.