INTERPRETABLE CLASSIFICATION AND SUMMARIZATION OF CRISIS EVENTS FROM MICROBLOGS

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

DOKTOR DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

M.Sc. Thi Huyen Nguyen

geboren am 14. Dezember 1991 in Ninh Binh, Vietnam

2024

Referent: Prof. Dr. Wolfgang Nejdl
Korreferent: Prof. Dr. Prasenjit Mitra
Korreferent: Prof. Dr. Henning Wachsmuth
Tag der Promotion: 19.06.2024

ABSTRACT

The widespread use of social media platforms has created convenient ways to obtain and spread up-to-date information during crisis events such as disasters. Time-critical analysis of crisis-related information helps humanitarian organizations and governmental bodies gain actionable information and plan for aid response. However, situational information is often immersed in a high volume of irrelevant content. Moreover, crisis-related messages also vary greatly in terms of information types, ranging from general situational awareness - such as information about warnings, infrastructure damages, and casualties - to individual needs. Different humanitarian organizations or governmental bodies usually demand information of different types for various tasks such as crisis preparation, resource planning, and aid response. To cope with information overload and efficiently support stakeholders in crisis situations, it is necessary to (a) classify data posted during crisis events into fine-grained humanitarian categories, (b) summarize the situational data in near real-time.

In this thesis, we tackle the aforementioned problems and propose novel methods for the classification and summarization of user-generated posts from microblogs. Previous studies have introduced various machine learning techniques to assist humanitarian or governmental bodies, but they primarily focused on model performance. Unlike those works, we develop interpretable machine-learning models which can provide explanations of model decisions. Generally, we focus on three methods for reducing information overload in crisis situations: (i) post classification, (ii) post summarization, (iii) interpretable models for post classification and summarization. We evaluate our methods using posts from the microblogging platform Twitter, so-called tweets. First, we expand publicly available labeled datasets with rationale annotations. Each tweet is annotated with a class label and rationales, which are short snippets from the tweet to explain its assigned label. Using the data, we develop trustworthy classification methods that give the best tradeoff between model performance and interoperability. Rationale snippets usually convey essential information in the tweets. Hence, we propose an integer linear programming-based summarization method that maximizes the coverage of rationale phrases to generate summaries of class-level tweet data. Next, we introduce an approach that can enhance latent embedding representations of tweets in vector space. Our approach helps improve the classification performance-interpretability tradeoff and detect near duplicates for designing a summarization model with low computational complexity. Experiments show that rationale labels are helpful for developing interpretable-by-design models. However, annotations are not always available, especially in real-time situations for new tasks and crisis events. In the last part of the thesis, we propose a two-stage approach to extract the rationales under minimal human supervision.

Keywords: *classification, summarization, interpretability, multi-task learning, semi-supervised learning, crisis events, Twitter*

ZUSAMMENFASSUNG

Die weit verbreitete Nutzung von Social-Media-Plattformen hat vielfältige Möglichkeiten geschaffen, um in Krisensituationen wie z.B. bei Katastrophen aktuelle Informationen zu erhalten und zu verbreiten. Die zeitnahe Analyse krisenbezogener Informationen hilft humanitären Organisationen und weiteren Akteuren dabei, aktuelle und verwertbare Informationen zu erhalten und Hilfsmaßnahmen zu planen. Allerdings sind solche situationsbezogenen Informationen in der Regel in einer großen Menge irrelevanter Inhalte verborgen. Darüber hinaus gibt es auch sehr unterschiedliche Arten von krisenbezogenen Nachrichten. Diese reichen von allgemeinerSituationswahrnehmung - wie Warnungen, Informationen zu Infrastrukturschäden und Opfern - bis hin zu Informationen zu individuellen Bedarfen. Für unterschiedliche Aufgaben der einzelnen Akteure werden dabei unterschiedliche Arten von Informationen benötigt. Um die Datenflut zu bewältigen und die Beteiligten in Krisensituationen effizient zu unterstützen, ist es notwendig, (a) Daten, die in Krisensituationen veröffentlicht werden, in feingranualare humanitäre Kategorien zu klassifizieren, (b) die Situationsdaten in Echtzeit geeignet zusammenzufassen.

In dieser Arbeit befassen wir uns mit den oben genannten Herausforderungen und schlagen innovative Methoden für die Klassifizierung und Zusammenfassung von nutzergenerierten Inhalten insbesondere von Posts in Microblogging-Plattformen insbesondere im Kontext von Krisensituationen vor. In früheren Studien wurden verschiedene Techniken des maschinellen Lernens zur Unterstützung von humanitären oder Regierungsbehörden zu unterstützen, aber sie konzentrierten sich hauptsächlich auf die Modellleistung. Im Gegensatz zu diesen Arbeiten entwickeln wir interpretierbare Machine-Learning-Modelle, die Erklärungen für Modellentscheidungen liefern können. Dabei konzentrieren wir uns auf drei Hauptaspekte: (i) Klassifizierung von Posts, (ii) Zusammenfassung von Posts, (iii) interpretierbare Modelle für die Nachklassifizierung und Zusammenfassung. Die entwickelten Methoden werden mit Daten der Plattform Twitter sogenannten Tweets evaluiert. Zunächst erweitern wir öffentlich verfügbare, gelabelte Datensätze mit begründenden Annotationen. Jeder Tweet wird mit einem Klassenlabel und Begründungen (Rationales) annotiert, das sind kurze Ausschnitte aus dem Tweet, um das zugewiesene Label zu erklären. Anhand dieser Daten entwickeln wir vertrauenswürdige Klassifizierungsmodelle, die den besten Kompromiss zwischen Modelleffektivität und Interpretierbarkeit erzielen. Rationales vermitteln in der Regel wichtige Informationen in den Tweets. Daher schlagen wir einen auf ganzzahliger linearer Programmierung basierenden Zusammenfassungsansatz vor, der die Abdeckung der Begründungsphrasen maximiert, um Zusammenfassungen von Tweet-Daten auf Klassenebene zu generieren. Weiterhin schlagen wir einen Ansatz vor, der die latente Einbettung von Tweets im Vektorraum verbessern kann. Unser Ansatz hilft, den Kompromiss zwischen Klassifizierungseffektivität und Interpretierbarkeit zu verbessern und Fast-Duplikate zu vermeiden, um ein Zusammenfassungsmodell mit geringer Rechenkomplexität zu erhalten. Experimente zeigen, dass Die rationale

Annotationen für die Entwicklung interpretierbarer Modelle hilfreich sind. Annotationen sind jedoch nicht immer verfügbar, insbesondere in Echtzeitsituationen, für neue Aufgaben und in Krisensituationen. Im letzten Teil der Arbeit schlagen wir daher einen zweistufigen Ansatz vor, um die Rationales mit minimalem menschlichen Aufwand zu erlernen.

Schlagwörter: Klassifizierung, Zusammenfassung, Interpretierbarkeit, Multitasking Lernen, Halb-überwachtes Lernen, Krisensituationen, Twitter

ACKNOWLEDGMENTS

I would like to acknowledge everyone who has supported me throughout the course of my doctoral studies. First of all, I would like to thank Prof. Dr. techn. Wolfgang Nejdl for giving me the opportunity to work in a great working environment, together with many excellent scientists at the L3S Research Center. He has supervised my path as a PhD student from the first day and made it possible for me to write up this thesis.

I would like to express my sincere gratitude to Dr. Claudia Niederée for her time and invaluable suggestions. She has helped me overcome challenges and explore new collaborations during my time as a PhD student. Special thanks to Dr. Koustav Rudra and Dr. Tuan-Anh Hoang for their advice, guidance, and inspiring discussions, which have shaped my skills in doing research. Thanks to their huge contribution, I have successfully conducted the works published in this thesis. I am also grateful to Dr. Marco Fisichella, Tai Le Quy, and Huu Hoang Nguyen for the fruitful discussions and collaborations. Without a doubt, thanks should also go to my officemates and colleagues. They have shared many great experiences, joyful conversations, and memorable moments with me.

Finally, I would be remiss in not mentioning my family. I am indebted to all my family members for their unconditional love and encouragement throughout my life. They have accompanied my path and been a constant source of motivation to me. This thesis would not have been possible without their support.

FOREWORD

The algorithms and approaches presented in this thesis have been published at various conferences, as follows:

Chapter 3 describes the following works on filtering and summarizing event-related tweets of Breaking News in Twitter:

- Tuan-Anh Hoang, Thi-Huyen Nguyen, and Wolfgang Nejdl. Efficient Tracking of Breaking News in Twitter. In *Proceedings of the 11th ACM Conference on Web Science (WebSci), Boston, MA, USA*, July 2019 [52]

- Thi Huyen Nguyen, Tuan-Anh Hoang, Wolfgang Nejdl, Efficient Summarizing of Evolving Events from Twitter Streams, in *Proceedings of the 2019 SIAM International Conference on Data Mining*, Calgary, Alberta, Canada, May 2019. [103]

Chapter 4 builds up on the following publications about an interpretable approach to classify and summarize crisis-related tweets from Twitter:

- Thi Huyen Nguyen, Koustav Rudra, Towards an Interpretable Approach to Classify and Summarize Crisis Events from Microblogs, in *Proceedings of the ACM Web Conference 2022*, April 2022. [108]

Chapter 5 describes a contrastive learning-based approach to improve the interpretability and robustness of crisis-related classification and summarization models. The chapter presents the research published in:

- Thi Huyen Nguyen, Koustav Rudra, Rationale Aware Contrastive Learning Based Approach to Classify and Summarize Crisis-Related Microblogs, in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, October 2022. [107]

Chapter 6 focuses on a semi-supervised approach to learn faithful attention-based explanations for the classification of crisis events and builds up on the following work:

- Thi Huyen Nguyen, Koustav Rudra, Learning Faithful Attention for Interpretable Classification of Crisis-Related Microblogs under Constrained Human Budget, in *Proceedings of the ACM Web Conference 2023*, May 2023. [109]

The complete list of publications during my doctoral studies is as follows:

Conference papers

- Thi Huyen Nguyen, Tuan-Anh Hoang, Wolfgang Nejdl, Efficient Summarizing of Evolving Events from Twitter Streams, in *Proceedings of the 2019 SIAM International Conference on Data Mining*, Calgary, Alberta, Canada, May 2019. [103]
- Thi Huyen Nguyen, Hoang H. Nguyen, Zahra Ahmadi, Tuan-Anh Hoang, Thanh-Nam Doan, in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (WI-IAT), Melbourne, Australia, December 2021. [104]
- Thi Huyen Nguyen, Koustav Rudra, Towards an Interpretable Approach to Classify and Summarize Crisis Events from Microblogs, in *Proceedings of the ACM Web Conference 2022*, Lyon, France, April 2022. [108]
- Thi Huyen Nguyen, Koustav Rudra, Rationale Aware Contrastive Learning Based Approach to Classify and Summarize Crisis-Related Microblogs, in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management (CIKM)*, Atlanta, Georgia, USA, October 2022. [107]
- Thi Huyen Nguyen, Koustav Rudra, Learning Faithful Attention for Interpretable Classification of Crisis-Related Microblogs under Constrained Human Budget, in *Proceedings of the ACM Web Conference 2023*, Austin, Texas, USA, May 2023. [109]
- Thi Huyen Nguyen, Marco Fisichella, and Koustav Rudra, A Trustworthy Approach to Classify and Analyze Epidemic-Related Information From Microblogs, *IEEE Transactions on Computational Social Systems*, May 2024. [102]
- Thi Huyen Nguyen, Koustav Rudra, Human vs ChatGPT: Effect of Data Annotation in Interpretable Crisis-Related Microblog Classification, in *Proceedings of the ACM on Web Conference 2024*, Singapore, May 2024. [110]

Demo and Poster papers

- Tuan-Anh Hoang, Thi-Huyen Nguyen, and Wolfgang Nejdl. Efficient Tracking of Breaking News in Twitter. In *Proceedings of the 11th ACM Conference on Web Science (WebSci), Boston, MA, USA*, July 2019. [52]

- Thi Huyen Nguyen, Miroslav Shaltev, and Koustav Rudra. CrisICSum: Interpretable Classification and Summarization Platform for Crisis Events from Microblogs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA*, October 2022. [111]

Workshop papers

- Tai Le Quy, Thi Huyen Nguyen, Gunnar Friege, and Eirini Ntoutsi. Evaluation of Group Fairness Measures in Student Performance Prediction Problems. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases - International Workshops of ECML PKDD*, Grenoble, France, September 2022. [70]

- Thi Huyen Nguyen, Koustav Rudra, L3S at TREC 2022 CrisisFACTs track, in *Proceedings of the 31st Text REtrieval Conference (TREC)*, 2022. [106]

ArXiv and under-review papers

- Thi Huyen Nguyen, Tu Nguyen, Tuan-Anh Hoang, Claudia Niederée, On the Feasibility of Predicting Questions being Forgotten in Stack Overflow, arXiv, 2021, https://arxiv.org/abs/2110.15789 [105]

- Thi Huyen Nguyen, Koustav Rudra, Wolfgang Nejdl, Cross-Modality Rationale Learning for Interpretable Classification of Crisis-Related Microblogs.

List of Figures

List of Tables

Contents

1 Introduction

1.1 Motivation

The recent growth in popularity of microblogging platforms and other social media platforms has led to a major shift in global communication. Thanks to social media, information spreads faster, and the world seems to get smaller. People can easily get updated with the latest news about the world on a daily basis. Social media offers a fast indicator of currently occurring and developing events, so-called breaking news events. During these events, a large volume of real-time information is posted. Acquiring posts related to breaking news in real-time is vital in many important applications, such as event detection, trend analysis, social sensing, and public opinion monitoring. However, real-time acquisition of news-relevant posts is challenging due to the massive volume of irrelevant content, the prevalence of noise, and the wide range of topics. Besides, long-ranging breaking news often attracts a high number of relevant posts, making it impossible to understand the events by reading through all the posts. It is, therefore, necessary to develop automatic systems for filtering and summarizing news-relevant content of these events. These systems should be interpretable so that end-users and decision-makers can deploy them for their purposes.

Recently, microblogging platforms have been heavily leveraged to report and exchange information about natural disasters. Many previous studies have shown the vital role of user-generated posts from microblogging platforms in enhancing emergency situational awareness and planning aids during crisis situations [57]. However, situational data vary greatly in terms of information types, such as infrastructure damage, caution and advice, affected individuals, rescue, or irrelevant content [55]. These information types are defined and used by United Nations Office for the Coordination of Humanitarian Affairs (UN OCHA). The data of different information types are utilized for different purposes. For example, information about "caution and advice" is crucial for both humanitarian organizations and local people to obtain situational awareness and preparation. Meanwhile, posts about infrastructure

damage are beneficial for governmental bodies, NGOs, and rescue agencies to assess the situation's seriousness and for aid planning. The availability of information from social media sources can help decision-making tasks of humanitarian organizations and governmental bodies be easier. Nevertheless, a huge volume of information also becomes a bottleneck; hence, a streamlined mode of updating across different categories is desired by the agencies. It thus requires efficient methods to classify posts into fine-grained humanitarian classes and then summarize class-level posts in near real time.

Apart from high performance, interpretability has become an essential component in model building, which is decisive in the applicability of machine learning models in real-life scenarios. Tasks that have an impact on society and human lives are crucial, and explainability is an important criterion when designing machine learning models in such cases. Despite advances in Natural Language Processing [35] and interpretable Deep Learning models [36, 62] on formal text datasets, previous works on classification and summarization of crisis events primarily focus on performance but did not pay attention to the decision-making processes. Such systems need to be interpretable in nature [124, 125, 127] to increase the trust of end-users for application purposes. Hence, interpretable approaches that obtain the best trade-off between model performance and interpretability need to be developed for tasks in crisis domain. The European Union General Data Protection Regulation (GDPR) also extended the automated decision-making rights and regulated 'right to explanation' to double or legally related decision [168].

Among microblogging platforms, Twitter is emerging as one of the most popular network, with over 330 million monthly active users and more than 500 million tweets posted per day [169]. The platform allows users to express their own opinions in the form of short messages, known as tweets, on various topics ranging from business and politics to casual conversation. Twitter is fast in delivering news ,"beating traditional media", by providing real-time and eye-witnessed content. Some surveys and large-scale analyses have shown that a major proportion of Twitter users frequently tweet and receive information about news [13, 15]. Empirical studies also find that Twitter often has faster responses and more complete coverage than mass media when reporting breaking news [159]. In turn, Twitter is increasingly viewed as a crucial resource for detecting and monitoring breaking news events, especially during crisis situations such as natural disasters [56].

1.2 Scope of the thesis

Our primary goal is to develop efficient and trustworthy methods to assist humanitarian organizations in their decision-making processes, mostly during natural disaster events such as earthquakes, typhoons, etc. Due to the availability of Twitter API [152], it has become common to use the platform as a resource for many re-

search tasks. In this thesis, we address three core problems: (i) Identification and summarization of tweets related to breaking events from Twitter, (ii) Developing interpretable classification and summarization approaches for the crisis-related tweets, and (iii) Adapting the developed models to work under limited human supervisions. To achieve our overall goal, we lay out the following research questions:

Question 1: *Given a large tweet stream of an evolving breaking news event, how to efficiently identify and summarize relevant tweets in near real-time?*

At the onset of breaking news, especially during emergency events, Twitter receives an overwhelming amount of messages. As an example, approximately one million tweets containing some COVID-19-related hashtags were posted daily during the first two weeks of the COVID-19 outbreak in 2020. Then, the event evolved and attracted more than 6 million tweets daily for a long time [120]. Given a specific breaking news event, providing a dataset of relevant tweets is helpful for many further purposes, such as identifying specific needs or estimating damages during crises. However, event-related tweets are normally immersed in a high volume of irrelevant messages. Therefore, automatic and efficient tools for the automatic filtering of relevant tweets are demanded. Supervised learning approaches require human annotations for training. Besides, these methods are often not scalable in case of evolving events as they require retraining processes to adapt the filters periodically. Some common keyword-based approaches involve manual effort over time to update new keywords as events evolve. Our objective is to propose a method that requires minimum initial human effort and is able to efficiently filter relevant tweets of evolving events over time.

After the filtering step, there is still a vast number of tweets about an event. It is impossible for humans to digest the information by reading through all the tweets. To obtain a quick overview of the situation, an online summary of event-related tweets is valuable. This helps to handle the content overload and provides an overall understanding of the situation. Given a breaking news event, tweets can contain information on diverse sub-events. Some common approaches to capturing events' aspects are topic models [32], burst detection [43], or clustering [78]. However, as events evolve, the number of tweets is skewed toward large sub-events, and these methods often fail at covering less popular sub-events. Besides, clustering-based methods usually require the number of clusters or other hyper-parameters in advance, which is not practical in case of evolving events. Most previous studies focus on the one-time summarization of static or small-scale data. These methods are not scalable with large-scale Twitter streams since these models can not be updated incrementally with new data from evolving events. A few studies have made some first attempts to propose methods for online summarization of Twitter Streams [113, 140, 163], but their models still have high computational costs. In this thesis, we focus on a method that is efficient, scalable, and able to generate informative and diverse summaries of evolving events.

Question 2: *How to design an interpretable model for classification and summa-*

rization of crisis events?

During crisis events such as earthquakes, typhoons, etc., relevant messages are drenched in a huge mass of irrelevant posts. Human organizations may want to obtain concise summaries of tweets at fine-grained levels, such as "caution and advice", "infrastructure damage", "volunteering and rescue", "injuries and death", etc. To fulfill this, many studies have provided datasets with humanitarian class labels and different approaches for the classification and summarization of tweets. However, existing crisis-specific classification and summarization models mainly focus on model performance but not model transparency [55, 129, 98, 130]. Advanced deep-learning models [35, 121] perform better than traditional machine-learning methods on many tasks, yet it is quite opaque how they come to make output decisions. Recently, interpretability has arisen as an important topic. Models are interpretable when humans can understand the cause/reasoning of an output decision. Interpretability is important for researchers and developers to debug machine learning models and make informed decisions on how to improve them. Moreover, for tasks that have high impacts on society, such as in health or crisis domains, interpretability makes sure that a proposed model is right for the right reasons, increases the trust and confidence of end-users to use machine-based supporting systems or applications. Hence, we bring the trade-off between model accuracy and interpretability in designing the classification and summarization models of crisis-related tweets to the forefront.

Question 3: *How to learn better representations of crisis-related tweets in vector space for improvement of interpretable classification and summarization models?*

Generally, the performance and applicability of classification-summarization systems depend on two factors — (i) the representation of posts in latent embedding space and (ii) understanding the decision-making process of the model. While the first factor helps in boosting the performance, the second one ensures the interpretability of the model, which, in turn, helps in the adaptation of such systems in real-life usage. Some pre-trained language models such as BERT [35], BERTweet [99], RoBERTa [82] performs well on many downstream tasks. Nevertheless, the pre-trained embedding representations are unsuitable for unsupervised tasks such as clustering, similarity detection of tweets, etc. Generally, the cosine similarities between BERT-based embedding representations of any two tweets are skewed toward 1.0. In this thesis, we propose an approach to advance embedding representations of crisis-related tweets so that they can be further used to improve tweet classification and summarization performance. Also, we consider interpretability as a primary objective when designing the classification model of tweets during crisis events.

Question 4: *How to learn faithful attention-based explanations with minimal human supervision?*

Over the last few years, various approaches have been proposed in an attempt to open the black box of deep networks [79, 62]. Neural network models often use attention mechanisms [9] to improve the performance of various tasks in natural language processing. Attention produces a probability distribution over input tokens, and the

vector representation of the entire input sentence is the weighted sum of its constituent token vectors. Many works have used attention weights as explanations for model predictions, where tokens with higher attention weights are considered more important for the output decision. Nevertheless, recent studies illustrate that attention weights do not always provide faithful explanations [59, 165]. Other works [36, 174] proposed in-modeling interpretable approaches that make the model inherently interpretable. However, the authors relied on manual human annotations to train interpretable models and predict tokens responsible for determining the output labels. This approach is promising for designing interpretable crisis-related systems, but the human annotation also adds a bottleneck to the scalability of the method and its applications to unseen crisis events. This motivates us to solve the following question in designing classification models for crisis events: Can we learn faithful attention-based explanations for interpretable classification of crisis events under a given human budget, i.e., limited human annotation of explanations?

1.3 Contributions

This thesis addresses the identified research questions in the previous section. Overall, we focus on four principal contributions in the field of crisis-related classification and summarization. As the first contribution in this thesis, we propose approaches for online filtering out and summarizing relevant tweets of an evolving breaking news event, which can be natural disasters, man-made crises, or other popular breaking news. In the second part, we focus on crisis events only and introduce an interpretable classification and summarization framework for crisis-related tweets. The third contribution is to learn effective embedding representations of crisis-related tweets for a better classification (i.e., higher accuracy and interpretability scores) and a more robust summarization of tweets during crisis events by using a contrastive learning-based approach. Finally, we propose an attention-based interpretable model for the classification of crisis events under limited human guidance.

(I). Classification and summarization of evolving breaking news events. We present an efficient semi-supervised graph-based method for filtering tweets relevant to a given breaking news from Twitter's stream. The task is studied in the context where only a small set of news-related tweets is given at the early stage of the breaking news, and we have to decide, in real-time, if subsequent tweets in the tweet stream are relevant to the news. The main idea of our proposed approach is to first represent relevant tweets to a given breaking news by a graph whose nodes and edges are defined by words and their co-occurrence in the tweets, respectively. Besides, we also maintain another graph of background tweets that occur around the same time as the tracked news. Then, we introduce a method to measure the relevance score of each incoming tweet to the news for determining the tweet label. Defined in this way, the graphs can be updated incrementally. Our method only requires minimal human supervision. It performs better in filtering incoming relevant tweets and is

computationally more efficient than other baselines.

For summarization, we propose a graph-based method for extractive summarization of tweet streams. Our method employs a word graph to represent tweets from the stream. The graph allows us to update the representation in real time. To perform the summarization at a time point, we first apply an incremental algorithm inspired by a diversified ranking approach [67] to detect sub-events. This algorithm is totally unsupervised and able to select a diverse set of words representing sub-events. Lastly, most representative tweets are carefully chosen from a small set of candidates containing those selected words and returned as the summary. Our method, therefore, does not require prior information and is highly scalable while returning more informative, diverse, and readable summaries.

(II). Interpretable classification and summarization of crisis events. Following the ideas from some previous works for tasks on normal text datasets [36, 174], we introduce an interpretable-by-design multi-task model to classify tweets into fine-grained humanitarian classes during crisis events. Our classification model can provide explanations or rationales [1] for its decisions. Rationales are short snippets from the original text that provide supporting evidence for the output label. For example, the tweet "*RT @USER: Nearly 1,805 dead in Nepalś killer quake, India mounts massive rescue operation*" reports information about death, and "*Nearly 1,805 dead*" is annotated as a rationale, which captures essential information to classify the tweet into "injures and death" class. In the summarization phase, we employ an Integer Linear Programming (ILP) based optimization technique along with the help of rationales to generate summaries of event categories.

Our classification model obtains the best trade-off between model performance and interpretability compared to existing methods. Besides, the summarization model benefits from rationale data. Our generated summaries are informative regarding both groundtruth-based and human evaluations.

(III). Learning efficient tweet representation for interpretable classification and summarization of crisis events. To obtain good representations of crisis-related tweets and further boost the performance of classification and summarization tasks, we propose a rationale-aware contrastive learning-based classification and summarization framework. Our proposed Rationale Aware Contrastive Learning based Classification (RACLC) model consists of two learning stages. In the first stage, the model learns rationales by jointly optimizing three loss values, i.e., losses of class label prediction, rationale extraction task, and an additional contrastive loss [65], which learns to bring semantically similar tweets closer and dissimilar tweets far apart. In the second stage, we feed the extracted rationales from the first stage to a simple BERTweet [99] model with a Softmax output layer on top to classify tweets into humanitarian classes. This step shows the interpretability of the predicted rationales. Next, we propose an integer linear programming-based summarization approach that maximizes the coverage of rationale words and minimizes redundancy

[1] rationales and explanations are used interchangeably in this thesis

by discarding duplicate or near-duplicate tweets. Contrastive learning-based latent representations of tweets help in the detection of near-duplicate tweets. Thus, our Contrastive Learning-based Tweet Summarization (RACLTS) model can generate informative summaries with low computational complexity. Besides, we evaluate and show a promising application of our classifier in extracting actionable snippets from a subset of actionable tweets provided by TREC-IS [90]. Actionable tweets generally contain immediate and critical alerts useful for crisis response. For example, the tweet "*Just received an email from Paris. Some* ***french people missing in #Langtangvalley*** *#NepalEarthquake . Any info!! http://t.co/fUbBBCKRhY*" and the bold texts are the actionable tweet and actionable snippet, respectively.

(IV). Learning faithful attention-based explanations for the classification of crisis events under a constrained human budget. In previous parts of the thesis, we consider interpretable-by-design classification models that require human annotations of rationales to train and extract explanations for the output prediction. In this part, we focus on the following two contributions: (i). we introduce a two-stage framework that exploits the power of a semi-supervised learning approach to learn faithful explanations under a limited amount of human-annotated data. The first stage is to train and predict rationale snippets under a semi-supervised setup. The second stage predicts the class label based on the extracted rationales only. (ii). We try to align the attention weights of tokens with the predicted probabilities of rationales to make the attention weights faithful. Our model obtains better or comparable classification performance to baselines and faithful attention heatmaps using only 40-50% human-level supervision. Furthermore, we also explore the application of our classifier in identifying actionable labels and actionable snippets from the dataset provided by TREC-IS [91] in the *transfer-learning* setup.

1.4 Thesis Structure

The remainder of the thesis is organized as follows.

In Chapter 2, we review the literature for the thesis. In particular, we focus on selected theories and techniques for three main problems: Tweet Classification, Tweet Summarization, and Model Interpretability.

Chapter 3 discusses our proposed approach of filtering and summarizing tweets relevant to a given breaking news event from large Twitter streams. We consider graph-based semi-supervised and unsupervised approaches that can efficiently classify relevant tweets and summarize evolving events in near real-time.

Chapter 4 focuses on interpretable classification and summarization approaches in the crisis domain. We introduce a multi-task model that can simultaneously classify tweets into fine-grained humanitarian classes and provide explanations for the output decisions. The outputs of the classification phase are then used for the summarization of class-level tweets during crisis events.

Chapter 5 presents our proposed rationale-aware contrastive learning approach to classify and summarize tweets during crisis events. The contrastive learning framework helps improve the interpretability of the classification model and results in better representations of tweets in the embedding space. We make use of the fine-tuned embeddings and design a summarization model that performs equally well or better than existing methods with low computational cost.

Chapter 6 describes our work on improving the faithfulness of attention-based explanations for the classification of crisis events under a constrained human budget. In particular, we apply a semi-supervised approach to derive faithful machine attention for the model's decisions. Further, we employ a zero-shot transfer learning setup in the identification of rationales in actionable tweets, which contain crucial information and immediate alert during crisis events.

Finally, we conclude our contributions of this thesis and discuss future research direction in Chapter 7.

2 Literature Review

This chapter presents an overview of essential backgrounds and recent studies that are related to our works in this thesis. First, a short description of the Twitter platform and its practical usage is presented. Next, research works on two important tweet classification problems, which are the identification of relevant tweets during generic breaking news events and tweet classification during crisis events, are discussed. Then, I review works on text summarization and tweet summarization. Finally, I present an overview of machine learning interpretability and representation learning methods.

2.1 Twitter as Data Source

The microblogging platform Twitter was launched in 2006 as a social networking service for users to send and respond to texts, images, and videos, known as "tweets". Since its onset, Twitter has experienced rapid growth. As of 2022, the platform has more than 350 million users and 500 million tweets posted per day [7]. Many studies have shown the crucial impacts of Twitter in many aspects, such as communication, education, politics, or emergency management [173, 172, 24]. Twitter's usage spikes during prominent breaking news. For example, millions of tweets posted daily during the recent COVID-19 outbreak [58]. The real-time functionality makes the site an effective *de facto* emergency communication channel for breaking news. Twitter provides us with an API (application programming interface) [152] that we can use to easily retrieve tweet data. The data availability and easy accessibility make the platform a crucial resource for the research community. Sakaki et al. [134] suggested Twitter as a real-time sensor for the detection of natural disasters such as earthquakes. The authors proposed models to discover the center and trajectory of earthquakes' location. A recent work [145] showed that crowdsourced detection of seismic activity from Twitter provides reliable locations of earthquakes, which are in many cases faster than seismological protocols. Some studies provided surveys of research works on processing social media messages in mass emergencies [54].

2.2 Tweet Filtering and Classification

Identification of event-relevant tweets is useful for users and stakeholders to easily keep track of situational updates for their own purposes. However, at the time of a specific event, many irrelevant messages are posted along with useful and relevant information. This section discusses some recent studies on tweet classification to prioritize important information during generic and crisis-related breaking news events.

2.2.1 Tweet classification during evolving breaking news

A series of studies have tried to identify relevant tweets during evolving events. Prior works on the online extraction of relevant tweets during an evolving generic breaking news can be grouped by main approaches as follows:

Keyword based approaches: These methods focus on defining and expanding the set of keywords that capture essential aspects of the event for tweet retrieval. A tweet is considered relevant to an event if it contains one or more event-related keywords. Wang et al. [161] proposed to use hashtags as keywords for filtering tweets related to some evolving event. Starting from an initial set of manually selected hashtags, the set is expanded by adding new hashtags most similar to the ones in the set. Similarly, Li et al. [72] proposed a greedy method for automatically selecting keywords by estimating the tweet usefulness on recent samples from the Twitter stream. The proposed method is, however, not incremental. Cotelo et al. [33] proposed to select the keywords using a graph-based method that can be updated incrementally.

Information retrieval based approaches. These approaches retrieve relevant tweets to query topics based on information retrieval strategies, where tweets and queries are transformed into suitable representations (i.e., vectors, matrices, tuples, etc.) for retrieval. Different retrieval strategies employ different models for their text representation purposes. Lin et al. [76] examined different smoothing techniques for language model-based tweet filtering. They, however, only considered broad topics such as 'baseball' and 'fashion' with many past relevant tweets for training. Later, TREC started its *microblog track* for real-time searching and filtering in Twitter [143]. The track has attracted a number of contributions based on adapting information retrieval models for tweet filtering [6, 71]. All approaches rely on extensive external information sources and/or require manual decisions on some threshold settings. Albakour et al. [4] proposed to use pseudo-relevance feedback techniques for enriching tweets instead of external sources.

Supervised learning based approaches. These approaches employ labeled data to train machine learning models and then determine relevant tweets for a given topic/event. There are proposed supervised models within TREC's microblog track for tweet filtering [148, 14]. However, they heavily rely on external information sources. Later contributions suggested different methods for engineering features solely based on tweets' content [177, 88].

Unlike the above approaches, we propose the semi-supervised graph-based ranking approach for measuring terms' importance in a document as proposed in [19, 126] to filter relevant tweets of evolving breaking news in real time. Our method is inspired by the idea of graph-based models for summarization [136, 93]. Previous works, however, do not consider the importance of edges connecting terms when measuring documents' similarities as we do. We further devise an intuitive strategy for automatically setting decision thresholds that neither require domain knowledge nor prior information.

2.2.2 Tweet classification during crisis events.

During mass convergence breaking news, such as natural disasters, tweets can contain information about different humanitarian classes, such as infrastructure damage, caution, rescue, etc. Many previous works have attempted to provide datasets with humanitarian labels for efficient extraction of tweets into fine-grained categories [56, 2, 91]. Some popular humanitarian classes that are defined by UN OCHA and considered in many existing works and also in our thesis are:

- Infrastructure damage: Reports of damaged roads, bridges, buildings, monuments, interrupted or restored services and utilities. This information is helpful for the severity assessment of the damage.
- Caution and advice: Information about warnings issued or lifted. The data is crucial for enhancing situational awareness and preparation.
- Rescue, donation, and volunteering efforts: Reports of emergency needs or donations of food, water, money, shelter, clothes, medical supplies, etc. This information can be helpful for human organizations and volunteers to plan for relief operations.
- Injuries or death: Reports of injured people, fatality. The data is important to plan for relief where it is needed and assess the situation's severity.
- Affected people and evacuations: Reports about missing, trapped, or displaced people due to the crisis. This information provides human organizations with essential information to plan for aid support.
- Other useful information: Other information that provides useful information to understand the situation
- Not related or irrelevant: Emotional, unrelated to the situation, or irrelevant content.

Recent TREC-IS track [22] has defined a different ontology of 25 tweet information types. Among them, six are identified as actionable information that contains immediate alerts of high importance for emergency responses:

- Requests for goods/services: Reports of particular service or physical goods needed, i.e., equipment, shelter, psychiatric needs, etc.
- Requests for search/rescue: User requests on self-rescues or other rescues
- Call to action - move people: Calls on evacuation, asking people to leave an area and or go to another area
- Reports of emerging threats: Reports of potential problems that can cause damage or loss of life (i.e., buildings at risk, looting, power outage, etc.)
- Reports of new sub-event: A new occurrence that health organizations or governments need to respond to (i.e. trapped people)
- Reports of service available: Providing information on available services (i.e., shelters, hospitals)

Generally, the above six actionable information types and humanitarian classes are overlapped in terms of information. For example, the information type "Reports of emerging threats" mainly includes tweets of the 'infrastructure damage' humanitarian class. In this thesis, we mainly consider models to classify tweets into humanitarian classes. However, in the later part of the thesis, we also apply transfer learning to take steps toward the detection of tweets belonging to actionable information types.

Classification of crisis events has been a topic of increasing interest and has attracted growing research attention. Various methods have been proposed to classify tweets during crisis events. Approaches range from traditional supervised classification methods such as Support Vector Machine (SVM), Naïve Bayes (NB) to recent deep learning and embedding-based models. Verma et al. [156] applied standard machine learning models such as Naive Bayes and Maximum Entropy to identify tweets that contribute to situational awareness during crisis events. The authors utilized both hand-annotated and automatically extracted features for classification. Similarly, Rudra et al. [129] used a Support Vector Machine (SVM) but considered low-lexical and syntactic features. Generally, these studies mainly apply binary classification methods to identify situational tweets. Some works also employed traditional methods for the classification of tweets into multiple humanitarian classes during crisis events [55, 56].

A substantial number of previous works focused on deep learning models with pre-trained embeddings for crisis-related data classification [98, 89, 64, 80]. Nguyen et al. [98] employed a Convolutional Neural Network (CNN) with word embeddings pre-trained on Google news or crisis datasets. Mama et al.[89] compared the performance of Neural Network models with pre-trained word embeddings and traditional machine learning approaches in the classification of crisis-related tweets. Recently, Transformer-based models [155] have been proposed and archived superior performance compared to previous approaches. Liu et al. [80] introduced a robust Transformer for crisis classification. The authors ran experiments on two classification

tasks, namely crisis recognition, crisis detection, and showed that the model achieves better performance than conventional word embedding-based methods. Moreover, many classification models were proposed in TREC-IS track [91] to classify tweets into different information types and predict tweet priorities. The best performing runs also tend to rely on transformer-based models [22].

All the existing works on the classification of tweets during crisis events primarily aim at improving model performance. In this thesis, we try to design models with the best trade-off between model accuracy and interpretability. In other words, our proposed crisis-related classification models in this thesis can both classify tweets into fine-grained humanitarian classes and provide human-understandable explanations of class decisions.

2.3 Tweet Summarization

Automatic summarization is the process of shortening a set of data computationally, to create a subset that represents the most important and relevant information within the original content [167]. In this section, we discuss previous studies on short-text summarization and tweet summarization during crisis events that are closely related to our works in this thesis.

2.3.1 Short-text summarization

Automatic summarization of short texts has become a topic of increasing interest due to the recent exponential growth of user-generated content from social media platforms or e-commerce websites. Summarization approaches can be broadly grouped into two main categories: Abstractive and extractive summarization.

Abstractive summarization. It is the task of generating an informative and concise summary that captures the essential information from the source text. The summary may contain new phrases that are not the same as the original text. One of the typical abstractive methods is Opinosis [40], which was proposed for summarizing short user reviews collected from Tripadvisor, Amazon, and Edmunds. Opinosis uses a word graph to represent input sentences and generates abstractive summaries using the most redundant paths on the graph. Adopting the same approach, Sharifi et al. [138] proposed to use shortest paths instead. Both these algorithms are computationally expensive and not incremental. Later, Olariu [113] proposed TOWGS algorithm that employs a tri-gram graph to generate online summaries incrementally. Nevertheless, these abstractive summarization algorithms often require input texts to be high-overlapping and well-written, hence often returning less informative and less readable when applied to tweets.

Extractive summarization. Extractive methods work by identifying representative text units (e.g., sentences and phrases) from the original text as sum-

maries. This approach is quite prevalent in prior works of automatic tweet summarization [140, 16, 32, 41, 17, 162]. For short tweets, existing extractive methods for summarizing events' tweets generally consist of two steps: (i) sub-event detection, followed by (ii) tweet selection for each sub-event. The first step is crucial for capturing different aspects of events. This step is often based on either burst detection, topic modeling or tweet clustering methods [32, 17, 140, 162]. In the second step, representative tweets or phrases are selected using some variant of PageRank [20] or LexRank [37] algorithms. Though these methods return highly readable summaries, they suffer from the diversity challenge and might not capture less prominent aspects of events. In addition, they are often computationally expensive and/or require prior knowledge for setting their key engineering parameters.

Generally, previous works on short-text summarization mainly focus on the one-time summarization of static datasets. To apply these methods for online summarization of evolving events, we need to re-run the models and make several passes over the data. This is neither scalable nor suitable for large Twitter streams. Few studies worked on online summarization of evolving events [140, 113], but they still suffer from some short-comings such as the requirement of prior knowledge or low diversity. Inspired by previous works on abstractive graph-based summarization, in this thesis, we develop an efficient graph-based summarization method for evolving breaking news events. However, we generate extractive summaries by scoring and extracting representative tweets instead. We address the diversity and scalability shortcomings of previous extractive methods by adapting a scalable, diversified ranking technique [67] for detecting sub-topics. Our method does not require any prior information and returns a diverse set of tweets representing different aspects of an evolving event.

2.3.2 Tweet summarization during crisis events

Unlike general breaking news, crisis-related messages contain distinct features which can be used to design efficient summarization models. Some recent studies have proposed methods specifically for the summarization of crisis events [63, 129, 131, 128, 101, 133].

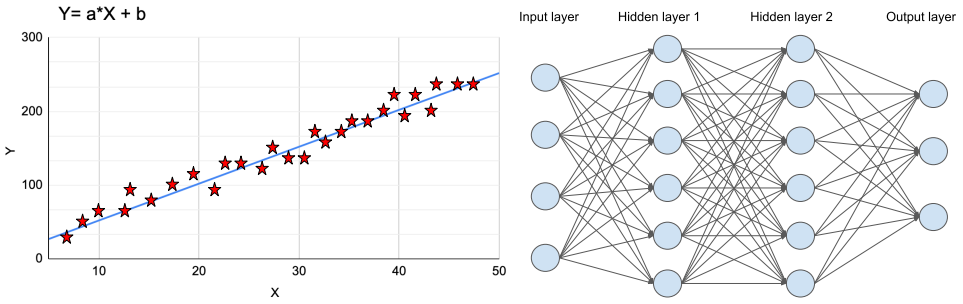
Some works applied clustering-based techniques to detect subtopics and generate diverse summaries of crisis events. Kedzie et al. [63] presented an extractive summarization system that predicts sentence salience and then uses a clustering algorithm to select updates for disaster events. The authors explored disaster-specific features such as geo-locations, disaster-specific language modeling, etc., for summarization. However, the paper focuses on well-written news articles about disasters instead of short, noisy Twitter texts. Similarly, Nguyen et al. [101] applied a Pagerank-based approach for the summarization of a disaster event, where each tweet is represented as a set of important entities, such as *subject*, *event phrase*, *location*, and *number*. The authors suggested that the *subject* answers the question WHAT, which is a cause (such as a hurricane or a road). Meanwhile, *event phrase* represents the action or effect of the

subject (i.e., kill), *location* specifies WHERE the event occurs, and *number* answer the question of HOW MANY (i.e., number of victims). A weighted similarity graph of tweets is built for ranking and extracting the most informative and diverse tweets for the final summary.

A few works focus on specific traits of tweet texts posted during crisis events and develop methods that maximize the coverage of important words. Rudra et al. [129] proposed an extractive summarization technique to summarize situational information from Twitter during disaster events. In this work, some disaster-specific terms were specified as 'content words' such as numerals (i.e., number of injuries, death, casualties, contact numbers, etc.), nouns (i.e., names of places, hospitals, etc.), and main verbs (i.e., died, killed, trapped, etc.). A small set of tweets that have the highest coverage of essential content words were then included in the summary. Later, the authors proposed follow-up works in this direction by using the AIDR platform [55] and introduced extractive methods for summarization of tweets in different humanitarian classes during disasters [131, 130].

Some recent studies employed pre-trained language models for summarization. Saini et al. [133] proposed a multi-objective extractive-based approach for microblog summarization. Different aspects of summary, such as length, TF-IDF score of tweets, and tweet dissimilarity, are optimized simultaneously. The dissimilarity between two texts is calculated using Word Move Distance (WMD) measure, which is the minimum amount of distance that embedded words [95] of one text need to "travel" to reach the embedded words of the other one [68]. However, this method may include redundant tweets in the final summary. For example, two tweets having high TF-IDF scores and differing by only one word can both be selected since the different embedding representations of the two words can lead to a high dissimilarity WMD value. A few works [81, 60, 176] have shown the potential of recent pre-trained models in summarization tasks of news articles. However, news articles contain well-written and formal texts whose traits are completely different from Twitter texts. Moreover, some recently proposed BERT-based models have constraints on the length of input texts (i.e., number of sentences in input documents) and computation time, so it is not effective and robust for disaster situations with millions of input tweets.

In this thesis, we try to capture important phrases in tweets and focus on this content for developing tweet summarization methods. Our goal is to design efficient models that have high performances in terms of standard measuring metrics and require computational complexity that can be suitable for the summarization of disaster events in near real-time.



(a) A linear model (b) A simple neural network with layers of neurons

Figure 2.1. Visualization of a linear relationship and a simple neural network.

2.4 Interpretability

2.4.1 Interpretable Machine Learning

Machine learning systems have demonstrated remarkable performance in many tasks of different domains [118]. However, most of the high-accuracy models remain black boxes whose internal logic workings are hidden, and the reasons for model decisions are opaque. Machine learning systems can be limitedly applicable in real-life scenarios due to a lack of trust on behalf of users. Nowadays, the interpretation of machine learning models becomes an important requirement for any kind of task that has an impact on society or human lives [69]. *Interpretability is the degree to which a human can understand the cause of a decision and correctly predict the method's results [66].* The objective of machine learning interpretability is to interpret or explain the model output decisions so that the model's behaviors can be more transparent and trustworthy. Some machine learning models are interpretable by nature. As an example, for a linear model, the association between input features X and output values Y can be modeled linearly (Figure 2.1a). Meanwhile, for a simple fully connected neural network, it is completely opaque which role each neuron plays or which features are important for model outputs (Figure 2.1b). For this reason, many neural network models are called "black boxes".

Depending on the time point of application, interpretable methods can be broadly categorized into three groups: (i) pre-modeling, (ii) in-modeling, and (iii) post-modeling approaches [62]. *Pre-modeling* approaches are independent of the model and mostly deal with data understanding, visualization, dimensionality reduction, etc. Some statistical techniques for data visualization include t-Distributed Stochastic Neighbor Embedding (t-SNE) [153], Principal Component Analysis (PCA) [1], and clustering methods. *In-modeling* approaches provide explanations for the output decisions by making inherently interpretable models. Some popular approaches are linear models [135], decision trees [115], and rule-based models [39]. For example, Haufe et al. [50] interpreted weight vectors of linear models by visualizing the weight and sign

of features for a specific input. *Post-modeling* approaches improve interpretability by applying methods that analyze and explain decisions of a black-box model after training. Most of the previous interpretability studies fall into this category. Such post-modeling methods could be further categorized into four categories [62]:

- *Feature importance-based explanations*: These methods provide explanations by assigning feature importance scores to input variables. Ribeiro et al. [124] proposed LIME - Local Interpretable Model-agnostic Explanations, which highlights important features that lead to specific decisions. Similarly, Lundberg et al. [85] introduced SHAP - SHapley Additive exPlanations, which give explanations to specific decisions by computing the contribution of each feature to the prediction.

- *Example-based explanations* - these approaches provide interpretability by creating proxy examples of the model, selecting input instances, and observing model outputs to explain the system. Multiple example-based approaches were introduced in the literature. Recently, Wachter et al. [157] proposed a counterfactual method, which reveals the most important variables for predictions of individual instances and how slight changes in input variables can lead to a completely different outcome. Mothilal et al. [96] introduced diverse counterfactual explanations for explanations of machine learning classifiers.

- *Rule-based explanations*: These models extract useful information or comprehensive rules from trained models by re-tracing their internal processes for interpretability. Hailesilassie [49] reviewed various rule extraction algorithms for an artificial neural network.

- *Visualization-based explanations*: These approaches visualize the internal working of machine learning systems for model interpretation. For example, Casalicchio et al. [26] proposed tools to visualize how changes in a feature affect the model performance.

As per model usage, interpretable models are either *model specific*, where model parameters are accessible, or *model agnostic* (i.e., they don't have access to model parameters). Model-specific interpretability methods are limited to a single model or a specific model family. These approaches exploit the internals of machine learning models and reverse engineering approach to provide explanations for model decisions. For example, the interpretation of weight vectors of a linear model is model-specific interpretability. On the other hand, model-agnostic approaches can be applied to any machine-learning model and belong to the post-modeling group.

As per the scope of explanation, approaches are categorized into local and global types. Local interpretability is instance-based, which provides explanations of particular decisions made by the system. Meanwhile, global interpretability approaches explore the whole logic and overall decision process of a model. These methods

provide a general picture of the model and reasonings for output decisions. Global interpretability is generally challenging to achieve in practice since models with many parameters are unlikely to fit into human memory.

Interpretability has been widely studied for many tasks on formal texts but not on noisy, short texts from microblogs. Most of the existing works on the classification of crisis events from Twitter only focused on improving model performance. In this thesis, we develop models that can be both effective and highly interpretable. As mentioned earlier, post-modeling (post-hoc) interpretability approaches are popular. However, it is difficult to evaluate the models due to missing groundtruths. Besides, post-modeling approaches are unreliable and could be easily fooled [142]. Concealed data poisoning attacks can be made to detect the reasoning process of the trained models [158]. Recent research also showed that counterfactual explanations could be manipulated [141], and risk measurement strategies may be used to verify the importance of counterfactuals. Alternatively, some recent papers proposed *interpretable-by-design* models [36, 174] that return explanations/rationales along with output decisions. Inspired by those works, we introduce datasets and in-modeling interpretability approaches for classification in crisis domain.

2.4.2 Attention based explanations

The attention mechanism was introduced by Badhanu et al [9] to address the bottleneck of compressing input sequences into a fixed-length encoding vector in encoder-decoder networks. Such compression is problematic in the case of long sentences, where the decoder would have limited access to the information from the input. Figure 2.2 illustrates a Recurrent Neural Network (RNN) encoder-decoder with attention mechanism in machine translation. The RNN encoder takes an input sequence

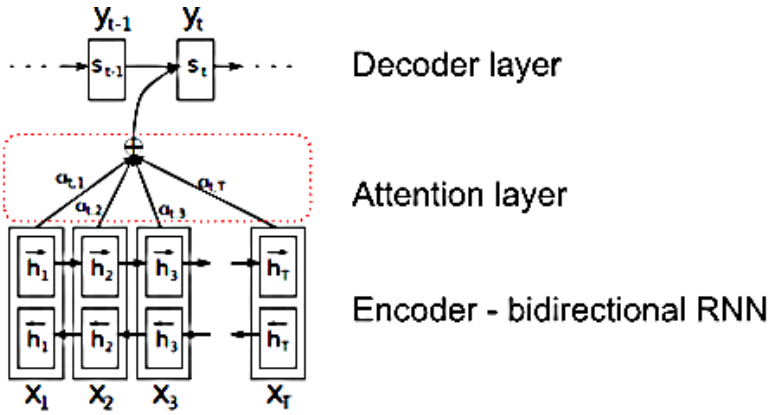


Figure 2.2. The graphical illustration of attentive attention mechanism [9]

.

of length T : $\{x_1, x_2, .., x_T\}$ and produces encoder hidden states H: $\{h_1, h_2, .., h_T\}$. At each step i, the decoder takes encoder hidden vectors and the previous decoder

hidden state s_{i-1} as its inputs and generates outputs y_i token by token. The attention layer allows the decoder to access the encoded input vectors. It induces the attention weights α over the input sequence, which computes the importance of each source hidden state h_j for predicting the current output.

Attention mechanism has shown remarkable performance gains for deep neural networks in many natural language processing tasks. Many researchers proposed attention as a way of modeling explanations [12, 44]. Attention weights over input features have been interpreted as a measure of their contribution to the output predictions. However, recent studies [165, 59, 119] showed that attention is not an explanation as attention can noisily predict the overall importance of input components. Tutek et al [151] analyzed reasons behind the failure of attention weights as a transparency tool. On a similar note, Chrysostomou and Aletras [31] tried to improve the faithfulness of attention-based explanations with task-specific information for text classification. In this thesis, we propose a model that learns faithful attention-based explanations for the classification of crisis events. Unlike previous works, we learn faithful rationales under limited human supervision that take the human comprehension/readability part into account. Our model tends to give high attention weights to consecutive phrases that provide supporting evidence for model outputs.

2.5 Representation Learning

The success of machine learning models highly depends on text data representation. In this section, we present some data representation techniques that are mentioned and used in this thesis.

2.5.1 Word Representation

Words are typically the smallest units for data representation. Simple techniques usually treat words as atomic units where words are represented as indices in a fixed-size vocabulary. However, such a simple method or similar techniques can not capture the similarity between words. Meanwhile, advanced methods were proposed for computing continuous word vector representation from huge unlabeled datasets. Among those, Word2Vec [95] gained great popularity and was used for many Natural Language Processing tasks, including tasks in the crisis domain [56, 98, 89]. The learned word representations, called word embeddings, were shown to capture many linguistic regularities between words, and many types of word similarities can be expressed as linear translations. For example, *vector('king') - vector('man') + vector('woman')* results in a vector that is close to *vector('queen')*. Wod2Vec can be obtained using two neural network techniques: Common Bag-Of-Words (CBOW) and Skip-Gram.

- CBOW: The model takes the context (surrounding words) of each word as input. It learns the word vector representation and predicts the target word.

- Skip-Gram: The model tries to predict context words for a given target word. The training objective is to learn word embeddings that are good at predicting the surrounding words.

Google research group [95] trained Word2Vec models on part of Google News dataset (about 100 billion words) and published pre-trained word embeddings for use of many downstream tasks.

2.5.2 Tweet Representation

The most commonly used document representation technique is sparse Bag-of-Words vectors. Each document is represented by a sparse vector of high dimensions, where each dimension corresponds to the occurrence (i.e., binary indicator or weighting frequency) of a specific word from a dictionary. A simple and popular representation method is TF-IDF, where TF represents the frequency of a word w in a given document d, $\text{TF}(w, d) = count(w, d)$. A word that appears more often in a document is more important. IDF counts the number of documents in the corpus D that a word w appears. Common words are less important in distinguishing documents. $\text{IDF}(w, D) = log(\frac{N}{count(d \in D : w \in d)})$. The combination shows the tradeoff between the two scores $\text{TF-IDF}(w, d, D) = \text{TF}(w, d).\text{IDF}(w, D)$.

In case of tweets, each tweet is usually considered as a document, the set of all tweets forms the corpus. A TF-IDF representation of tweets would be simple, yet it can not capture the semantic meaning of words in a document. Besides, TF-IDF ignores word order and suffers from memory inefficiency triggered by the high dimensionality. Another alternative is to represent documents as the aggregation of pre-trained word embeddings. However, static word embeddings such as Word2Vec lack the ability to represent different meanings of a word in context.

Many recent language representation models have been proposed to learn contextualized embeddings of words and documents. Among those, BERT [35] has created state-of-the-art models for a wide range of natural language processing tasks. BERT is a bidirectional Transformer encoder [155], which is a deep neural network including a stack of attention and fully connected layers. The model has a vocabulary of 30,000 tokens. It is pre-trained on huge unlabaled data over two unsupervised tasks. The first task is *masked language model* (masked LM), where a small percentage of input tokens are masked at random, and the model learns to predict those masked tokens. The second task is *next sentence prediction* (NSP), which predicts the relationship between two sentences A and B whether B is the actual next sentence of A. Figure 2.3 illustrates the BERT pre-training procedure. The inputs are token sequences, which may be a single sentence or two sentences packed together. The first token is a special [CLS] token, the final hidden state corresponding to this first token is used as the aggregation embedding representation for classification tasks. Two input sentences are separated by a [SEP] token. The pre-trained BERT embeddings can be

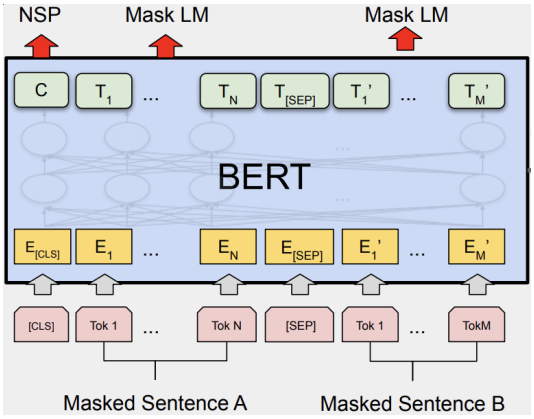


Figure 2.3. BERT pre-training procedures [35].

fine-tuned by appending one additional task-specific layer on top of BERT. Following this idea, Nguyen et al. [99] presented BERTweet - a large-scale pre-trained model for English tweets. In some parts of this thesis, we employ BERTweet for tweet embedding representations and fine-tuning.

3 Efficient Tracking and Summarizing Evolving Events

3.1 Introduction

Twitter has been an extremely popular platform[1] for users to report, seek, and share information about real-world events. It is thus a crucial resource for detecting and monitoring events, e.g., disasters, incidents, etc. [56, 120]. Real-time acquisition and summarization of news-relevant tweets are challenging, though. The first challenge lies in the large scale of the Twitter data stream. The second challenge is due to the prevalence of noise and the wide range of covered topics in Twitter [11, 170]. These challenges require efficient methods for filtering and summarizing relevant tweets from Twitter streams.

Many methods were proposed for the online filtering of tweets during evolving events, but they suffer from some shortcomings when dealing with evolving events.

- A common approach for tweet filtering is to rely on keyword matching methods, where a tweet is considered relevant to a topic if it contains some selected keyword of the topic. The keywords can be frequent words, named entities, or hashtags [160, 161]. While these methods are computationally simple, their performance is poor due to the high diversity of the tweets. Moreover, the selection of keywords requires manual effort, domain knowledge, and/or a large amount of past relevant tweets.

- Some other works propose to filter tweets based on their relevance score to the tracked news, i.e., a tweet is considered relevant if its score exceeds some threshold [132, 178, 6]. An appropriate setting of the relevance threshold requires domain knowledge, extensive data analysis, and/or prior information from similar cases. These requirements are either not available or time-consuming hence not practical when tracking breaking news.

[1] https://en.wikipedia.org/wiki/Twitter

- Another common approach is to employ supervised learning models for determining the relevance of tweets to some specific topic [87, 177, 88]. This approach's performance greatly depends on the training data, i.e. tweets that are labeled *relevant* or *non-relevant* beforehand. However, when tracking breaking news, the number of relevant tweets available for training is initially limited, hence, results are often poor. Moreover, as news evolves over time, existing supervised learning methods often adapt filters by re-training periodically or after accumulating enough new relevant tweets. These methods are, therefore, not scalable due to the high cost of the training process.

After filtering relevant tweets, there is still a vast number of tweets about a specific event, which makes it impossible to have an overall understanding of the event by reading through all the tweets. Online summarization of tweet streams is, therefore, an essential task for studying evolving events. Although there exist many studies on tweet summarization, there are some common drawbacks as follows.

- **Requiring prior information**. A common approach to capturing events' aspects is to apply sub-event detection techniques, which are based on topic models, burst detection or clustering algorithms [43, 140, 32]. These methods' performance highly depends on some pre-defined parameters (e.g., bursting and/or similarity thresholds, numbers of clusters, or topics) whose settings require insights from data and/or prior information, which are not available and not practical in the context of evolving events.

- **Diversity**. As events evolve, their sub-events are often highly skewed in the number of tweets: at a given time point, some are prominent and attract many more tweets than others. Sub-event detection-based methods, therefore, usually result in summaries biased toward major and more mature sub-events, while less popular and/or emerging sub-event might not be detected. Moreover, these methods further assume that tweets written around the same time are about the same sub-event [112, 32], which is not practical in the context of evolving events.

- **Readability**. Another common approach is to summarize tweets by frequent n-grams, phrases, and their concatenations [113]. This allows generating abstractive summaries that are not the same as any original tweet. However, it is only suitable for high-overlapping and well-written short texts but not for diverse and often grammatically incorrect ones like tweets. Our experiments show that these abstractive algorithms when applied in our context of highly noisy tweets, return low-quality and less readable phrases.

- **Scalability**. Most existing models mainly focus on one-time summarization of static datasets using complex algorithms [32, 45]. To apply these methods for generating summaries of evolving events, we need to re-run the models and

make several passes over the data. This is neither scalable nor practical when working with large data streams.

In this thesis, we overcome the shortcomings of previous works by developing novel graph-based methods for online filtering and summarizing tweet streams. Our methods employ word graphs to represent tweets. The graphs allow us to update the representation in real time. Our classifier is a lightweight semi-supervised model that requires minimal human supervision. For summarization, we first apply an incremental algorithm inspired by a diversified ranking approach [67] to detect sub-events from a word graph. This algorithm is totally unsupervised and able to select a diverse set of words representing sub-events. Lastly, most representative tweets are carefully chosen from a small set of candidates containing those selected words and returned as the summary. Our summarization model, therefore, does not require prior information and is highly scalable while returning more informative, diverse, and readable summaries.

3.2 Efficient Tracking Evolving Breaking News

3.2.1 Methodology

Overview. We consider an infinite tweet stream S in which tweets arrive in the order of their published time, and computing infrastructure with limited resources that can store in its primary memory only a small chunk of the stream. Given a breaking news event happening at time t_N, and a small set of tweets relevant to the news T_N that are published within the time duration $[t_N, t_0]$ where $t_0 = t_N + \Delta t$ is a time point shortly after t_N, we want to filter out from S tweets that are relevant to the news and published after t_0. Precisely, when a new tweet arrives, the filter has to *instantly decide* if the the tweet is relevant to the news.

The main idea of our proposed method is to employ a graph-based approach for measuring tweets' relevance to the breaking news and to the *background*. With background, we refer to a representation of all topics in the tweet stream S that occur at around the same time as the news. As the stream consists of tweets in a vast variety of topics, we assume that incoming tweets are mostly relevant to the background and irrelevant to the news we want to track and that news-relevant tweets are outliers. We, therefore, adopt a simple outlier detection approach to distinguish the news-relevant tweets based on the ratio of their relevance scores to the news and background. The overview of our proposed method is depicted in Figure 3.1. The method consists of two phases: initialization and filtering.

In *initialization phase*, we initialize the filter by first building the term graph G_N from the set of initial relevant tweets T_N - which is given as input (line 2, Figure 3.1). We then compute the importance of terms in G_N using a ranking method (line 3). To capture the background of the stream at around the time when the news happens, we

Input:

- T_N - relevant tweets that are published before t_0 (- the start filtering timepoint)
- T_B - randomly sampled tweets published within a short time window before t_0
- S - stream of tweets published after t_0

Output: relevant tweets in S

1: **//Initialization phase**
2: $G_N \leftarrow BuildTermGraph(T_N)$ ▷ **Build term graph for the news**
3: $ComputeTermImportance(G_N)$ ▷ **Compute importance of terms in** G_N
4: $G_B \leftarrow BuildTermGraph(T_B)$ ▷ **Build term graph for *background***
5: $ComputeTermImportance(G_B)$ ▷ **Compute importance of terms in** G_B
6: $(\mu_N, \sigma_N) \leftarrow StatRelevanceScore(T_N, G_N)$ ▷ **Compute mean & standard deviation of relevance scores of relevant tweets**
7: $(\mu_R, \sigma_R) \leftarrow StatRelevanceRatio(T_N \cup T_B, G_N, G_B)$ ▷ **Compute mean & standard deviation of ratios of tweets' relevance scores to news and background**
8: **//Filtering phase**
9: $(\bar{\mu}_N, \bar{\sigma}_N, \bar{\mu}_R, \bar{\sigma}_R) \leftarrow (\mu_N, \sigma_N, \mu_R, \sigma_R)$ ▷ **Record current means & standard deviations**
10: **while** True **do**
11: $m \leftarrow GetTweet(S)$ ▷ **Read a tweet from stream**
12: $r_N \leftarrow RelevanceScore(m, G_N)$ ▷ **Measure** m**'s relevance to the news**
13: $r_B \leftarrow RelevanceScore(m, G_B)$ ▷ **Measure** m**'s relevance to background**
14: **if** $r_B > 0$ **then**
15: $UpdateRatioStats(\bar{\mu}_R, \bar{\sigma}_R, r_N/r_B)$ ▷ **Update mean & standard deviation of tweets' ratio of relevance scores to news and background**
16: **end if**
17: **if** $IsRelevant(r_N, r_B, \mu_N, \sigma_N, \mu_B, \sigma_B)$ **then** ▷ **Determine if** m **is relevant**
18: $Output(m)$ ▷ **return** m **as a relevant tweet**
19: $UpdateTermGraph(m, G_N)$ ▷ **Update** G_N **with terms and edges induced by** m
20: $UpdateStats(\bar{\mu}_N, \bar{\sigma}_N, r_N)$ ▷ **Update mean & standard deviation of relevance scores of relevant tweets**
21: $AddTweet(m, T_N)$ ▷ **add** m **into** T_N
22: **else if** IsSampled(p) **then** ▷ **Determine if** m **is sampled for updating background**
23: $UpdateTermGraph(m, G_B)$ ▷ **Update** G_B **with terms and edges induced by** m
24: $AddTweet(m, T_B)$ ▷ **add** m **into** T_B
25: **end if**
26: **if** IsToUpdate(m) **then** ▷ **Check if it is time to update, based on** m**'s published time**
27: $RemoveOldTweets(T_N, G_N)$ ▷ **Remove old tweets in** T_N **and update** G_N **accordingly**
28: $RemoveOldTweets(T_B, G_B)$ ▷ **Remove old tweets in** T_B **and update** G_B **accordingly**
29: $ComputeTermImportance(G_N)$ ▷ **Re-compute importance of terms in** G_N
30: $ComputeTermImportance(G_B)$ ▷ **Re-compute importance of terms in** G_B
31: $(\mu_N, \sigma_N, \mu_R, \sigma_R) \leftarrow (\bar{\mu}_N, \bar{\sigma}_N, \bar{\mu}_R, \bar{\sigma}_R)$ ▷ **Update means & standard deviations of relevance scores and ratios**
32: **end if**
33: **end while**

Figure 3.1. System overview

randomly sub-sample a sub-set of tweets T_B from all tweets published within a short time window before the start filtering time t_0. The term graph G_B is built from T_B to represent background (line 4). We also compute the importance of terms in G_B using the same way as in G_N (line 5). The construction and updating of term graphs and the computation of terms' importance will be described in detail in the subsequent sections of this paper. Lastly, we compute the mean μ_N and standard deviation σ_N of relevance scores of tweets in T_N - i.e., the relevant tweets - to the news (line 6), and compute mean μ_R and standard deviation σ_R of *relevance ratio* of tweets in $T_N \cup T_B$ (line 7). Here, a tweet's relevance ratio is the ratio between its relevance scores to the news and background. The means and standard deviations will then be used for

deciding incoming tweets' relevance label.

In *filtering phase*, we first use utility variables $\bar{\mu}_N$, $\bar{\sigma}_N$, $\bar{\mu}_R$, and $\bar{\sigma}_R$ to record the current means and standard deviations of relevance scores and relevance ratios (line 9, Figure 3.1). These utility variables are updated after each incoming tweet while μ_N, σ_N, μ_R, and σ_R will be updated periodically. This makes the filter robust against extremely abnormal tweets. Tweets from the stream S are processed in turn as follows. For each incoming tweet m, its relevance scores to the news r_N and to background r_B are measured based on G_N and G_B respectively (lines 12, 13). We will describe in detail the computation of the scores in subsequent sections. If $r_B > 0$, the ratio r_N/r_B is used to update the relevance ratio mean μ_R and standard deviation σ_R by employing Welford's algorithm [164] whose complexity is constant. Next, m's relevance label is determined based on r_N, ratio r_N/r_B, and their means and standard deviations μ_N, σ_N, μ_R, and σ_R (line 17). If m is relevant then it is emitted as output (line 18); G_N is updated using terms and edges induced by m (line 29); $\bar{\mu}_N$ and $\bar{\sigma}_N$ are updated using r_N (line 20); and m is added into T_N (line 21). If m is irrelevant, with some probability $p < 1$, it is chosen for updating G_B (line 23) and added into T_B (line 24). Finally, m's is used for checking if updating of terms' importance is needed (line 26). The condition for this can be either time difference or number of (relevant) tweets found since the last update. If an update is needed, the oldest tweets in T_N and T_B are removed, and G_N and G_B are updated accordingly to the removed tweets (lines 27 - 28). Also, term importance in the graphs is re-computed (lines 29 - 30), and μ_N, σ_N, μ_R, and σ_R are updated (line 31).

Term Graph. Given a set of tweets T, we preprocess each tweet by removing stopwords, punctuation marks, and special symbols (e.g., braces and quotations). The remaining tokens, which we call *terms*, are converted to lower-case, except the URLs embedded in tweets that are case-sensitive. The term graph G of T is then defined as follows. The node set of G consists of all terms appearing in some preprocessed tweet(s) in T. For two terms u and v, if they both appear in some window size L of a preprocessed tweet $m \in T$, then an undirected edge is drawn between u and v in graph G. Here, a window size L of m is a sequence of at most L consecutive terms in m. We also say m contains the edge (u, v), or the edge is induced by m. The weight of edge (u, v) is the summation of weights of all preprocessed tweets in T that contain the edge. In this work, we assume that all tweets have the same weight of 1 though our proposed method works smoothly with a more complex weighting scheme for tweets.

Figure 3.2 shows an example term graph constructed from tweets given in the figure's upper part when the window size L is set to 3. In this example, edges are drawn between *#breaking* and *gun*, and *#breaking* and *fire* as these terms appear in a window of size 3 of Tweet-1. Edges are also drawn between *#knife* and *palace*, and *palace* and *grounds* as these terms appear in a window of size 3 of Tweet-2. The edge (*man*, *knife*) has weight 2 as it is contained in both the two tweets and the other edges have weight 1 as they are contained in only one tweet.

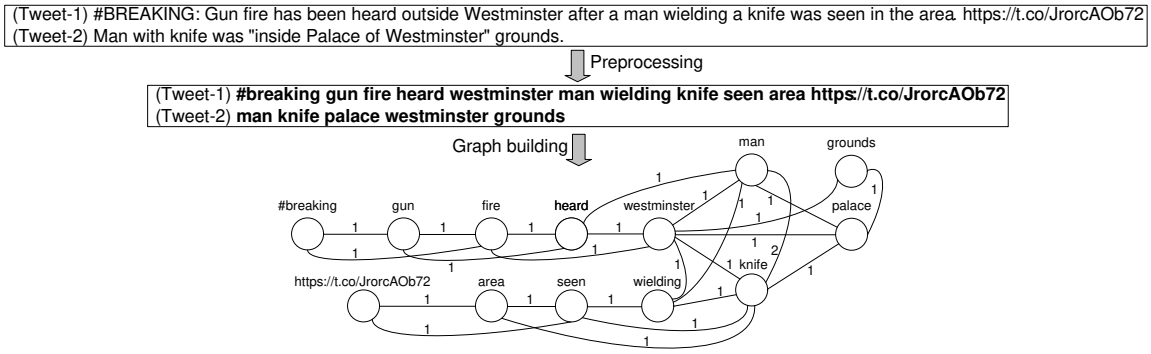


Figure 3.2. Example term graph with window size $L = 3$

Given the term graph G built from the set of tweets T as above, when a new tweet m is added into T, G is updated as follows. We preprocess m in the same way as preprocessing tweets in T. The remaining terms in m are added to G's node set. For each edge (u, v) induced by m, if the edge does not exist in G, it is assigned weight 1 and added into G's edge set. Otherwise, if edge (u, v) is already in G's edge set, its edge weight is increased by 1. When a tweet m is removed from T, the graph G is updated by reducing the weight of edges induced by m by 1. Edges with weight 0 are removed from G's edge set. Terms without edges are removed from G's node set as well.

Computing Terms' Importance. Following prior work on graph-based keyword extraction [94, 37] and information retrieval models [19, 126], we employ Pagerank algorithm [21] for computing the importance of terms in a term graph. Given a term graph $G = (V, E)$ where V and E are node set and edge set respectively, the importance of a term v is a non-negative value that satisfies the following equation for $\forall v \in V$.

$$\pi(v) = \frac{1-d}{|V|} + d \times \sum_{(u,v)\in E} \left[\pi(u)\frac{w(u,v)}{w(u,\cdot)}\right] \tag{3.1}$$

where $\pi(u)$ and $\pi(v)$ are importance score of u and v respectively; $w(u, v)$ is the weight of edge (u, v), $w(u, \cdot) = \sum_{(u,\bar{v})\in E} w(u, \bar{v})$; and $d < 1$ is a constant that is often set to 0.85. Defined in this way, terms' importance can be computed very efficiently using incremental random walk methods [34, 10] or power methods [46].

Computing Tweets' Relevance Score. Given a term graph $G = (V, E)$ and a tweet m, our approach for measuring m's relevance to the topic(s) represented by G leverages both importance of terms and of edges in G. Formally, the relevance score r of m is computed as follows.

$$r = \sum_{(u,v)\in E_m \cap E} \left[\pi(u)\frac{w(u,v)}{w(u,\cdot)} + \pi(v)\frac{w(u,v)}{w(\cdot,v)}\right] \tag{3.2}$$

where E_m is the set of edges induced by m; $\pi(u)$, $\pi(v)$, $w(u, v)$, and $w(u, \cdot)$ are defined in Equation 3.5; and $w(\cdot, v) = \sum_{(\bar{u},v)\in E} w(\bar{u}, v)$. Since both term importance and edge weights are non-negative, tweets' relevance score is also non-negative.

Defined by Equation 3.2, our scoring function gives high relevance scores to incoming tweets that contain not only important words of the input news but also other words which often appear close to such important words in past relevant tweets. On the other hand, tweets where important words appear close with other less frequent words in past relevant tweets are assigned low scores. For example, when filtering tweets relevant to an attack in London, tweets containing *London* together with words like *victims*, *shot*, etc. are assigned higher scores than tweets containing *London* together with words like *music*, *match*, *traffic*, etc. Our scoring function thus improves the precision of keyword-based approaches, while relaxing the strict conditions on consecutive words/terms of language-based approaches, resulting in better recall.

Determining Tweets' Relevance Label. Given an incoming tweet m, if m's relevance score to background $r_B = 0$, we decide that m is irrelevant as we expect that topic evolution is generally smooth, and only noise or tweets about new topics have 0 relevance score to background. Otherwise, we determine the relevance label of m as follows. Assuming that news evolves smoothly, incoming relevant tweets should have relevance scores r_N that do not deviate too much from their mean. We also assume that, since tweets in the stream cover a large number of topics, most tweets are more relevant to background than to the news. Hence, the relevance ratio r_N/r_B is generally small, and only the news-relevant tweets would have large ratios that highly deviate from their mean.

We assume that r_N follows a Gaussian distribution with mean μ_N and standard deviation σ_N, while the ratio r_N/r_B follows a Gaussian distribution with mean μ_R and standard deviation σ_R. We therefore measure the deviation d_N of respectively r_N and deviation d_R of r_N/r_B from their means as follows.

$$d_N = \frac{r_N - \mu_N}{\sigma_N} \quad \text{and} \quad d_R = \frac{(r_N/r_B) - \mu_R}{\sigma_R} \tag{3.3}$$

Then, m is assigned *relevance* label if $d_N \geq -1.3$ and $d_R \geq 1.05$. That means only tweets whose relevance score is out of the bottom 10% and whose relevance ratio is among the top 15% are considered relevant to the news [2].

3.2.2 Datasets

To the best of our knowledge, there is no existing dataset with groundtruth that fits the context of this work[3]. We therefore conducted our experiments on synthetic datasets with groundtruth, and real datasets with proxy groundtruth.

Synthetic datasets with groundtruth. These datasets are synthesized following the procedure shown in Figure 3.3. Given an event $\mathcal{E}$ and a collection of both tweets relevant and irrelevant to the event, we first use the collection to simulate a tweet stream, called *labeled stream*. We then fuse this stream with a real tweet stream, called *carrier stream*, to obtain *fused stream*. The fusion is performed based on the

[2] https://en.wikipedia.org/wiki/Standard_normal_table

[3] The datasets provided by TREC-2012's microblog track are too small: there are only few tens relevant tweets for each dataset

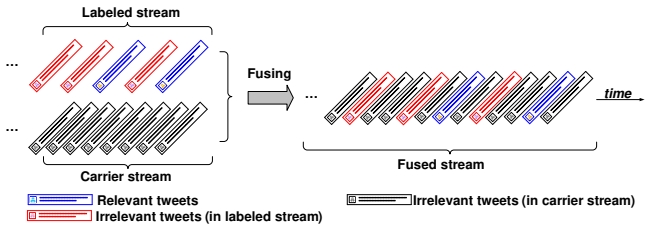


Figure 3.3. Procedure for generating synthetic datasets.

Event	#relevant tweets	#irrelevant tweets	Duration
Sandy Hurricane	6,138	3,870	3 days
Boston Marathon Bombing	5,648	4,364	5 days

Table 3.1. Set of tweets used for generating synthetic datasets.

published time of tweets. The fusion also satisfies the condition that tweets in the labeled stream are located within a time duration Δ of the fused stream that is significantly far from the time of $\mathcal{E}$. This large time gap makes most of carrier stream streams' tweets within Δ not relevant to $\mathcal{E}$. Hence, within Δ, we can confidently use the set of relevant tweets in the original collection as the groundtruth of relevant tweets for the fused stream.

In this work, we re-use two sets of tweets about the *Sandy Hurricane*[4] and *Boston Marathon Bombing*[5] events that were collected by [114] to simulate labeled streams. Basic statistics of these tweet sets are shown in Table 3.1. As the events happened in 2012 and 2013, we crawled tweets in 2017 to simulate the carrier stream. We used Twitter's real-time sample API[6] which gives us about 1% of the whole Twitter stream around the time when the API is called. We kept calling the API through the year 2017 to obtain a consistent sample. This resulted in a stream of around *5 million* tweets per day. For each event, the labeled stream is fused into 15 different time durations of the carrier stream. The durations are carefully selected as to minimize the overlap between the topics of the streams (e.g., not around the date of the events, not within the hurricane seasons, etc.). Finally, we obtained 15 datasets for each event. In our experiments, each event is tracked for the duration of its tweet set as shown in Table 3.1. That means a *Sandy Hurricane* dataset has around $5 \times 3 = 15$ millions tweets, and a *Boston Marathon Bombing* dataset has around $5 \times 5 = 25$ millions of tweets.

Real datasets with proxy groundtruth. Our real datasets consist of two recent breaking news and tweet streams that are formed by crawling Twitter using its sample API as above. The first news, *2017 Westminster Attack*[7], is about the terrorist attack

[4] https://en.wikipedia.org/wiki/Hurricane_Sandy

[5] https://en.wikipedia.org/wiki/Boston_Marathon_bombing

[6] https://developer.twitter.com/en/docs/tweets/sample-realtime/overview/GET_statuse_sample

[7] https://en.wikipedia.org/wiki/2017_Westminster_attack

that took place in the vicinity of the Palace of Westminster in London, UK on March 22nd, 2017. The second news, *Flight 3411 Incident*[8], is about the incident in which a passenger was forcibly removed from United Express' flight 3411 on April 09th, 2017. These are highly dynamic events with many sub-events and topics and have attracted a large number of tweets. In our experiments, each breaking news is tracked for a duration of 3 days. That means each real dataset also has around 15 million tweets.

For real datasets, we employ a pooling approach to construct the groundtruth. That is, tweets returned by any comparative method are judged for relevance. The union set of relevant tweets returned by all the methods is then considered as the groundtruth. Manual assigning labels to tweets requires much human effort due to a large number of tweets. We, therefore, adopt an automatic method for mining topics of the tweets returned by each method and manually judge if the topics are relevant to the news. Then, a tweet is considered relevant to the news if its topic is judged relevant. We used the union set of these tweets as the proxy groundtruth for evaluating the methods.

3.2.3 Experimental Settings

Baselines. We choose the following state-of-the-art methods[9] of keyword-based and supervised learning-based approaches as baselines for evaluating our method. Similar to our proposed method, these methods do not require manual effort nor prior information, hence are suitable baselines.

- **KW**: It is a keyword-based method proposed by Cotelo et al. [33].
- **SL**: It is a supervised learning-based method proposed by Magdy et al. [87, 88].

For each synthetic dataset, its first 50 relevant tweets are given to the filtering methods as input. The remaining relevant tweets are used as groundtruth for evaluating the performance of the filtering methods. For the real datasets, we scan the input streams for tweets around the time when the relevant news happened which contains any of the manually selected strings/ string pairs. These strings/ string pairs are {*#westminsterattack*, *#londonattack*, pairs of *westminster* or *london* with *car*, *terror*, or *pedestrian*} for *2017 Westminster Attack*, and {*united airlines*, *3411*, *#ua*, *#unitedairlines*, *@united*, pairs of *overbook* with *united*, *plane*, *flight*, *passenger*, or *drag*, and pairs of *passenger* or *man* with *drag* or *remove*} for *Flight 3411 Incident*. We manually examine these tweets and select the first 50 relevant tweets for each event as input. Tweets returned by the filtering methods are used for constructing the proxy groundtruth for evaluating the methods. For all the datasets, the tracking starts from the published time of the last input relevant tweets.

[8] https://en.wikipedia.org/wiki/United_Express_Flight_3411_incident

[9] We consider state-of-the-art methods at time of our experiments

In our experiments, for each dataset and each filtering method, the filter is updated after every time step of 30 minutes. For our proposed method, we set window size $L = 4$ (refer to term graph construction). We use $p = 10\%$ of determined-irrelevant tweets to update the background's term graph (refer to line 23, Figure 3.1). Tweets published earlier than twelve time steps are considered old and removed from the term graphs (lines 28 - 29).

Evaluation Metrics. We measure the performance of the filtering methods at different time points across the tracking duration, with respect to the groundtruth relevant tweets up to the time points. That is, if m is the K-th truly relevant tweet, and t_m is the published time of m, then the performance of filtering method $\mathcal{M}$ at K is measured as F_1 score of $\mathcal{M}$ up to t_m. Formally, We denote this score by $F_1(\mathcal{M}, K)$, and denote the sets of all tweets and truly relevant tweets returned by $\mathcal{M}$ up to t_m by $\mathcal{A}(\mathcal{M}, K)$ and $\mathcal{R}(\mathcal{M}, K)$ respectively, then $F_1(\mathcal{M}, K)$ is computed as follows.

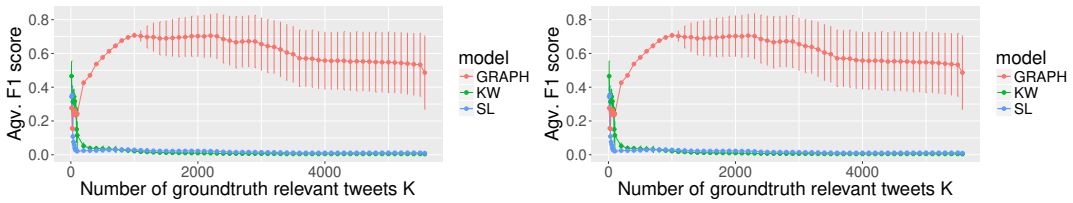
$$F_1(\mathcal{M}, K) = \frac{prec(\mathcal{M}, K) \times rec(\mathcal{M}, K)}{prec(\mathcal{M}, K) + rec(\mathcal{M}, K)} \tag{3.4}$$

where

$$prec(\mathcal{M}, K) = \frac{|\mathcal{R}(\mathcal{M}, K)|}{|\mathcal{A}(\mathcal{M}, K)|} \text{ and } rec(\mathcal{M}, K) = \frac{|\mathcal{R}(\mathcal{M}, K)|}{K}$$

$F_1(\mathcal{M}, K)$ is hence in $[0, 1]$. $F_1(\mathcal{M}, K) = 1$ only when $|\mathcal{R}(\mathcal{M}, K)| = |\mathcal{A}(\mathcal{M}, K)| = K$, i.e, up to t_m the method $\mathcal{M}$ perfectly returns all and only relevant tweets.

3.2.4 Results



(a) *Sandy Hurricane* datasets (b) *Boston Marathon Bombing* datasets

Figure 3.4. Experimental results on synthetic datasets.

Synthetic Datasets. Figure 3.4 (a) shows the F_1 scores over time of the filtering methods on 15 *Sandy Hurricane* datasets. Since all the datasets have the same groundtruth, we average their scores. Similarly, Figure 3.4 (b) shows the average F_1 scores of the filtering methods on 15 *Boston Marathon Bombing* datasets. The figures show that both the two baseline methods have better performance than ours - denoted by **GRAPH** - in a short time duration after the news happens when there are not many relevant tweets. However, their performance decreases rapidly later when there are much more relevant tweets. This is due to the fact that, at first, the baseline methods' filters are unigram-based and weakly trained by datasets with only a small number of truly relevant tweets (i.e., the set of input relevant tweets).

Therefore, they often inaccurately classify all incoming tweets that contain a frequent word in input relevant tweets to be relevant. For example, they classify all tweets containing the word *Boston* to be relevant to *Boston Marathon Bombing*. Hence, they may have good recall but rather low precision. This inaccuracy is amplified when the filters are re-trained in subsequent steps using inaccurately classified tweets obtained from the previous steps. This also makes the baseline methods not robust against the evolution of the news in subsequent stages, hence their performance drops dramatically. The figures also show that our method obtains lower performance at earlier stages but significantly outperforms the baseline methods to obtain much higher performance consistently across subsequent states. This is expected as our proposed method combines the advantages of both unigram- and bigram-based approaches to measure tweets' relevance, hence may have lower recall at first but much higher precision. Consequently, the method is much more robust against the news' evolution, thus effectively filtering incoming relevant tweets.

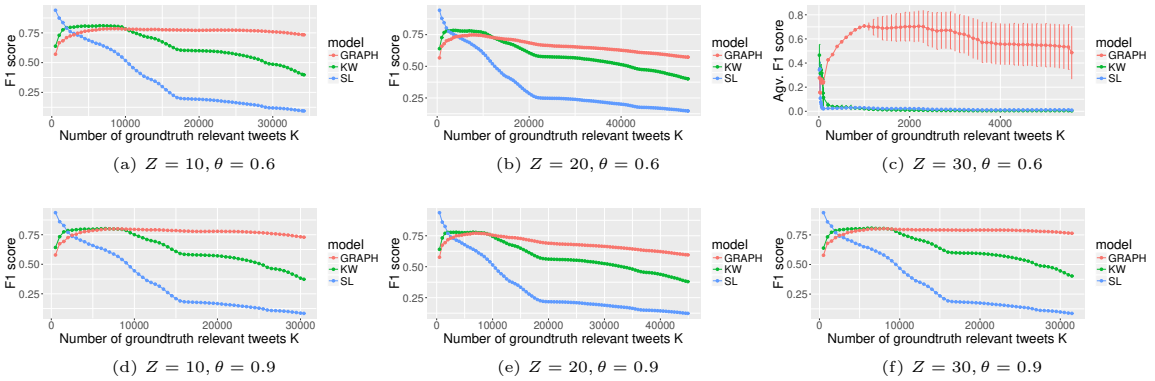
Results on Real Datasets. We used a Twitter-LDA topic model [175] to mine topics of tweets returned by the filtering methods[10]. This model takes as input a set of tweets and a number of topics Z and returns a set of Z topics and probabilities $P(z|m)$ that tweet m is about topic z. Each topic is represented as a probabilistic distribution over terms.

For each filtering model, we run Twitter-LDA on the set of tweets returned by the method with the number of topics Z set to 10, 20, and 30. For each value of Z, the obtained topics are manually judged for relevance based on their top terms and top tweets. Three independent annotators were recruited to judge the topics. These annotators were chosen among our colleague researchers who are knowledgeable about the news and social media but did not participate in this work. The final relevance label for each topic is then decided based on the majority vote of the three annotators. We obtained a Fleiss' Kappa agreement among the annotators of $\kappa = 0.852$ reflecting a high agreement.

A tweet m is considered relevant at threshold $\theta > 0.5$ if $p(z|m) \geq \theta$ for some annotated-relevant topic z. Relevant tweets at θ of all filtering methods are pooled to form the proxy groundtruth for evaluating the methods at Z and θ.

Figure 3.5 shows the performance of the methods on *2017 Westminster Attack* dataset as evaluated at different values of Z and θ, and Figure 3.6 shows similar results on *Flight 3411 Incident* dataset. Again, the figures show that our method has lower performance at early stages but significantly outperforms the baseline methods later in subsequent stages. Moreover, this pattern consistently holds across different settings of Z and θ, which implies the reliability of the evaluation.

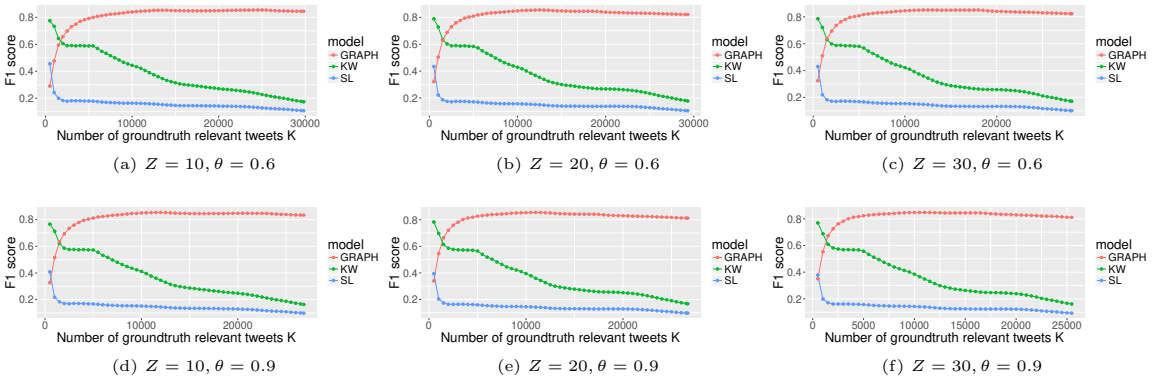
[10] We group tweets by time steps instead of users as in the original Twitter-LDA model.



(a) $Z = 10, \theta = 0.6$ (b) $Z = 20, \theta = 0.6$ (c) $Z = 30, \theta = 0.6$

(d) $Z = 10, \theta = 0.9$ (e) $Z = 20, \theta = 0.9$ (f) $Z = 30, \theta = 0.9$

Figure 3.5. Experimental results on *2017 Westminster Attack* dataset at different number of topics Z and threshold θ.



(a) $Z = 10, \theta = 0.6$ (b) $Z = 20, \theta = 0.6$ (c) $Z = 30, \theta = 0.6$

(d) $Z = 10, \theta = 0.9$ (e) $Z = 20, \theta = 0.9$ (f) $Z = 30, \theta = 0.9$

Figure 3.6. Experimental results on *Flight 3411 Incident* dataset at different number of topics Z and threshold θ.

3.2.5 Complexity Analysis

Computational Complexity. In our implementation, each term graph is represented by a hashmap whose <key,value> elements are pairs of terms and the terms adjacency list. The terms' adjacency lists are also represented by hashmaps whose <key,value> elements are pairs of the term's neighbor and the weight of the corresponding edge. These hashmaps allow us to update each term/edge of the term graph in $\mathcal{O}(\log(|V|))$ operations where $|V|$ is the number of terms in the graph. Term frequency is highly skewed [29], hence $|V| << |T| \times l_{avg}$ where T is the set of tweets used to construct the graph, and l_{avg} is the average number of terms in a tweet - which is quite small as tweets are short[11]. The computational cost of maintaining the term graph is therefore of order log-scale in the number of tweets.

The main cost in processing an incoming tweet includes the cost for preprocessing

[11] https://blog.twitter.com/engineering/en_us/topics/insights/2017/Our-Discovery-of-Cramming.html

the tweet, and that for computing its relevance score. The former is $\mathcal{O}(l_{avg})$, the latter is $\mathcal{O}(l_{avg}L\log(V))$ as the tweet induces $\mathcal{O}(l_{avg}L)$ edges, and each edge needs $\mathcal{O}(V)$ for updating the graph. L is the window size used to extract edges from tweets, which is a small number. The cost for processing a tweet is therefore also of order log-scale in the number of tweets. For computing terms' importance, we employ the power method which converges quickly after a few iterations and has cost $\mathcal{O}(|E|)$ per iteration. $|E|$ is the number of edges in the graph. Again, as term frequency is highly skewed, the term graph is sparse. That is, in most cases $|E| = |V|d_{avg} << |V|^2$ where d_{avg} is the average number of edges of the term in the graph, which is a small number. Hence, the cost of updating term importance usually scales linearly in the number of tweets.

Finally, the main spatial cost of our proposed method is in storing the term graphs. As we use hashmaps to represent the graphs, this cost is $O(|E|)$, so the spatial cost also scales linearly in the number of tweets.

Running Time. We theoretically analyzed the complexity of our proposed method in the previous section. We now empirically examine the method's efficacy by comparing its running time with that of the baseline methods. Table 3.5 shows the running time of the methods on experimental datasets. For the synthetic datasets, the running time is averaged over all 15 datasets for each event. The table shows that, for most of the cases, our proposed method is slightly faster than **KW** method. This is expected as our method has complexity for updating the filter similar to that of **KW** method but has much higher performance in filtering relevant tweets, hence has to deal with a smaller amount of data when updating the filter. The table also shows that our method is much faster than **SL** method. For example, for *2017 Westminster attack* dataset, our method takes around 27 minutes for filtering from a stream of around 1% sample of Twitter in 3 days while **SL** method takes around 9 hours. Our method, therefore can cope well with the large volume of Twitter stream.

3.3 Summarization of Evolving Breaking News

In this section, we describe our proposed method for the online summarization of tweet streams in detail. We start by presenting the principles of our method and sketching its main steps. Then, we discuss the details of each step and analyze the complexity of the method.

3.3.1 Methodology

We consider the summarization task in the following context. We are given a large tweet stream $\mathcal{S}$ in which tweets arrive in order of their published time, and all tweets are assumed to be relevant to a certain event. As the event evolves, at each regular time step, we would like to generate concise summaries of tweets published so

Algorithm 1 Online summarization on tweet streams

Input: Event's tweet stream $\mathcal{S}$
Output: Summaries of event at every time step

```
1: G ← (∅, ∅)                                          ▷ Initialize term graph
2: while !S.end() do
3:     t ← S.next();
4:     UpdateGraphByAddingNewTweets(t, G)
5:     if IsTimeToReGenerateSummary(t) then
6:         UpdateGraphByRemovingOldTweets(G)
7:         DetectSubEvents(G)
8:         ExtractRepresentativeTweets(G)
9:     end if
10: end while
```

far. For simplicity, we focus on the summarizing of the most recent tweets in $\mathcal{S}$ - precisely, tweets published within the sliding time window consisting of the current time step and several previous steps. This is also practical as provided computing infrastructures are usually limited in resources and hence cannot handle the whole stream history, and people are often more interested in recent updates of the event.

In this work, we adopt an extractive approach for the above summarization task. That is, at each time step, the summary is formed by selecting a certain number of most representative tweets in the corresponding sliding time window. Also, as the windows are highly overlapping and events often evolve smoothly, we aim at designing an incremental method for the task. Similar to our proposed approach for tweet filtering task in the previous section, our main idea is to employ a term graph for representing co-occurrence relationship among words in tweets. By using graph based ranking algorithms, we are able to identify terms that represent sub-events described by the tweets. These algorithms are highly scalable and allow incremental update. For dealing with diversity of the sub-events, we further refine the top ranked terms based on diversified ranking algorithms. A small set of tweets containing the terms are then carefully chosen as candidates for the summary. Lastly, a text rank method is applied to select top ranked candidates and output as the summary at desired length.

The main steps of our method are depicted in Algorithm 1. Whenever a new tweet arrives, it is first used for updating the term graph (lines 3-4, Algorithm 1), and then for checking if it is time for generating a new summary (line 5). Before generating a new summary, old tweets (which are published before the current time window) are removed from the graph (line 6). We then detect sub-events (line 7), extract the most representative tweets based on these detected sub-events and return these tweets as summary (line 8).

Term Graph. The term graph G is initialized empty and is updated when a new tweet t arrives as follows. Firstly, t is preprocessed by tokenizing and removing

stopwords, punctuation marks, and special symbols (e.g. quotations or braces). The remaining words are converted into lowercase except URLs, which are case-sensitive. The remaining words in t are considered as terms and added into G node set. An edge between two nodes u and v with weight 1 is drawn if u and v appear within a window of L continuous words of the preprocessed tweet t. If the edge is already in G, we increase its weight by 1. An example of a term graph is illustrated in Figure 3.2.

Sub-event Detection. Given the term graph G, one may detect sub-events mentioned in tweets by performing PageRank [20] on G and return top-ranked terms as sub-events [83, 27]. Formally, let V and E are node set and edge set of G respectively, the PageRank score of nodes in V is the solution of following equation.

$$\pi_v = \frac{1-d}{|V|} + d \times \sum_{(u,v)\in E} \left[\pi_u \frac{w(u,v)}{w(u,\cdot)}\right] \tag{3.5}$$

where π_u and π_v are PageRank score of u and v respectively; $w(u,v)$ is the weight of edge (u,v), $w(u,\cdot) = \sum_{(u,\bar{v})\in E} w(u,\bar{v})$; and $d < 1$ is a constant which is often set to 0.85. There are highly scalable algorithms for computing π and also allowing incremental updates when G is changed [8, 10]. However, the top-ranked nodes by PageRank are often dominated by closely related ones without caring about the diversity [92, 67]. Hence, the sub-events identified by PageRank are likely to be redundant and less diverse. We, therefore, employ a re-ranking approach to adjust the PageRank scores of terms in G when selecting top-ranked terms so as to reduce the redundancy while increasing the diversity. Specifically, we first adapt the algorithm proposed in [67] for selecting top representative and diverse terms that most cover the graph. We then refine the selected terms by removing the ones that appear in most tweets and do not represent any sub-event. In the following, we describe these in detail.

Let $N(S)$ denotes the neighborhood of $S \subset V$ - i.e. $N(S)$ consists of nodes in S and all other nodes adjacent to some node in S. The *marginal utility* $u(v,S)$ when adding a node v into S is the coverage that v adds to $N(S) \cup \{v\}$ and is measured as below.

$$u(v,S) = \sum_{u\in U} \pi_u \text{ where } U = N(\{v\}) \backslash N(S) \tag{3.6}$$

In [67], the authors proposed to find diversified top k nodes by iteratively selecting a node that maximizes the marginal utility when added to the set of previously selected nodes. However, for an evolving event, its number of sub-events varies across time. Hence, we propose to keep adding nodes until the relative marginal utility of the added node falls under a threshold. Specifically, our procedure to detect sub-events is shown in Algorithm 2. In the algorithm, the value of $u(v^*,S)$ decreases significantly after each node added to S as both nodes' number of neighbors and Pagerank scores are highly skewed [77]. Therefore the threshold ϵ can be set to be a small value to stop adding nodes when the new node does not add much coverage increment.

Moreover, for each event, most of its tweets contain some extremely popular terms (e.g., the event's hashtags or entities). These terms should not be used to represent any sub-event though they dominate both the top nodes by PageRank and the top

Algorithm 2 Best coverage set detection

Input: Term graph G, PageRank scores π, a relative marginal utility $\epsilon \in (0, 1)$
Output: a list of representative nodes S

1: $S \leftarrow \emptyset$
2: $utilitySum \leftarrow 0$
3: **while** true **do**
4: $v^* \leftarrow \arg.\max_v(u(v, S))$
5: $utilitySum \leftarrow utilitySum + u(v^*, S)$
6: **if** $u(v^*, S)/utilitySum < \epsilon$ **then**
7: **break**
8: **end if**
9: $S \leftarrow S \cup \{v^*\}$
10: **end while**
11: **return** S

nodes by Algorithm 2. Therefore, we propose the following greedy solution to exclude these popular terms. We measure v's *marginal popularity* $p(v, R)$ when added to node set R by the number of tweets that contain v but do not contain any term in R.

$$p(v, R) = |T(\{v\})\backslash T(R)| \tag{3.7}$$

where $T(R) = \{\text{tweets containing some } u \in R\}$ for any node subset $R \subset V$. Now, we modify Algorithm 2 to filter extremely popular terms by keep ignoring nodes until the relative marginal popularity of the node falls under a threshold. Precisely, our procedure to detect sub-events is shown in Algorithm 3. In the algorithm, the value of $p(v^*, R)$ also decreases significantly after each node added to R as terms' number of tweets is highly frequency [29]. Hence, the threshold θ can be set to be a small value to stop ignoring nodes when the new node does not add much popularity.

Summary Extraction. Given the sub-events S detected as above, we extract K representative tweets using the procedure shown in Algorithm 4. For each sub-event $v \in S$, we iteratively select its n representative tweets $\mathcal{O}_v$ from all tweets that contain v (lines 3-15, Algorithm 4). To do that, we first choose from the (remaining) tweets containing v the one that has the highest average PageRank score over its unique terms (line 6). We then check if the chosen tweet has enough number of unique terms (lines 8-10). This condition is to ensure that the chosen tweet is more likely well written, not just consisting of a few keywords and not informative. Next, we check if the chosen tweet does not highly overlaps with some previously chosen one (lines 11-13). Here, $overlapping(t^*, O_v)$ is the maximum overlap between t^* and any tweet in O_v as measured by Jaccard coefficient[12]. This condition is to reduce the redundancy in the set of tweets selected for each sub-event. Finally, the set of tweets chosen for all the sub-events is ranked using LexRank algorithm [37], and the K top-ranked tweets are returned as the summary.

[12] https://en.wikipedia.org/wiki/Jaccard_index

Algorithm 3 Sub-event detection

Input: Word graph G, PageRank scores π, relative marginal utility threshold $\epsilon \in (0, 1)$, relative marginal popularity θ
Output: a list of representative nodes S

$R \leftarrow \emptyset$
$popularitySum \leftarrow 0$
$S \leftarrow \emptyset$
$utilitySum \leftarrow 0$
while true **do**
  $v^* \leftarrow \arg.\max_v(u(v, S))$
  $popularitySum \leftarrow popularitySum + p(v^*, R)$
  if $p(v^*, R)/popularitySum \geq \theta$ **then**
    continue
  end if
  $utilitySum \leftarrow utilitySum + u(v^*, S)$
  if $u(v^*, S)/utilitySum < \epsilon$ **then**
    break
  end if
  $S \leftarrow S \cup \{v^*\}$
end while
return S

3.3.2 Datasets.

Since we have found no publicly available dataset of events' tweets with groundtruth summaries, we use tweets related to highly popular events as experimental datasets and employ a well-studied offline summarization approach for building the groundtruth.

We conducted experiments on the Twitter datasets related to the following events.

- ***Westminster Attack***: the terrorist attack that took place in London on March 22nd, 2017[13].
- ***Travel Ban***: US's president Donald Trump signed the executive order banning citizens from seven countries from entering US[14].
- ***UA Incident***: the incident in which the aviation security officers forcibly removed a passenger from United Express's flight 3411[15] on April 09, 2017.
- ***DNC 2016***: the national convention of US's democratic party in July, 2016[16].

[13] https://en.wikipedia.org/wiki/2017_Westminster_attack
[14] https://en.wikipedia.org/wiki/Trump_travel_ban
[15] https://en.wikipedia.org/wiki/United_Express_Flight_3411_incident
[16] https://en.wikipedia.org/wiki/2016_Democratic_National_Convention

Algorithm 4 Summary Extraction

Input: Term graph G, PageRank scores π, set of nodes S as sub-events, number of tweets selected for each sub-event n, minimum number of unique terms d, overlapping threshold δ, number of tweets selected for output summary K
Output: Set of K representative tweets $\mathcal{O}$ as a summary

$\mathcal{O} \leftarrow \emptyset$
for $v \in S$ **do**
  $\mathcal{O}_v \leftarrow \emptyset$;
  $T_v \leftarrow getAllTweetsContaining(v)$
  while $(|\mathcal{O}_v| < n)$ **do**
    $t^* \leftarrow \arg.\max_{t \in T_v} averageUniqueWordPageRank(t)$
    $T_v \leftarrow T_v \backslash \{t^*\}$
    if $nUniqueWords(t^*) < d$ **then**
      continue
    end if
    if $overlapping(t^*, O_v) \geq \delta$ **then**
      continue
    end if
    $\mathcal{O}_v \leftarrow \mathcal{O}_v \cup \{t^*\}$
  end while
  $\mathcal{O} \leftarrow \mathcal{O} \cup \mathcal{O}_v$
end for
$\mathcal{O} \leftarrow LexRank(\mathcal{O}, K)$
return $\mathcal{O}$

- ***Hurricane Harvey***: the hurricane Harvey that landed south-east USA and its subsequent flooding in late August 2017[17].

We collected the first three datasets by filtering their events' relevant tweets from Twitter's one-percent sample stream. That is, we continuously crawled tweets during the events' duration using Twitter's sample API[18] to simulate tweet streams. For each event, several modest tweet stream filtering algorithms (i.e., [33, 88]) were then applied to filter out tweets related to the targeting event. The returned tweets are unified and further refined. Specifically, we first mine topics of the tweets using topic modeling techniques [175] and manually judge if topics are relevant to the events, then remove tweets about irrelevant topics. The last two datasets were collected by University of North Texas Libraries [116, 117]. The basic statistics of the datasets are shown in Table 3.2[19]. The table shows that these datasets are diverse in size and duration. Furthermore, the events are also diverse in nature and pace of evolution. These diversities allow us to evaluate the proposed model comprehensively.

[17] https://en.wikipedia.org/wiki/Hurricane_Harvey

[18] https://developer.twitter.com/en/docs/tweets/sample-realtime/overview/GET_statuse_sample.html

[19] For each *DNC 2016* and *Hurricane Harvey* dataset, we used only tweets published in a sub-duration of the original dataset, which

Table 3.2. Basic statistics of the experimental datasets.

Dataset	#tweets	Duration (days)
Westminster Attack	20,694	3
Travel Ban	123,385	7
UA Incident	18,876	3
DNC 2016	2,196,766	9
Hurricane Harvey	5,895,516	10

Groundtruth construction. For each dataset, as the comparative methods generate summaries periodically for every sliding time window (as shall be presented in the following section), we construct a groundtruth summary for each of the windows. To do that, we employ a topic modeling approach for mining topics in the whole dataset and then select the most representative tweets for each window's major topics as groundtruth summary for the window. This approach has been shown to give high-quality summaries when applied on tweet datasets [32, 147, 17, 30]. Here we use TwitterLDA model [175] to mine the topics. This model takes as input the number of topics and returns as output, among the others, the learned topic for each tweet. The proportion of a topic in a sliding time window is then defined by its number of tweets divided by the total number of tweets in the window. The major topics of a window are chosen from the ones having highest proportion in the window until their accumulative proportion exceeds 0.95. Following the previous works, we select from all tweets of the window, the most representative tweet(s) for each major topic based on the tweets' perplexity given the topic[20] as computed from the learned topic model. Lastly, the set of all these top representative tweet(s) selected for a window is considered as its groundtruth summary.

3.3.3 Experimental Settings.

Baselines. We evaluate our proposed method by comparing it against the following methods for text summarization and tweet stream summarization.

- **LexRank** [37]. This is a typical extractive method for text summarization. It ranks sentences by their representativeness using an adaptation of PageRank algorithm [20] for the graph whose nodes are sentences and edges are weighted by pair-wise similarity among the sentences.
- **Opinosis** [40]. This is a typical method for abstractive summarization of short texts. It works by finding prevalent paths on the graph whose nodes POS tagged words and edges are formed between consecutive words in sentences.

consists of continuous days whose number of collected tweets are significantly larger. Hence, the statistics of these datasets are slightly different from their origins

[20] https://en.wikipedia.org/wiki/Perplexity

- **TOWGS** [113]. This is state-of-the-art method for abstractive summarization on tweet streams. It maintains a graph whose nodes are bi-grams in tweets and edges are formed between overlapping bi-grams. The summaries are generated by finding prevalent paths starting from some keywords.

- **Sumblr** [140, 162]. This is the state-of-the-art method for extractive summarization on tweet streams. This method performs stream clustering using nearest neighborhood strategies and maintains a set of representative tweets for each cluster. The summaries are generated by using LexRank on the set of all clusters' representative tweets.

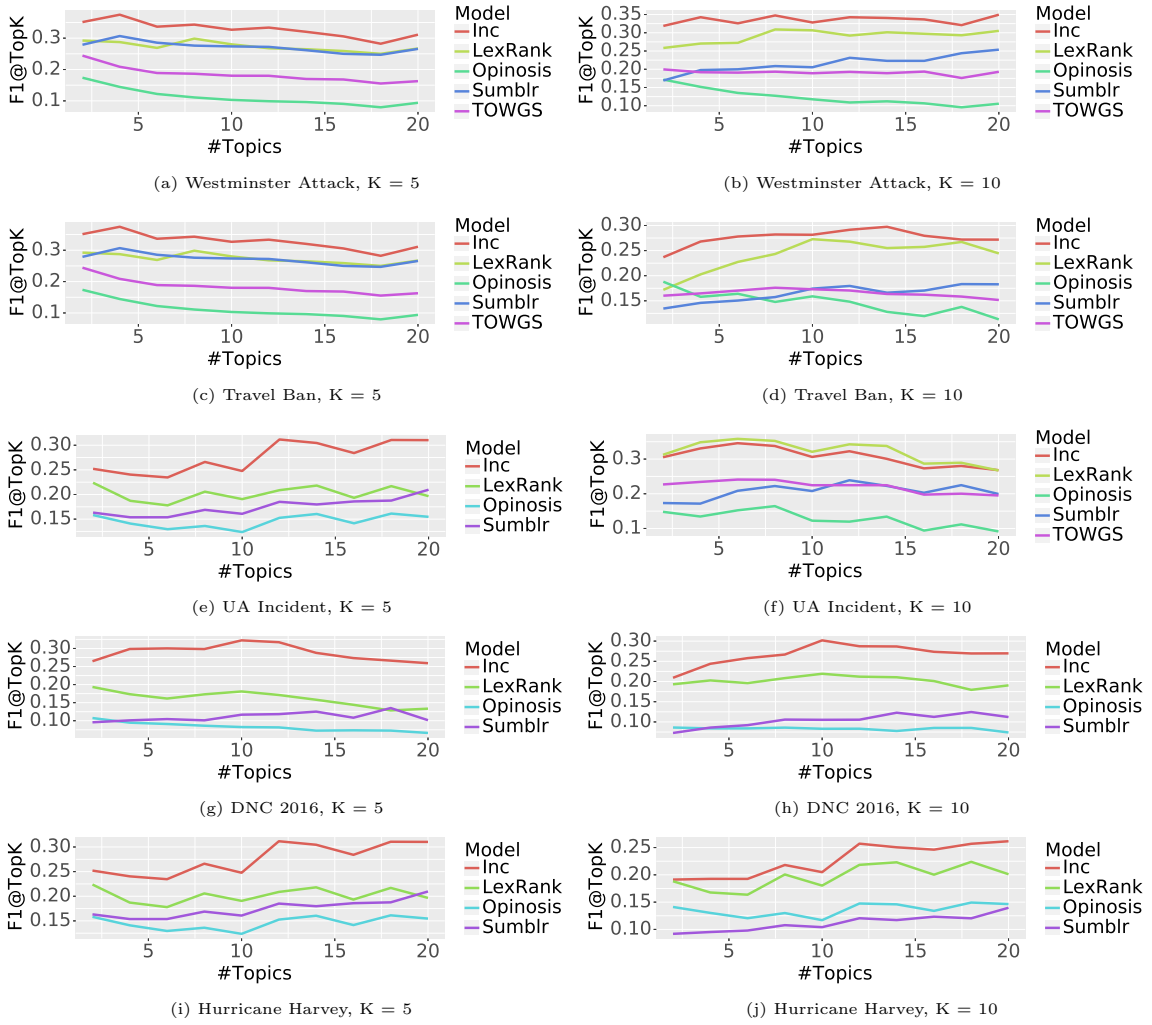
In our experiments, for each dataset and each method, the summaries are produced after every time step of 1 hour, and each summary is generated for a sliding window of 12 time steps (i.e., 12 hours). For our proposed method, we set window size $L = 4$ (refer to term graph construction). Also, we set the marginal utility threshold $\epsilon = 0.05$ and the popularity threshold $\theta = 0.1$ (refer to sub-events detection - Algorithm 3), and set the number of tweets selected for each sub-event $n = 50$, minimum number of unique terms $d = 7$, and overlapping $\delta = 0.3$ (refer to summary extraction - Algorithm 4). For the baselines, we reused and adapted the implementations released by their authors and keep all the parameter settings as originally recommended by the authors.

Evaluation Metrics. Both our proposed methods and the above baselines return the summaries in form of ranked lists of representative tweets or sentences. We, therefore, evaluate these models by examining their **length-K** summaries - i.e., the top K tweets/sentences in each summary. This allows us to evaluate both the conciseness and diversity of the summaries consistently across the methods.

- **Groundtruth based evaluation.** Based on the groundtruth summaries constructed as above, we evaluate the length-K summaries returned by the methods using ROUGE metrics[21]. As suggested by previous studies [75] and following the baselines [140, 162], we choose to use ROUGE-1 scores as it has shown to be most consistent with human judgment. Basically, these scores measure how well the generated summaries cover uni-grams of the groundtruth summary.

- **Human evaluation.** We also recruited students to manually assess the length-K summaries. The students were first educated about the events. They are requested to carefully read the events' Wikipedia page, major timelines, and related articles published on large news sites, as well as scan through the topics mined from the datasets. For each dataset and each sliding time window, each student was then given pairs of length-K summaries - one generated by our method and the other generated by one of the baselines - and asked to choose which summary is more informative about the event. To avoid bias toward any

[21] https://en.wikipedia.org/wiki/ROUGE_(metric)



(a) Westminster Attack, K = 5

(b) Westminster Attack, K = 10

(c) Travel Ban, K = 5

(d) Travel Ban, K = 10

(e) UA Incident, K = 5

(f) UA Incident, K = 10

(g) DNC 2016, K = 5

(h) DNC 2016, K = 10

(i) Hurricane Harvey, K = 5

(j) Hurricane Harvey, K = 10

Figure 3.7. Average ROUGE-1 scores of length-K summaries generated by comparative methods across time steps.

method, the student was not informed about which summary is generated by which method. Also, we randomly order the summaries in pairs and randomly order the pairs given to each student so that consecutive given pairs are mostly independent.

3.3.4 Results

Groundtruth based evaluation. For each dataset, we constructed different sets of groundtruth summaries by varying the input numbers of topics Z of the TwitterLDA model (refer to groundtruth construction - Section 3.3.2) from 2 to 20. This allows us to evaluate the comparative models comprehensively across different granularities of

topics assumed to be presented in the dataset. For each value of Z, at each time step, ROUGE-1 scores of all length-K summaries of all methods are computed against the obtained groundtruth summary.

Figure 3.7 shows the ROUGE-1 scores of the length-K summaries generated by comparative methods on the experimental datasets with $K = 5$ and $K = 10$. For each dataset, the scores are averaged across all its time steps. Though ROUGE-1 scores consist of precision, recall, and $F1$-score, we observe qualitatively similar patterns in all these scores across the datasets and therefore report here only the $F1$-scores to save space. We also do not report the performance of the TOWGS method on DNC 2016 and Hurricane Harvey datasets as the method could not finish the work on these datasets within the time budget of two weeks. The figure clearly shows that, in most cases, our proposed method, denoted by **Inc**, outperforms the baselines by a large margin. It is expected that, on UA Incident dataset, our proposed method outperforms the baselines significantly when $K = 5$ but slightly worse than the LexRank baseline when $K = 10$. This is due to the low diversity of the event: the event has only a few sub-events. Hence, at each time step, the event can be well summarized by a few representative tweets. Therefore, our proposed method's length-K summaries for the event are concise and less redundant when K is small, but get redundant when K is increased. On the other datasets with highly evolving events containing many more sub-events, our proposed model consistently generates better summaries. Overall, the result demonstrates both the outperformance and the robustness of our proposed method over the baselines.

Human evaluation. As Opinosis and TOWGS often return much less readable and less informative summaries (as illustrated case studies below), for human evaluation, we only compare our proposed method against LexRank and Sumblr. Also, since the events evolve smoothly, the summaries of consecutive time steps are largely overlapping. We, therefore, do not manually evaluate summaries of all time steps but for only one after every two consecutive time steps - i.e., one-third of all the time steps. This helps us to reduce the cost of human evaluation work to be manageable while still obtaining a qualitatively consistent result. In the end, the comparative methods' summaries on each dataset were manually assessed by 3 to 5 judges. We got a high agreement among the judges: across the datasets, at least 70% of times the judges have mutual choices[22].

Table 3.3 shows the proportion of evaluated pairs where the length-K summaries generated by our proposed method are judged more informative than those by baseline methods. The proportion is averaged across all the judges of the same dataset. The table clearly shows that, for most cases, our method's summaries are consistently more informative than those of the same length of the baseline methods. This implies that our method is able to extract more diverse sub-events.

[22]We do not report Kappa measures here as they might be misleading in cases with imbalanced choices like ours. Please refer to https://en.wikipedia.org/wiki/Cohen%27s_kappa#Limitations for a more illustrative explanation

Table 3.3. Proportion of evaluated pairs where Inc's length-K summaries are judged more informative.

	LexRank		**Sumblr**	
	$K = 5$	$K = 10$	$K = 5$	$K = 10$
Westminster Attack	0.69	0.75	0.56	0.53
Travel Ban	0.80	0.80	0.89	0.89
UA Incident	1.00	0.94	1.00	0.92
DNC 2016	0.99	0.97	0.86	0.91
Hurricane Harvey	0.93	0.96	0.82	0.88

3.3.5 Discussions

Case Studies. We now present here case studies to illustrate how our proposed method works differently from the baselines. Table 3.4 shows the length-5 summaries generated by all the comparative methods on UA Incident dataset at time step 20, which is around one day since the event happened. As expected, the summaries generated by the Opinosis and TOWGS methods are the least readable and do not contain much information. The former consists of the most frequent phrases while the latter even does not have any meaningful sentence or phrase. The summaries generated by the Lexrank and Sumblr methods are also not very informative. They include short tweets containing some keywords and emotions expressed toward the subject. This is also expected as these tweets have high cosine similarity to many other tweets containing the same keywords, and hence often highly ranked by the similarity-based ranking algorithm used in these two baseline methods. Lastly, the table clearly shows that the summary generated by our proposed method is much more informative than those of the baselines.

Computational Complexity. We use hashmaps to store the word graph's nodes and adjacency lists. These hashmaps allow us to update the graph in $\mathcal{O}(\log(|V|))$ operations where $|V|$ is the number of words in the graph. Since word frequency is highly skewed [29], $|V| << |T| \times l_{avg}$ where T is the set of tweets in the current time window, and l_{avg} is the average number of words in a tweet - which is quite small as tweets are short. Hence, the computational cost of maintaining the word graph is the log scale of the number of tweets.

The computation of PageRank scores from scratch is rather expensive. Since tweets arrive in the stream and events are assumed to evolve smoothly, the word graph should also change gradually over time. We, therefore, employ random walk based methods to update the scores incrementally [10]. These methods have been shown to have almost constant cost and extremely low running time for each incremental update [51]. Moreover, they also allow us to update the scores frequently, i.e., over sub-intervals of each time step, hence do not require much workload whenever we need to generate the summaries.

Table 3.4. Example length-5 summaries

Method	Summary
LexRank	A passenger was forcibly dragged off an overbooked flight. The United CEO says he is sorry for having to “re-acco
	@united I’m still never flying with United again.
	@united Never flying United again!! You should be ashamed.
	Yeah fuck @united flights
	@united If you take someone’s money for a certain flight; they have a right to be in that flight.
Opinosis	Friendly Skies.
	overbooking problem.
	airlines frequent flyer.
	united airlines.
	United card .
TOWGS	jimmy kimmel goes off on united united airlines
	@united i can be a @united flight
	watch jimmy kimmel rips into united airlines
	people are not in that flight they have a flight
	pay for a @united flight that was united airlines
Sumblr	United Passenger Dragged From Overbooked Flight: A man on an overbooked United Airlines flight...
	@united Fuck your airline ,
	Here’s a statement from United Airlines on the man who was dragged off an overbooked flight in Chicago
	@united No. It’s totally unacceptable. You overbooked the flight so paying customers should NOT BE REMOVED. FULL STOP.
	United airlines “fly the friendly skies” ... somehow I don’t think so
Inc	Video appears to show a man being forcibly removed from a United plane by law enforcement in Chicago.
	@united beat a passenger for doing but sitting in the seat he paid for when they overbooked the flight. I hope he sues the
	@United “Fly The Friendly Skies” we throw N a BEATING 4 free! Hey we were not airborne yet!!! We R trying 2 figure out how 2 beat MORE!
	How can #united deplane (violently more so) a random passenger who has paid for his ticket & sitting on his assigned seat
	VIDEO: United Airlines passenger dragged off overbooked flight at O’Hare Airport; officer placed on leave - WLS-TV https://t.co/XCdzpV8EJu

The major computation of our method in each time step is in detecting sub-events based on nodes’ PageRank scores - Algorithm 3, and selecting representative tweets in the final summary - Algorithm 4. The former has a cost of $O(|S| \times |V| \times \overline{deg})$, where $|S|$ is the number of sub-events detected, and $\overline{deg}$ is the average number of edges in the word graph [67]. The latter has a cost of $O(|S| \times \overline{T}_{v \in S} \times n + C_{LexRank})$, where $\overline{T}_{v \in S}$ is the average number of tweets of each sub-event, n is the number of

representative tweets extracted for each sub-event, and $C_{LexRank}$ is the cost for the tweet ranking at line 18, Algorithm 4. Again, as word frequency is highly skewed, $\overline{deg}$ is often a small number. Similarly, since nodes' neighbors and Pagerank scores are highly skewed, Algorithm 3 terminates quickly after a few iterations, i.e., $|S|$ is a small number. Moreover, $\overline{T}_{v \in S} < |T|$, and $C_{LexRank}$ is almost a small constant as we run LexRank on a small set of tweets. All these make the cost of generating the summary for each sliding time window linear to its number of tweets. Our method is, therefore, scalable to large-size tweet streams.

Running Time. We now empirically examine the method's efficacy by comparing its running time with that of the baseline methods. Table 3.5 shows the running time of the methods on the experimental datasets. In the table, the "-" notation in a cell denotes that the corresponding method cannot handle the corresponding dataset within two weeks. The table shows that, for all the cases, our proposed method is much faster than LexRank, Opinosis, and TOWGS methods. Also, on small datasets (refer to Table 3.2 for the datasets' size), our method is slightly slower than the Sumblr method - the only scalable method among the baselines. However, on a large dataset, our method is much faster. Our method is therefore more scalable than Sumblr and can cope well with large volume streams.

	LexRank	**Opinosis**	**TOWGS**	**Sumblr**	**Inc**
Westminster Attack	300+	400+	200+	6	10
Travel Ban	1K+	17K+	18K+	15	20
UA Incident	100+	200+	200+	5	8
DNC 2016	100K+	800K+	-	7.9K+	3.5K+
Hurricane Harvey	500K+	1.1M+	-	9.6K+	6.6K+

Table 3.5. Running time, in seconds, of the comparative methods on the experimental datasets.

3.4 Chapter Summary

In this chapter, we introduced graph-based methods for online filtering and summarizing relevant tweets of evolving events from large-scale Twitter streams. Our classification and summarization methods are highly scalable while efficiently classifying relevant tweets and generating diverse and informative summaries. Experimental results on diverse datasets show that (a) our classifier has better performance in filtering incoming relevant tweets and is more computationally effective than baselines, (b). the proposed summarizer outperforms typical and state-of-the-art baselines as measured by ROUGE-1 scores and human judgment.

In the next chapter, we develop models specifically for the classification and summarization of crisis events. Unlike generic breaking news, messages posted during cri-

sis events can be grouped into fine-grained categories such as infrastructure damage, affected individuals, rescue, etc. We aim to support stakeholders such as governments or health organizations to obtain situational updates and plan for rescue actions. For tasks in crisis domain, interpretability is an important criterion when designing machine learning models, where it is transparent to users how models come to make a specific decision. Hence, we focus on models that obtain the tradeoff between model performance and interpretability.

4 Interpretable Classification and Summarization of Crisis Events

4.1 Introduction

Crisis events work as a trigger for a large volume of real-time information over social media such as Twitter. Local people and authorities post a lot of updates from the ground. Some previous studies [156, 23, 154] have shown the vital role of the Twitter resource in enhancing emergency situational awareness and planning aids. However, in disaster situations, crisis-related messages are immersed in massive sentimental and irrelevant tweets. Besides, humanitarian organizations usually want to obtain information in multiple categories, such as infrastructure and utility damage, caution and advice, injured and dead people, etc. Besides, Twitter users also want to quickly get brief information about events without being overwhelmed with massive data. To fulfill the needs of these organizations and effectively cope with large-scale disasters, it is necessary to develop automated methods to classify tweets into different humanitarian categories and then summarize those tweets in real time.

All existing crisis-specific classification and summarization approaches primarily focus on performance measures, but they did not pay any attention to their decision-making processes. However, such critical systems need to be interpretable in nature [124, 125, 127] so that decision-makers can use them for the purpose. Models are interpretable when humans can understand the reasonings behind output predictions. Besides, in many applications, users prefer simple models with high interpretability. It, therefore, brings to forefront the trade-off between accuracy and interpretability of a model. Despite advances in Natural Language Processing [35] and interpretable Deep Learning models [124, 36, 127, 174], interpreting classification of short, noisy tweets has not been explored. In this work, we aim for a classification model in crisis domain to be interpretable by design. We observe that there are short snippets in tweets, so-called explanations/rationales[1] [36], which provide sufficient evidence

[1] These two terms are used interchangeably throughout the thesis.

to support classification outputs. For example, "*03 Dec 2012 – At least 475 people are killed after Typhoon Bopha, makes landfall in the Philippines*", the phrase "*At least 475 people are killed*" captures essential and sufficient information to classify the tweet to a category about injuries and death. Furthermore, we show that the use of rationales helps improve summarization results of crisis events.

In this chapter, we present an interpretable classification and summarization framework[2] to classify and summarize tweets during disaster events. In the classification phase, we develop a crisis-related microblog classifier based on the idea proposed by Zhang et al. [174]. First, we extract rationales based on a BERT-based multi-task learning approach [25]. Then, the extracted rationales are used to predict the class labels of tweets. Our model is interpretable by design, which is transparent to users about the interpretability of predicted rationales. In the summarization phase, the categorized tweets and rationales are used as the input of an Integer Linear Programming (ILP) framework to summarize tweets. Our summarizer optimizes multiple criteria with flexible constraints, which aim to satisfy different needs of end-users. Experiments on two long-ranging natural disaster events show that our multi-task learning approach achieves high classification performance along with high-quality rationales for the model decisions. Besides, the proposed summarization method surpasses various state-of-the-art baselines in terms of ROUGE-1 F-score and informativeness with human judgment. To the best of our knowledge, this is the first study on interpretable classification-summarization approach on crisis-related microblogs.

4.2 Dataset

Humanitarian Class	**THagupit**	**NEquake**
Caution and advice	467	NA
Infrastructure damage	421	425
Injured or dead people	NA	451
Affected people and evacuations	495	508
Rescue, donation efforts	409	636
Other useful information	434	433
Emotional support and irrelevant	500	500

Table 4.1. Labeled data of two disaster events. NA indicates that the class is absent or merged with another class.

We consider tweets posted in three days of the following two publicly available crisis datasets from CrisisNLP [56].

i. Typhoon Hagupit (THagupit): an intense tropical cyclone, known as Typhoon

[2]Our code is available at https://github.com/HPanTroG/Bert2Bert.

Class	Event	Tweet text
Caution and advice	THAGUPIT	@USER: Super Typhoon Hagupit strengthens with 178 mph max winds as storm tracks toward Philippines.
Infrastructure damage	NEQUAKE	Nepal Earthquake: RT @USER: Kathmandu airport closed following 7.8 #NepalEarthquake.
Injured or dead people	NEQUAKE	RT @USER: Nearly 1,805 dead in Nepalś killer quake, India mounts massive rescue operation
Affected people and evacuations	NEQUAKE	RT @USER: We are a local tampa family and my son is #missing due to the #NepalEarthquake [url]
Rescue, donation efforts	THAGUPIT	#WorldVision is prepared to respond to 55,000 people with emergency essentials. #RubyPH [url]
Other useful information	THAGUPIT	NOW ON ANC: Pagasa update on Typhoon #RubyPH via ANC Alerts
Emotional support or irrelevant	THAGUPIT	R-evenge of the\nU-nfinished\nB-usiness of\nY-olanda\n\nHAHAHAHAHAHA xD stay safe mo guys

Table 4.2. Examples of tweets from various humanitarian classes, the highlighted snippets are rationales.

Hagupit in Philipines. The dataset includes 0.21M tweets posted between December 06 and 08, 2014.

ii. Nepal Earthquake (NEquake): a devastating earthquake in Nepal. This dataset consists of 1.19M tweets posted between April 25 and 27, 2015.

Around 2000 tweets from each dataset are labeled by crowd workers into different humanitarian categories [56], such as "injured or dead people", "infrastructure and utility damage", "caution and advice", etc. These categories are defined and used by United Nations Office for the Coordination of Humanitarian Affairs (UN OCHA). Nevertheless, we have observed many tweets that are wrongly annotated in those datasets. For example, the tweet *"*In real time * #NepalEarthquake India : So sad .. Bangladesh : That wasn't No ball.."* is marked as "infrastructure and utilities damage" in Nepal Earthquake, or *"RT @MyJaps: Stay safe everyone. ὢFὢFὢF #RubyPh"* is labeled as "caution and advice" in Typhoon Hagupit dataset. Besides, such annotations do not contain any rationale labels. Our rationales are short snippets that convey important information for the classification decision. A tweet can contain multiple non-consecutive snippets as rationales. All in all, we perform another round of annotation to revise labels, make them more accurate and annotate rationales.

Unlike some previous works that only consider classes with a sufficient number of tweets [131, 133], we take into consideration tweets of all classes. However, we merge

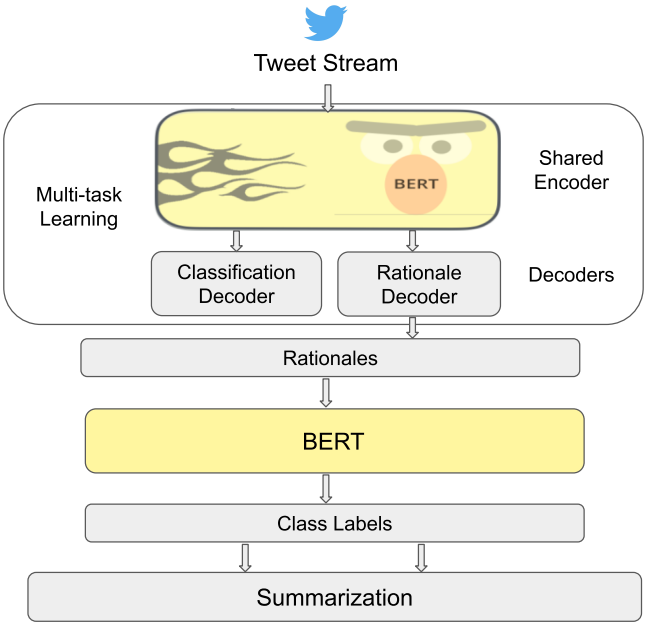


Figure 4.1. An overview of our interpretable classification and summarization framework.

some small classes that report similar information and create a new label for the tweets as "affected people and evacuations" so as to capture all important information. In THAGUPIT, three classes "missing, trapped and found people", "displaced people and evacuations" and "injured and dead people" are merged (there are not so many reports of injuries or death in flood events). Similarly, in NEQUAKE, two classes, "missing, trapped and found people" and "displaced people and evacuations" are merged (reports about injuries and death are prevalent in such events and should be kept as a separate class). The final set of classes is listed in Table 4.1. We illustrate examples of tweets in the pre-defined classes, along with rationales in Table 4.2.

4.3 The Proposed Method

This section presents our proposed method for interpretable classification and summarization of disaster events.

4.3.1 Overview

We consider our classification-summarization approach in the following context. Given a large stream of tweets in chronological order during disaster events, we aim to classify incoming tweets into humanitarian classes with human-understandable explanations and generate summaries of class-level tweets. Figure 4.1 presents the

overview of our framework. Tweets are pre-processed and fed into a BERT-based multi-task learning model that jointly trains two tasks: tweet classification and rationale/explanation extraction of the classifier. Next, the extracted rationales are employed to again classify tweets into humanitarian classes. The second classification step ensures that the model relies on extracted rationales to make predictions. Finally, the set of labeled tweets along with rationales are utilized as inputs of our summarization model. In this paper, we use an Integer Linear Programming (ILP) algorithm to extract salient, non-redundant tweets as summaries. Due to the large-scale disaster events, we allow users to generate snapshot summaries of a specific time interval and a defined length limit.

4.3.2 BERT based Multi-task Classification Pipeline

Data preparation

Our initially labeled data is imbalanced, the majority of tweets belong to the "emotional support or irrelevant" class, while some other classes have only a few tweets. To efficiently supervise our BERT-based interpretable classifier, we decide to gather more data for small classes and annotate rationale information. Firstly, we randomly sample and manually label new data of each event so as to obtain roughly 400 labeled tweets in each class. Next, rationales are annotated. Besides, we sub-sample irrelevant tweets to make our data more balanced. The final classes and number of labeled tweets used for our training process are shown in Table 4.1.

Rationale Identification and Classification

Our pipeline model is a BERT-based supervised encoder-decoder network with two learning stages. In the first stage, we extract rationales based on a multi-task learning structure that jointly classifies tweets into humanitarian classes and identifies rationales in the tweets using a BERT encoder and two decoders. The second stage ignores classification labels in the first stage and applies another BERT encoder to generate the classification prediction based on the extracted rationales alone. We formalize the classification as follows:

Input: Given a set of tweets T, each $t \in T$ is represented as $t =< t_1, t_2, .., t_n >$, where t_i is a BERT-based tokenized token in t.

- **Stage1 Output (Tweet class + Rationale tokens)**:
  - **Output Task 1 (Classification decoder)**: Label $l \in L$ of any given tweet $t \in T$, where L: set of humanitarian classes in Table 4.1.
  - **Output Task 2 (Rationale decoder)**: Token label $r =< r_0, r_1, .., r_n >$, where $r_i \in \{0, 1\}$ to specify whether a token t_i is a part of rationales ($r_i = 1$).

- **Stage2 Output (Classification decoder)**: Final label $l \in L$ of any given tweet $t \in T$, where L is the set of humanitarian classes.

BERT Encoder. We employ BERTweet model [99] to encode input data. Our input tweets are first tokenized and split into a sequence of tokens of the form [CLS] t_1 t_2 .. t_n, where [CLS] is a special token added to mark the beginning of a tweet. We also keep the correspondence between a word and its tokens to later retrieve original words. Rationale labels are assigned to each tokenized token. BERTweet trains a masked language model to generate encoding vectors. Input tokens are padded to a maximum length of 128 - maximum sequence length of BERTweet [99], in each mini-batch. The final hidden state corresponding to the first token [CLS] is used as the aggregate representation of a tweet. BERT Encoder generates embeddings of size 768 dimensions for input tokens. An example of a tokenized tweet in BERT Encoder and our pipeline model is illustrated in Figure 4.2.

Classification Decoder. Our classification decoder generates a class label for each input tweet. The model is trained by appending a fully connected layer with Softmax on top of the final hidden vector in the encoder, corresponding to the first input token [CLS]. We compute a standard cross-entropy loss between the predicted probability p and the true labels y.

$$\mathcal{L}oss_{cd} = -\sum_{l=1}^{|L|} y_l log(p_l) \tag{4.1}$$

where, —L— is number of class labels, $y_l \in \{0, 1\}$ - binary indicator if the current tweet t belongs to class label $l \in L$. p_l is the predicted probability that tweet t is of class label l.

Rationale Decoder. The rationale extraction task is formalized as a binary classification task over input tokens. Given a sequence of tokens in an input tweet, the rationale decoder assigns a binary label to each token, which indicates whether the token is a part of the rationales. In this step, we append a Gated Recurrent Unit (GRU) layer followed by an output layer with Sigmoid function to the last hidden token embedding layer of the shared encoder. The GRU layer helps to capture the dependency between input tokens, yet has fewer parameters than a long short-term memory (LSTM). The presence of rationales can be sparse in some classes, i.e., around 20%-30% words (excluding mentions, URLs) in tweets of "caution and advice" contribute rationale information. To address the class imbalance, we use a weighted binary cross-entropy loss function [28], in which weights are proportional to token probabilities in the input tweets. The loss value of the rationale decoder is as follow:

$$\mathcal{L}oss_{rd} = -\sum_{i=1}^{|N|} \frac{|N|}{|N_{y_i}|}(y_i log(p_i) + (1 - y_i) log(1 - p_i)) \tag{4.2}$$

where y_i and p_i are the true label and prediction value of i-th token respectively, $y_i \in \{0, 1\}$, $|N|$ is the length of the tweet, $|N_{y_i}|$ is the number of tokens with label y_i.

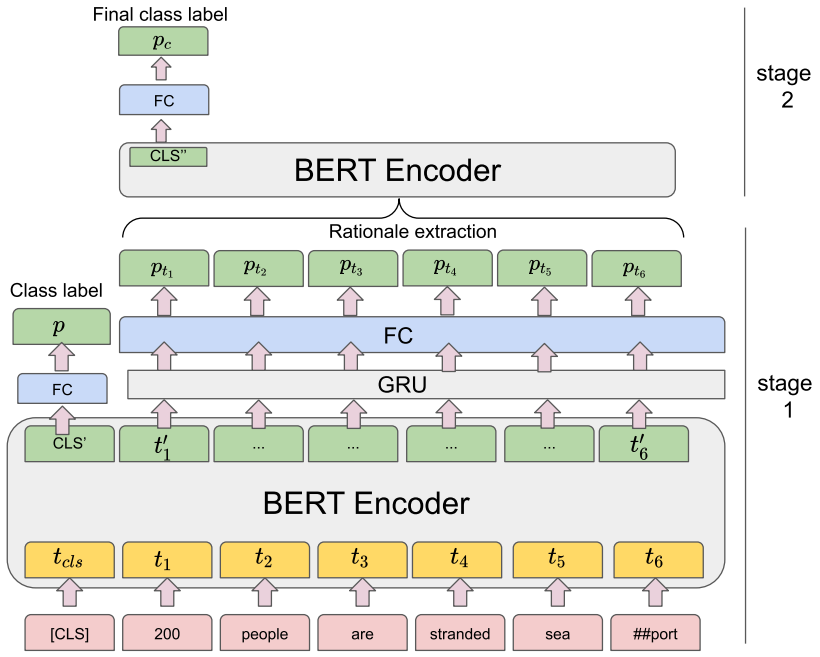


Figure 4.2. Our BERT2BERT model with example of an input tweet. FC indicates a fully connected layer.

Stage1 Prediction. In the first stage, our BERT-based multi-task classifier jointly optimizes losses in the above two decoders. Formally, the overall loss function is defined as follow:

$$\mathcal{L}oss = \mathcal{L}oss_{cd} + \alpha \mathcal{L}oss_{rd} \tag{4.3}$$

where, α is the weight value to regulate losses of the two tasks.

The output of the rationale decoder is at token level. We merge split sub-tokens to retrieve the original words and word-level labels through max-pooling.

Stage2 Prediction. In this stage, we only consider rationale tokens of the tweets, mark other ones with a special character '*' and feed them to the second BERT classifier. The classification decoder of stage 2 generates the final class labels of tweets.

4.3.3 Tweet Summarization

In this section, we propose a method to summarize tweets of different humanitarian classes. First, we apply our trained classification model to generate labels and rationales on data of our three event dates. We observe that the extracted rationales cover the essential content of tweets. Side by side, numerals also play a key role. Thus, our summarization method aims to optimize the coverage of the rationales and numerals.

Given a stream of tweets along with tweet labels and rationale snippets in a

humanitarian class, we build a model to generate summaries of any user-specified time window. We employ an Integer Linear Programming (ILP) framework for our summarization task. Considering a time window of T tweets, a summary of a desired length M words is generated by optimizing the following ILP objective function:

$$max(\sum_{j=1}^{T} t_j + \sum_{i=1}^{U} S(i).u_i) \tag{4.4}$$

where: $t_j \in \{0, 1\}$ indicates whether a tweet j is chosen. U is the number of unique rationale words and numerals in T tweets, $u_i \in \{0, 1\}$ specifies whether a rationale word or numeral i is chosen. $S(i)$ indicates the importance of a word i computed using logarithm of document frequency.

The objective function is optimized with following constraints:

- The summary length should contain at most M words, where M is specified by users.

$$\sum_{j=1}^{T} t_j \cdot Length(j) \leq M \tag{4.5}$$

- If the objective function selects a rationale word or numeral i in the summary, i.e., if $u_i = 1$, then it should select at least one *tweet* containing that word i.

$$\sum_{j \in Z_i} t_j \geq u_i, i = [1 \cdots U] \tag{4.6}$$

  where Z_i is the set of tweets containing the word i.

- All rationale words/numerals in a *tweet* j must be included in the summary if *tweet* j is selected for the summary.

$$\sum_{i \in R_j} u_i \geq |R_j| \times t_j, j = [1 \cdots T] \tag{4.7}$$

  where R_j is the set of rationale words/numerals in tweet j.

The above constraints consider both number of *tweets* (through the t_j variables) and number of important rationale words or numerals (through the u_i variables). Hence, our ILP-based summarizer takes care of multiple requirements, i.e., informativeness, diversity, redundancy, etc. We ensure that the most important informative words get selected in summary, and the optimization function does not get any benefit by selecting the same word multiple times. Overall, this process selects a set of tweets that form an informative and diverse summary. We validate our results in Section 4.5.

We employ the GUROBI Optimizer [48] to solve the ILP. After that, the set of *tweets* j such that $t_j = 1$, represent the summary at the current time window. We define our proposed **RA**tionale word-based **T**weet **SUM**marization approach as **RATSUM**.

4.4 Classification Results

4.4.1 Baseline models

There exist no previous work on the interpretable classification of crisis-related tweets that is similar to our study. Hence, we compare the performance of our disaster classification model with the following previous baselines:

1. **SVM**: A strong and supervised baseline [23, 56, 98] for the classification of crisis events. AIDR [55] also adopted a similar strategy.
2. **RoCNN** [98]: A robust classification of crisis-related data on social networks using Convolutional Neural Network (CNN) with pre-trained word embeddings.
3. **BERT-CLS** [99]: BERTweet model with a sequence classification head on top [53].
4. **BERT-GRU**: BERTweet model combined with a GRU + Attention layer and a final output layer with Softmax. We apply the additive attention formulation proposed by Bahdanau et al. [9] and extract top-k tokens with the highest attention weights as rationales. The value k is set to the average rationale length of human groundtruth for each category in each dataset. Tokens are then merged into original words to obtain final rationales through max-pooling.
5. **BERT-MTL**: Our model with only first stage prediction.

4.4.2 Evaluation Metrics

We use Macro F1 score to evaluate prediction results of the classification models. Besides, we report how well our generated rationales agree with those marked by humans (rationale groundtruth) using Token-F1 metric. Basically, token precision measures the fraction of relevant rationale tokens (words) among the generated tokens, while token recall is the fraction of correctly retrieved rationale tokens among the groundtruth tokens. The Token-F1 reports the trade-off between token precision and token recall.

4.4.3 Experimental settings

We evaluate our model and baseline methods using a 5-fold cross-validation setting. We follow pre-processing or other setting steps in original papers for SVM and RoCNN. For BERT-based models, we pre-process tweets by removing mentions, URLs and then convert tweets to lower case. At each cross-validation run, we sample training, validation, and test sets with ratios 70%, 15%, 15%, respectively. The validation set is used for early-stop settings and hyper-parameters tuning of our model

and all the baselines. BERT-based models are trained with the same setting of 10 epochs, AdamW optimizer [84] with an initial learning rate of 2e-5, and batch size of 16. The bidirectional GRU layer has a hidden size of 128. We specify a grid of candidate values in the range [1e-2, 4e-1] for our hyper-parameter α and compute average F1-scores of classification and rationale extraction tasks with respect to each candidate on validation sets. We select the hyper-parameter that results in the highest mean F1-score (average of Macro-F1 and Token-F1) over five runs on validation sets for test evaluation and new data prediction. The best hyperparameters α on both THAGUPIT and NEQUAKE are 0.07.

4.4.4 Classification Results

We report average scores on test sets over 5-fold cross-validation in Table 4.3. It is not surprising that BERT-based models return superior performance than the traditional machine learning approaches, such as SVM and ROCNN. BERT-GRU achieves high Macro F1, yet low Token-F1 scores on both the datasets. It is consistent with conclusions of previous studies [59, 137] that attentions do not provide a faithful explanation for classification decisions. BERT-MTL and BERT2BERT have the same Token-F1 score since they share the same encoder-decoder structure. Among all the methods, BERT-MTL has the highest classification Macro F1. However, one cannot surely say whether the model relies on rationales for its prediction. BERT2BERT gets high classification performance, and it is transparent to users that the model is interpretable by design, extracted rationales alone are sufficient for correct classification prediction. Our model also performs well (F1 $\geq$ 0.80) for each individual class.

Model	**THagupit**		**NEquake**	
	Macro F1	Token-F1	Macro F1	Token-F1
SVM	0.802	-	0.799	-
RoCNN	0.814	-	0.834	-
BERT-CLS	0.852	-	0.865	-
BERT-GRU	0.850	0.508	0.875	0.642
BERT-MTL	0.857	0.820	0.880	0.856
BERT2BERT	0.847	0.820	0.869	0.856

Table 4.3. Average F1 score over 5 fold cross-validation, '-' indicates that rationales are not extracted by a given method.

4.4.5 Faithfulness of Rationales

In this section, we evaluate the faithfulness of our rationales in terms of *comprehensiveness* and *sufficiency* [36]. We run the second stage of BERT2BERT with two

Dataset	Comprehensiveness↑		Sufficiency↓	
	Human Rationales	**Predicted Rationales**	**Human Rationales**	**Predicted Rationales**
THAGUPIT	0.218	0.294	-0.066	0.005
NEQUAKE	0.283	0.406	-0.097	-0.004

Table 4.4. Faithfulness of rationales.

different input settings and compute the two metrics as follows:

- **Comprehensiveness:** Earlier, we train the classifier with the input tweet t_i. In this part, we train the classifier again with 5-fold cross-validation using $t_i \backslash r_i$, that is, the original input with r_i (rationales) replaced by a special character *. Finally, we evaluate the performance of both the input settings on the test set. For example, "*at least 13 dead after avalanches at mount everest*" and "** * * * after avalanches at mount everest*" present the original and modified data. Next, we measure comprehensiveness as Macro F1(t_i) - Macro F1($t_i \backslash r_i$). High comprehensiveness indicates that rationales highly influence the model performance.

- **Sufficiency:** In this case, we train the classifier using only rationales r_i (other tokens are replaced by *). Finally, we apply the model trained on the original text and the current one on test data and measure sufficiency as follows: Macro F1(t_i) - Macro F1(r_i). A low sufficiency score means our rationales are adequate for the model to make predictions.

In Table 4.4, the comprehensiveness score shows that our predicted rationales are important for classification. Specifically, the prediction performance drops significantly on both datasets when we mask rationales in the input text. Besides, the sufficiency scores are 0.005% and -0.004% on THAGUPIT and NEQUAKE, respectively. This ensures that extracted rationales are adequate for the model to make predictions. Compared to human rationales, higher comprehensiveness and the higher sufficiency of predicted rationales reflects that our extracted rationales are covering more tokens, yet some are false positive. The average ratio of extracted rationale words in input tweets is higher than that of human rationale words by 11%. The token-precision on THAGUPIT and NEQUAKE are 77% and 83%, respectively. Meanwhile, the token-recall are 95% and 94% correspondingly on THAGUPIT and NEQUAKE. Thus, there is still the scope for token-precision improvement.

4.4.6 Agreement between first and second stage prediction

Our BERT2BERT returns two different classification outputs - one in stage 1 and the other in stage 2. We measure the agreement/similarity between the two predicted label sets in terms of accuracy. The average agreement/accuracy scores between the

the two predicted label sets are 90.7% and 92.2% on THAGUPIT and NEQUAKE respectively. The disagreement cases are mainly from tweets with mixture of information, i.e, "*RT @USER: In Sindhupalchok alone, death reaches 1,300. 90% homes destroyed, desperate wait for help. [url] #NepalE...*". The high agreement shows that our rationale extraction in stage 1 is effective for the final classification.

4.5 Summarization Results

In this section, we evaluate our generated summaries in both quantitative and qualitative ways.

4.5.1 Groundtruth summaries

We employ five volunteers to prepare class-level summaries for each day of the events. In the summarization step, we ignore two classes that are not important from a situational point of view, such as "other useful information" and "emotional support and irrelevant". In total, we need to create 4 (class) x 3 (day) = 12 class-level summaries for each event. Volunteers were first asked to prepare summaries of 200 words (excluding #, @, URLs) independently. Next, we iteratively choose tweets selected by most volunteers until we reach a length limit of 200 words to form the groundtruth.

4.5.2 Baseline models

We consider both disaster-specific and recent deep learning-based neural summarization methods as baselines.
1. TSum4Act [101]: A Pagerank-based extractive summarization method for Twitter disaster events. It uses LDA to detect sub-topics before summarizing tweets.
2. APSAL [63]: An affinity clustering-based extractive summarization method for summarization of disaster-related news articles.
3. COWTS [129]: An unsupervised, extractive summarization model of crisis events on Twitter.
4. MOO [133]: An extractive summarization method for Twitter disaster events by jointly optimizing several objective functions.
5. BERTSUM [81]: The recent supervised summarization model for news articles. It formulates the summarization problem as a classification task to identify sentences in the final summary.
6. PACSUM [176]: The strong unsupervised summarization method for news articles. It builds a sentence similarity graph using fine-tuned BERT embeddings and selects sentences with the highest centrality scores in the summary.

7. BERT-GRU: Our summarization model using the extracted rationales of the BERT-GRU classifier.

The first four strategies are disaster-specific approaches, PACSUM and BERT-SUM are neural BERT embedding-based approaches. For all the models, we generate summaries of length M = 200 words.

4.5.3 Evaluation metrics

We measure the summarization performance in both quantitative and qualitative ways.

Groundtruth based evaluation: We use a popular ROUGE toolkit for evaluation [74]. Following baselines and previous works on Twitter summarization [63, 129, 140, 179, 103], we choose ROUGE-1 F-score for evaluating summaries. ROUGE-1 score has shown to be the most consistent with human assessments [75].

Human evaluation: We asked five volunteers to evaluate summaries generated by our model and all the baselines by answering two questions. **Q1**. For each summarization method, we generate 12 summary instances per dataset (hence, 24 instances in total). We give volunteers summaries returned by different methods and ask: Which summary is more informative about the event. This measures the coverage of information in summaries. A summary that contains more informative sentences is considered to have higher information coverage. **Q2**. We give two versions of RATSUM summaries (i). with highlighted rationale words, (ii). without highlighting, and ask volunteers which version they prefer. This evaluates whether the highlighted text reflects important content and helps end-users comprehend the situation better.

4.5.4 Summarization Results

Groundtruth-based evaluation. Table 4.5 shows the ROUGE-1 scores for 24 summary instances returned by our model and all the baselines. Though ROUGE-1 metric includes precision, recall, and F-score, we observe quantitatively similar patterns in all these scores. Hence, we report only F-score in the table. In most cases, RATSUM performs better than all the baseline approaches. On average, our summarization model outperforms COWTS, BERT-GRU, PACSUM by 5%, 8%, 14% respectively. The remaining baselines such as APSAL, TSum4Act, MOO and BERTSUM fall behind RATSUM with a large margin of more than 18% in term of average ROUGE-1 F-score. We also perform Wilcoxon signed-rank test [166] between RATSUM and other baselines. The performance of RATSUM turns out to be significantly better than the baselines with 95% confidence interval ($p-value < 0.05$). Side by side, this trend also holds for ROUGE-2 and ROUGE-L.

Human Evaluation. As MOO and BERTSUM shows low performance compared to other models, and BERT-GRU applies the same method as ours, we do not give

Model	ROUGE-1 F-score (THagupit)											
	Caution & advice			**Affected people**			**Infrastruture**			**Rescue efforts**		
	06/12	07/12	08/12	06/12	07/12	08/12	06/12	07/12	08/12	06/12	07/12	08/12
RATSUM	**0.574**	**0.647**	0.516	**0.642**	**0.615**	**0.641**	**0.516**	**0.483**	**0.609**	0.528	**0.657**	0.535
TSum4Act	0.327	0.419	0.461	0.314	0.356	0.253	0.328	0.303	0.363	0.485	0.401	0.376
APSAL	0.333	0.370	0.423	0.434	0.369	0.383	0.397	0.439	0.421	0.447	0.412	0.317
COWTS	0.544	0.621	**0.561**	0.639	0.574	0.624	0.487	0.469	0.526	0.465	0.594	0.552
MOO	0.330	0.297	0.343	0.386	0.340	0.290	0.337	0.274	0.292	0.394	0.262	0.324
BERTSUM	0.352	0.364	0.431	0.397	0.397	0.368	0.395	0.345	0.398	0.415	0.383	0.327
PACSUM	0.417	0.378	0.467	0.392	0.333	0.408	0.424	0.396	0.389	0.512	0.538	0.545
BERT-GRU	0.465	0.408	0.515	0.454	0.417	0.335	0.442	0.366	0.440	**0.567**	0.511	**0.602**
Model	ROUGE-1 F-score (NEquake)											
	Injuries & death			**Affected people**			**Infrastruture**			**Rescue efforts**		
	25/04	26/04	27/04	25/04	26/04	27/04	25/04	26/04	27/04	25/04	26/04	27/04
RATSUM	0.521	**0.564**	0.404	0.529	**0.526**	**0.556**	**0.581**	**0.580**	**0.472**	**0.644**	**0.651**	**0.576**
TSum4Act	0.336	0.295	0.294	0.446	0.359	0.346	0.422	0.347	0.231	0.390	0.383	0.314
APSAL	0.372	0.336	0.376	0.329	0.307	0.291	0.448	0.323	0.246	0.382	0.363	0.312
COWTS	**0.539**	0.476	0.359	**0.548**	0.439	0.390	0.538	0.409	0.386	0.456	0.459	0.549
MOO	0.372	0.303	0.339	0.278	0.355	0.238	0.333	0.273	0.297	0.300	0.228	0.300
BERTSUM	0.377	0.393	0.379	0.350	0.326	0.421	0.415	0.391	0.380	0.418	0.309	0.305
PACSUM	0.409	0.345	0.327	0.515	0.389	0.446	0.402	0.492	0.460	0.473	0.440	0.300
BERT-GRU	0.501	0.536	**0.422**	0.451	0.506	0.554	0.441	0.556	0.373	0.553	0.608	0.522

Table 4.5. ROUGE-1 F-score of summarization models. The best scores are in bold, the second bests are in brown color.

the results of these models to volunteers to reduce workload. For each dataset, we get 60 responses to a given question (5 volunteers x 12 summary instances). Table 4.6 illustrates the fraction of responses. In THagupit dataset, 47% of respondents find our generated summaries more informative. The second and third informative models are COWTS and PACSUM. It is generally consistent with the above groundtruth-based evaluation results. In NEquake dataset, 83% of respondents prefer our model in terms of informativeness. It is significantly higher than the evaluation on THagupit dataset. We observe that the NEquake dataset is much bigger, each category covers more sub-events. The human evaluation and our observation indicate that RATSUM tends to work well on large datasets with many sub-topics. Table 4.6 also illustrates the high preference of highlighted text. 100% of volunteers think the highlighting is useful and more user-friendly. We illustrates an example of 100-word summaries generated by RATSUM and COWTS in Table 4.7. RATSUM is shown in the format with highlighted rationales.

4.5.5 Discussion on Performance

In this section, we discuss possible reasons why our model is superior to the baseline methods. The disaster-specific summarization baselines generally perform worse than RATSUM due to various reasons. TSum4Act [101] clusters tweets to sub-topics

Datasets	Model	Q1	Q2
THAGUPIT	RATSUM	47%	100%
	COWTS	19%	NA
	APSAL	6%	NA
	TSUM4ACT	11%	NA
	PACSUM	17%	NA
NEQUAKE	RATSUM	83%	100%
	COWTS	8%	NA
	APSAL	3%	NA
	TSUM4ACT	3%	NA
	PACSUM	2%	NA

Table 4.6. The fraction of responses that a method is preferred by users. NA indicates that the question is not asked for a given method.

Reports indicate **80% homes near #Nepal #Earthquake epicenter collapsed**. CARE's responding. **Some of Nepal's world heritage sites are damaged or destroyed** in earthquake. **India Flights to Kathmandu put on hold**: **Domestic airlines today put on hold their services** t... #business #kerala. The 7.9 earthquake dat hit nepal has **dstroyed buildings, cellphone netwrks r down nd power is out** #MSGHe... Initial pictures after #Nepalquake show **major damage to buildings and structures**. **Nepal earthquake devastation could cost billions**: Here's how to help. **#Tibet severely affected** by #NepalEarthquake; **houses collapsed, communications cut off**. **Nepal declares state of emergency** after killer quake.	Reports indicate 80% homes near #Nepal #Earthquake epicenter collapsed. CARE's responding Terrible news from Nepal. Donations here. Pic of devastated Palace area taken 10 days ago. Witnesses: Some buildings collapse in Nepal capital after 7.7 quake: By Gopal Sharma and Ross Adkin KATHMANDU (Reuters) - Nepal urged... Devastating visuals of destruction in Nepal....thoughts,prayers and all protective energies for this tragic loss of life..... Katmandu's poorly constructed buildings worsen quake outcome. Nepal earthquake devastation could cost billions: Here's how to help. Nepal Earthquake: Extensive Destruction, Rising Death Toll. Still can't believe what I witnessed in #NepalQuake today. History crumbling, a nation in despair.

Table 4.7. An example of 100-word summaries (excluding #, @, URLs) generated from tweets in "infrastructure damage" class (NEQUAKE 26/04) by RATSUM and COWTS.

and selects the most informative ones in each cluster using a Pagerank-based method. The model assumes that all clusters are equally important and select the same number of tweets in each cluster. This assumption might not be valid in disaster scenarios, in which some sub-topics might cover more critical information than others. APSAL [63] selects tweets based on specific features of sentences in news articles such as sentence position or language models representing the language of disasters. These features are usually missing in noisy, short texts of Twitter datasets. BERT-GRU falls short behind our model due to the low quality and instability of extracted rationales, as we discussed in Section 4.4.4. It obtains the best performance for a few summaries, and the remaining cases are significantly worse than RATSUM. Finally, COWTS [129] considers nouns, numerals, and main verbs as important words and tries to cover these words in summaries. However, in some cases, other words (i.e., adjectives) also play an essential role in disaster-related tweets. RATSUM works

better because it does not only look at words separately but considers informative phrases of rationales in the context of tweets. COWTS behaves quite competitive with RATSUM model on small or less diverse datasets.

Our embedding-based summarization baselines show high computational complexity and low performance in the summarization of large-scale short texts. MOO [133] generally prefers long sentences with high TF-IDF scores. The extracted tweets by MOO are also redundant due to the drawback of Word Move Distance (WMD) based dissimilarity strategy. Besides, the computation of WMD scores is expensive. Next, the supervised model BERTSUM [176] falls short in our experiment due to the difference in specific traits of well-written news articles and tweets. BERTSUM and some supervised neural summarization models [97] grow parameters with the length of the input documents. Therefore, it fits well for news articles, but not large tweet sets. We adapt the model by breaking down our tweet datasets into sub-documents. However, BERTSUM faces another challenge of highly imbalanced data, with only a few tweets are in the groundtruth summary. Another embedding-based summarizer, PACSUM generates less diverse summaries than RATSUM. PACSUM is specifically designed for news articles, it learns similarity between input texts by fine-tuning BERT on news articles datasets. The model builds a directed graph for sentence selection under the assumption that relative positions of sentences influence the centrality, i.e., preceding sentences are more central. However, the assumption is not true for a set of equally important tweets on Twitter. Besides, it is also computationally expensive to extract BERT-based similarity scores for all pairs of tweets when building the PACSUM graph.

4.5.6 Discussion on generalization.

Our model requires intensive initial labor work for rationale annotation. However, it can generalize well on new data. To observe the ability of our approach for generalization, we download 1000 labeled tweets of the recent Mexico earthquake event [2] and evaluate both classification performance and rationale extraction. We first manually check labels and then annotate rationale snippets. Then, we train BERT2BERT model on 100% NEQUAKE dataset with 10 epochs and predict class labels and extract rationales for evaluation. The performance on new data is shown in Table 4.8. Although we do not use any in-domain data of Mexico dataset for training, our model achieves good performance on both tweet classification and rationale extraction tasks.

4.6 Chapter Summary

This chapter presents an interpretable classification and summarization framework for disaster events on Twitter. We leverage an interpretable by design approach to

Class	#Tweets	Macro F1	Token-F1
Infrastructure damage	164	83.95	86.09
Injured or dead people	166		
Affected people and evacuations	157		
Rescue, donation efforts	161		
Other useful information	168		
Emotional support or irrelevant	185		

Table 4.8. Performance of BERT2BERT on new Mexico dataset.

develop BERT2BERT classifier for crisis-related microblogs. Our evaluation shows the efficacy of BERT2BERT over baseline methods. We also show that the extracted rationales are beneficial for the summarization of tweets. Our RATSUM summarizer turns out to be good for both informativeness and human understanding. The model is robust and simple, yet able to generate informative summaries in near real-time. We observe that the performance of the classification-summarization model depends on the latent representations of tweets. Hence, in the next chapter, we focus on methods that can learn good tweet representations in vector space for a better classification performance-interpretability tradeoff and a less computationally expensive summarization method.

5

Contrastive Learning based Interpretable Classification and Summarization of Crisis Events

In the previous chapter, we develop a BERTweet-based interpretable approach for tweet classification during crisis events. However, it is observed that the language model BERTweet model was not pre-trained on similarity tasks, and tweets are not well represented in vector space. For example, semantically dissimilar tweets may have high cosine similarity scores. In this chapter, we propose an interpretable classification approach that has a better classification performance-interpretability tradeoff, and results in better embedding representations. Thenceforth, we develop an equally good summarization model with the one from the previous chapter but has significantly lower computational complexity.

5.1 Introduction

Classification and summarization of crisis events have attracted great attention from researchers [98, 128, 133, 150]. In general, the performance and applicability of classification-summarization frameworks depend on two factors — (i). the representation of tweets in latent embedding space and (ii). understanding the decision-making process of the model. While the first factor helps in boosting the performance, the second one ensures the interpretability of the model that, in turn, helps in the adaptation of such systems in real-life usage. Some recent studies have shown the success of contrastive learning approaches in advancing data representation and further boosting task performance on image and formal text datasets [144, 65, 42, 73]. Inspired by these studies, we employ a contrastive loss to learn better embedding representations of tweet data for improving our classification and summarization performance.

In this chapter, we present a classification and summarization framework[1] to classify tweets into different humanitarian classes and summarize this information.

[1] Our code is available at https://github.com/HPanTroG/RACLC

Our proposed *Rationale Aware Contrastive Learning based Classification* (RACLC) model is generally an advanced rationale-based interpretable-by-design model from the previous chapter. It can both classify tweets into fine-grained humanitarian classes and extract short snippets, so-called rationales as explanations for model decisions. The model consists of two learning stages. In the *first stage*, RACLC learns rationales by jointly optimizing three loss values, i.e., losses of class label prediction, rationale extraction task, and an additional contrastive loss [65]. In the *second stage*, we feed the extracted rationales to a simple BERTweet model with a Softmax output layer on top to classify tweets into humanitarian categories. This step shows the interpretability of the predicted rationales.

Next, we propose an integer linear programming-based summarization approach that maximizes the coverage of rationale words and minimizes redundancy by discarding duplicate or near-duplicate tweets. Contrastive learning-based latent representations of tweets help in the detection of near-duplicate tweets (Section 5.3.2). We call our summarization approach as *RAtionale Aware Contrastive Learning-based Tweet Summarization* (RACLTS).

5.2 Datasets

We download tweets posted on two consecutive dates of four publicly available Twitter datasets from CrisisNLP [56, 2].

(i) **Nepal Earthquake (NEquake):** The dataset consists of 0.83M tweets on April 25 and 26, 2015 during Nepal earthquake.
(ii) **Mexico Earthquake (MEquake):** The dataset contains 0.08 tweets on July 20 and 21, 2015 during Mexico earthquake.
(iii) **Typhoon Hagupit (THagupit):** The dataset contains 0.16M tweets on December 06 and 07, 2014 of Typhoon Hagupit disaster.
(iv) **Cyclone PAM (CPam):** We download 0.11M tweets on March 15 and 16, 2015 during Cyclone PAM.

Class Label	**#Tweets**			
	THagupit	**CPam**	**NEquake**	**MEquake**
Infrastructure damage	421	396	425	390
Rescue and donation efforts	411	398	636	381
Affected people and evacuations	502	396	508	399
Injured or dead people	NA	NA	451	395
Caution and advice	469	404	NA	NA
Other useful information	431	364	433	399
Emotional support and irrelevant	493	411	497	438

Table 5.1. Our labeled datasets. NA indicates an absent class.

Each of the above datasets contains roughly 1000-2000 tweets annotated by crowdworkers into different humanitarian classes [56, 2]. In the previous chapter, we revised class labels and provided annotated rationales for NEQUAKE and THAGUPIT datasets. Following this, we annotate rationales for MEQUAKE and CPAM datasets. Recall that rationales are short phrases that provide evidence for class labels. A tweet can contain multiple non-continuous rationale phrases. The final labeled datasets are shown in Table 5.1.

5.3 Methodology

5.3.1 Tweet Classification

In this section, we present our **R**ationale **A**ware **C**ontrastive **L**earning based **C**lassification of crisis events (RACLC). Figure 5.1 illustrates the overview of our RACLC model. We aim to build a classifier that is interpretable by design. Our model consists of a pipeline with two learning stages. The first stage applies a multi-task learning approach with the help of a contrastive loss [65] to extract rationales. The second stage employs the extracted rationales as inputs and predicts final class labels. This stage shows that the extracted rationales are explanations for the class labels and makes our model interpretable by design. We illustrate the structure of RACLC in Figure 5.2, and describe stages in the following parts.

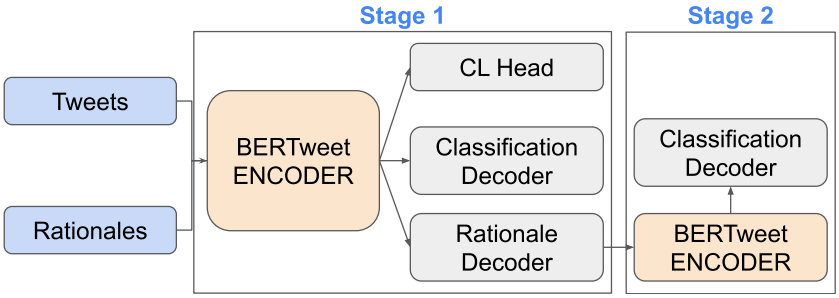


Figure 5.1. RACLC Overview.

Stage 1: Rationale Extraction

Our rationale extraction task is formalized as follow: Given a tweet set T, each $t \in T$ is represented as $t =< w_1, w_2, .., w_m >$, where w_i is a word in t. Our model learns to assign label $l =< l_1, l_2, .., l_m >$, $l_i \in \{0, 1\}$ indicates whether a word is a part of rationale ($l_i = 1$).

The first stage is a multi-task learning classifier with a shared BERT encoder, two decoders, and a Contrastive Learning Head (CL-head). We want the rationale

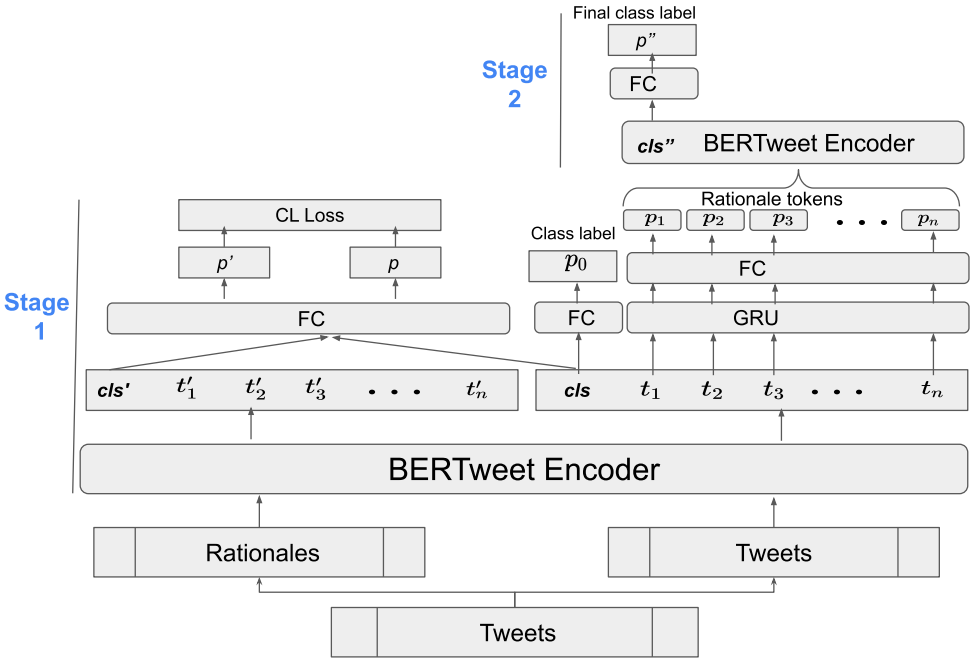


Figure 5.2. Our RACLC model. CL Loss is a contrastive learning loss. FC and GRU indicate fully connected and Gated Recurrent Unit layers, respectively.

extraction process to be influenced by the label prediction task. Thus, the two tasks are learned jointly. Besides, we also leverage the beneficial properties of CL loss to learn better latent representations of tweets and improve the model performance.

BERTweet encoder. We use BERTweet model [99] to encode input texts. BERTweet has the same architecture as BERT [35], but it is trained on Twitter datasets. Each tweet is tokenized and represented as [CLS] t_1 t_2 .. t_n, where t_i is the i^{th} token of the tweet, [CLS] is a special token added at the beginning, and it is used as aggregate embedding representation of the input tweet. Rationale labels are assigned at token level. We pad input tokens to a maximum length of 128 [99]. BERTweet encoder encodes tweets and generates embeddings of 768 dimensions.

Rationale decoder. The rationale decoder predicts a binary label for every token in input tweets. This decoder consists of a GRU (Gated Recurrent Unit) layer followed by a Sigmoid output layer. The number of rationale/non-rationale tokens in tweets can be highly imbalanced. To address this problem, we apply a weighted cross-entropy loss function [28], in which weights are inverse probabilities of token labels in the input tweets. The loss is then computed as follows:

$$\mathcal{L}oss_{rd} = -\sum_{i=1}^{|S|} \frac{|S|}{|S_{y_i}|}(y_i log(p_i) + (1-y_i) log(1-p_i)) \tag{5.1}$$

where $y_i \in \{0, 1\}$ - correct label of i-th token, and p_i - predicted probability that i-th token is of label y_i, $|S|$ - tweet length (token count), $|S_{y_i}|$ - number of tokens of

label y_i .

Classification decoder. The classification decoder classifies tweets into predefined humanitarian classes. We apply a sequence classification head on top of BERTweet encoder for prediction. The decoder optimizes the following cross-entropy loss:

$$\mathcal{L}oss_{cd} = -\sum_{c=1}^{|C|} y_c log(p_c) \tag{5.2}$$

where, $|C|$ - number of labels, $y_c \in \{0, 1\}$ - an indicator whether a considering tweet t is of label $c \in C$. p_c - predicted probability that the tweet t has label c.

CL Head. The CL approach aims at pulling semantically close sentences together and pushing apart non-neighbors. We follow the self-contrastive learning (SCL) framework [65], which optimizes an SCL loss with in-batch negatives. Given N instances in a training mini-batch, SCL requires augmenting positive examples for each original instance. Different from images, data augmentation remains inherently difficult in NLP due to the discrete nature of texts. *In this study, we suggest that the original tweet and its rationale snippets are semantically close and employ rationales as augmented data.* During training, rationale snippets of a tweet are concatenated and fed to the shared BERTweet encoder. The CL Head maps BERTweet-based embeddings of rationales and tweets to vectors $\{z_i\}$ by a single linear layer. The additional CL loss is then computed as follows:

$$\mathcal{L}oss_{cl} = -\sum_{i \in I} log \frac{exp(z_i \bullet z_{j(i)}/\tau)}{\sum_{a \in A(i)} exp(z_i \bullet z_a/\tau)} \tag{5.3}$$

where $i \in I \equiv \{1...N\}$ is the index of an arbitrary input in the batch, $j(i)$ is the index of the corresponding augmented positive sample. The $\bullet$ symbol denotes the inner dot product, $\tau \in \mathbb{R}^+$ is a scalar temperature parameter, and $A(i) = I \backslash \{i\}$ are negative samples.

Overall, our final loss function is:

$$\mathcal{L}oss = \mathcal{L}oss_{cd} + \alpha \mathcal{L}oss_{rd} + \beta \mathcal{L}oss_{cl}$$

The rationale decoder returns predictions at token level. We retrieve the labels of original words through max-pooling. A word is predicted as rationale if any of its constituent tokens is a rationale.

Stage 2: Label Classification

This step takes rationales returned by the first stage for class prediction. Specifically, the input data is the original tweets with extracted non-rationales masked by a special token '*', and the word order is maintained. We also employ a BERTweet encoder to encode data. The embedding representation corresponding to the first token [CLS] is then fed to a fully connected Softmax layer to determine tweet labels.

5.3.2 Tweet Summarization

In this section, we use tweets along with class labels and rationales predicted by RACLC to generate class-level summaries. Near-duplicate tweets of each class label are first removed. Then, the remaining tweets are fed into an optimization model for summarization. In this part, we ignore two classes which are 'emotional or irrelevant' and 'other useful information, since they do not provide important and actionable information in crisis events.

Temporal evolution and near-duplicate removal

Generally, messages posted at the time of crisis events are highly overlapped in terms of information. To reduce memory overload and computation time, we remove highly similar tweets for our summarization. Typical deduplication methods are usually based on word overlap. However, many tweets report the same or similar information, but have no or few words in common. For example, two tweets '*500K people flee #nepal earthquake*' and '*half million residents evacuate the quake area*' are semantically similar, but expressed in different ways. Some previous works [42, 123] have proposed fine-tuned embeddings for similarity tasks. However, these studies were designed for formal texts and not suitable for Twitter datasets. As observed in many recent works [65, 144, 42] and our experiments (section 5.4.4), pre-trained contrastive-based embeddings can generate good semantic representations of the input data. Therefore, we propose a temporal graph method based on RACLC embedding representations for near-duplicate removal. We expect that the deduplication based on semantic similarity can help remain good performance but reduce the computation time significantly.

Tweets to be summarized are fed to our RACLC model, and embeddings are extracted at CL Head. All the tweets are considered in time order. We build a temporal graph G = (V, E), where V is the set of all considering tweets. An edge $(t_0, t_1) \in E$ if cosine similarity between RACLC embeddings of the two tweets close to 1.00, tweet t_0 is posted before t_1. Then, we select tweets/nodes gradually. Whenever a node is selected, all its adjacent nodes in the graph are removed from consideration. In this way, we obtain the minimum set of nodes that are reachable to all the nodes in the graph by direct edges. The set of selected nodes forms a deduplicated set of

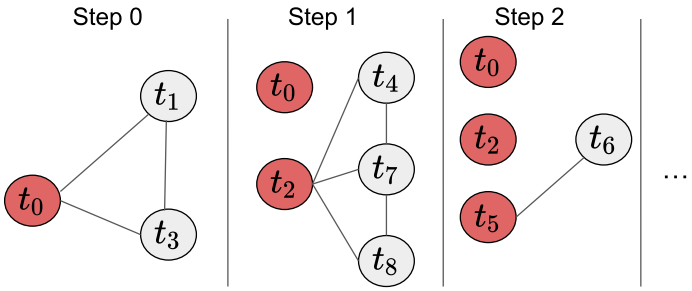


Figure 5.3. Near-duplicate removal. Nodes in red are selected.

tweets. An example of tweet selection is shown in Figure 5.3.

Summarization

We define the summarization task under the following context. After the classification phase, we have a set of tweets, their class labels, and rationales. Besides, we obtain a set of deduplicated tweets from our near-duplicate removal step. Our tweet summarization can be generated over any pre-defined time period. We apply an Integer Linear Programming (ILP) based framework that jointly optimizes a set of tweets (N), and rationales to generate a summary of length L. We observe numerals are equally important; hence, we also consider numerals along with the rationales. Henceforth, we use the term rationale for both original predicted rationales and numerals. The ILP objective function is computed as follows:

$$max(\sum_{i=1}^{T} z_i + \sum_{k=1}^{R} I(k).r_k) \tag{5.4}$$

where: $z_i \in \{0, 1\}$ - an indicator whether a tweet i is selected. R - number of unique rationale words in T, $r_k \in \{0, 1\}$ indicates whether a rationale k is selected. $I(k)$ specifies the importance score or weight of a rationale word k computed using the logarithm of the document frequency of k. We also tried other PageRank-based weightage methods, but it does not give any significant benefit over document frequency based scheme. *Note that, while computing the weights, we consider the entire tweet set corresponding to a specific disaster class and date, not just the deduplicated set.*

The objective function is optimized with the below constraints:

$$\sum_{i \in X_k} z_i \geq r_k, k = [1 \cdots R] \tag{5.5}$$

$$\sum_{j \in Y_i} r_k \geq |Y_i| \times z_i, i = [1 \cdots T] \tag{5.6}$$

$$\sum_{i=1}^{T} z_i \cdot Length(i) \leq L \tag{5.7}$$

Where X_k - set of tweets containing a rationale k, Y_i - set of rationales in tweet i.

Our model considers both number of *tweets* (z_i variables) and number of important rationales (r_k variables) in our objective function. We want to ensure in Eqn. 5.5 that if the rationale word k is selected by the objective function in the summary, i.e., if $r_k = 1$, then at least one *tweet* containing rationale word k must be selected. Eqn. 5.6 ensures that if the objective function selects a tweet i for the summary, i.e., $z_i = 1$, then all the rationale words present in tweet i must be included in the summary. Finally, Eqn. 5.7 guarantees that at most L words (user-specified) are present in the summary.

The above ILP formulation takes care of summarization requirements, i.e., informativeness, diversity, redundancy, etc. The optimization function gets benefits by discarding the selection of the same word multiple times. *Deduplication step helps the ILP method by reducing the number of tweets that need to be processed by the optimization function.* Our ILP problem is solved using Optimizer [48]. The set of *tweets* i, with $z_i = 1$, forms our summary at the current time window. Our proposed RAtionale Aware Contrastive Learning-based Tweet Summarization approach is defined as RACLTS.

5.4 Classication experiments and results

5.4.1 Baseline Models.

We employ the following models on Twitter classification tasks as baseline models.
1. SVM: A strong classification baseline on many Twitter problems [23, 56, 98].
2. Robust-CNN [98]: A CNN model with pre-trained word embeddings for Twitter classification of crisis events.
3. BERT [35]: BERTweet transformers with a classification layer on top of the pooled output.
4. BERT-GRU: BERTweet model combined with a bidirectional GRU layer + additive attention [9]. Top k tokens with the highest attention weights are extracted as rationales. k is set to be the average length of human rationale tokens in the considering category.
5. LCL [146]: Label-aware contrastive loss based classification model.
6. BERT2BERT(-2stg) [108]: It is similar to RACLC model without CL Head and second stage.
7. BERT2BERT: It is similar to our model without CL head
8. RACLC(-2stg): Our RACLC model, without the second stage.

5.4.2 Evaluation Metrics.

We evaluate both label prediction and rationale extraction tasks based on human groundtruth. The label prediction task is measured using Macro F1. For rationales, we compute token-level precision, recall, and F1-score of the extracted rationales using human groundtruth. Here, we report only F1-score for brevity.

Besides, we measure the faithfulness [36] of the extracted rationales to ensure that the rationales cover all important evidence and are sufficient for model prediction using the following metrics:

Comprehensiveness. It measures whether all supporting evidence is covered. We run experiments with two contrast input settings X and $X \backslash R$, that is, the original tweets with the predicted rationales marked by a special token '*'. Then, we observe the difference in performance. *Comprehensiveness = Macro F1(X) - Macro F1(X\R)*. A high score indicates a high impact of rationales on model performance.

Sufficiency. It measures the performance difference of a model on original data and the input data with non-rationales marked by '*'. *Sufficiency = Macro F1(X) - Macro F1(R)*. A low sufficiency value indicates that rationales alone are adequate to make predictions.

5.4.3 Experimental Settings.

In this study, we evaluate our model in both in-domain and cross-domain setups.

In-domain evaluation. The models are trained and evaluated on the same dataset with a 5-fold cross-validation setup. At each fold, we sample the train/validation/test set with the ratio 70%-15%-15%, respectively. The validation set is used to tune hyperparameters for the baselines and our model. Input tweets are first converted to lowercase, then URLs and mentions are removed. We conduct a grid search for hyper-parameters and choose values that return the highest average Macro F1 and Token F1 on validation sets to evaluate results on test sets. Our RACLC is trained for 10 epochs with learning rate 2e-5, a batch size of 8 and Adam optimizer [84]. The GRU and CL Head output size is 128. We set the temperature parameter τ to 0.05. The best α and β are $\alpha = 0.17, \beta = 0.05$ on NEQUAKE and MEQUAKE datasets, $\alpha = 0.15, \beta = 0.07$ on the other two datasets.

Cross-domain evaluation. We evaluate the performance of classification models on a new dataset. We train the models on one dataset and evaluate them on another dataset (i.e., models trained on NEQUAKE are used to generate class labels and evaluate results on MEQUAKE). However, we do not use models trained on earthquake datasets for the evaluation of THAGUPIT and CPAM datasets due to the mismatch between class label sets.

To make new predictions on unlabeled datasets for further purposes (i.e, summarization), we use our RACLC model trained on both NEQUAKE and MEQUAKE for

Model	In-domain							
	NEQUAKE		MEQUAKE		THAGUPIT		CPAM	
	Macro F1	Token F1	Macro F1	Token F1	Macro F1	Token F1	Macro F1	Token F1
SVM	0.799	-	0.738	-	0.802	-	0.768	-
Robust-CNN	0.833	-	0.787	-	0.817	-	0.843	-
BERTWEET	0.864	-	0.851	-	0.852	-	0.888	-
LCL	0.865	-	0.850	-	0.856	-	0.864	-
BERT-GRU	0.876	0.640	0.857	0.592	0.850	0.513	0.879	0.577
BERT2BERT(-2stg)	0.874	0.857	0.855	0.826	0.857	0.820	0.891	0.868
BERT2BERT	0.862	0.857	0.836	0.826	0.847	0.820	0.861	0.868
RACLC(-2stg)	**0.890**	**0.871**	**0.869**	**0.874**	**0.865**	**0.847**	**0.896**	**0.893**
RACLC	0.869	**0.868**	0.842	**0.874**	0.845	**0.847**	0.871	**0.893**
Model	Cross-domain(Train//Test)							
	NEQUAKE//MEQUAKE		MEQUAKE//NEQUAKE		THAGUPIT//CPAM		CPAM//THAGUPIT	
	Macro F1	Token F1	Macro F1	Token F1	Macro F1	Token F1	Macro F1	Token F1
SVM	0.679	-	0.661	-	0.524	-	0.523	-
Robust-CNN	0.683	-	0.730	-	0.602	-	0.671	-
BERTWEET	0.837	-	0.851	-	0.853	-	0.822	-
LCL	0.835	-	0.849	-	0.800	-	0.819	-
BERT-GRU	**0.852**	0.636	0.852	0.622	0.841	0.540	0.816	0.610
BERT2BERT(-2stg)	0.829	**0.862**	0.842	0.839	0.818	**0.873**	0.815	0.831
BERT2BERT	0.841	**0.862**	0.847	0.839	0.851	**0.873**	0.808	0.831
RACLC(-2stg)	0.849	**0.862**	**0.855**	**0.851**	**0.858**	0.867	**0.829**	**0.833**
RACLC	0.832	**0.862**	0.850	**0.851**	0.813	0.867	0.819	**0.833**

Table 5.2. Classification Performance. The best results are in bold. - if a model does not extract rationales.

earthquake-related tweets. Similarly, RACLC model trained on both THAGUPIT and CPAM are employed for the prediction of typhoon-related datasets.

5.4.4 Results

Model Performance

Table 5.2 shows the results of classification methods for both in-domain and cross-domain evaluation.

In-domain evaluation. BERT-based models achieve better results than SVM or ROCNN by a large margin. BERT-GRU classifies tweets well on our four datasets (i.e., 0.876 and 0.850 Macro F1 on NEQUAKE and THAGUPIT respectively). However, the model does not obtain high Token-F1. BERT-GRU is unable to identify proper rationales marked by humans, so it is not interpretable in that sense. For example, the model correctly classifies the tweet "*Typhoon #Hagupit Triggers Massive Evacuation In #Philippines #news HTTP*" to "affected people and evacuation", but words with highest attention weights in highlights are not covering the important content to explain the decision. Our observation is on par with some previous works [59, 137, 108]. All the models obtain pretty good performance, so the LCL approach does not

Dataset	Comprehensiveness↑		Sufficiency↓	
	Human Rationales	Predicted Rationales	Human Rationales	Predicted Rationales
NEQUAKE	0.288	0.352	-0.097	-0.005
MEQUAKE	0.331	0.259	-0.070	0.009
THAGUPIT	0.225	0.349	-0.051	0.007
CPAM	0.325	0.403	-0.029	0.017

Table 5.3. Faithfulness RACLC.

give benefits in our case. RACLC(-2stg) and BERT2BERT(-2stg) outperform all the other methods on both label prediction and rationale extraction tasks. However, these models are not transparent to users whether they rely on the extracted rationales to make predictions. RACLC has the same Token-F1 score with RACLC(-2stg) due to the same shared first learning stage. *Having the second learning stage in* RACLC *drops Macro F1 scores, but the models become interpretable by design.* The use of CL loss boosts the performance of both classification and rationale extraction tasks. For example, RACLC and RACLC(-2stg) significantly outperform variants without CL loss, which are BERT2BERT(-CL) and BERT2BERT(-CL-2stg) respectively.

Cross-domain evaluation. SVM and ROCNN falls short for cross-domain evaluation. Meanwhile, BERT-based models perform quite well in cross-domain. Our RACLC (-2stg) shows superior performance on both Macro F1 and Token-F1 metrics. RACLC (-2stg) obtains the best performance, and RACLC has the best trade-off between classification performance and interpretability.

Correlation between first and second stage prediction

Our RACLC model outputs two different classification results, which are from the 1st and 2nd stages. However, we observe a high agreement between the two outputs of the two stages. Specifically, the average accuracy scores of the two outputs over 5 folds on NEQUAKE, MEQUAKE, THAGUPIT and CPAM are 92.6%, 88.5%, 91.0% and 90.5%, respectively.

Faithfulness of rationales

Table 5.3 illustrates the faithfulness of machine extracted and human rationales. The low sufficiency shows that our extracted rationales alone are adequate for prediction. Negative scores indicate that non-rationale masking helps remove distractors (noise) and improve performance. However, the sufficiency of human-annotated rationales is better than machine-predicted ones, although they fall short in terms of comprehensiveness. We observe that machine-predicted rationales are generally longer than human-annotated ones. Some of these machine-predicted tokens are not rationales. However, RACLC is able to maintain the trade-off between precision and recall.

Misclassification Results

Tweet	**Correct Label**	**Predicted Label**
RT @USER: Typhoon #Hagupit heading to #Philippines http://t.co/MkZSuymDzW Oxfam team preparing contingency stocks in case we need to respo...	caution and advice	rescue and donation efforts
RT @USER: Situation critical in rural areas near epicenter where 90% of the people have lost homes, livestock and have no way of getting foo...	infrastructure damage	injured or dead people
RT @USER: Center of Typhoon #Hagupit #RubyPH NOW moving over Dolores on Samar Island #Philippines http://t.co/qDXS82JtE0	other useful information	caution and advice

Table 5.4. Examples of misclassified tweets, the highlighted snippets are generated rationales by RACLC.

From Table 5.2, it can be seen that our model generally misclassifies about 15% of tweets and misspecifies roughly 15% of rationale tokens in case of in-domain evaluation. We observe that most classification errors are due to the mixture of information in tweets or the similarity of the information reported in different classes. Sometimes, a tweet contains information from more than one class, i.e., many tweets about caution also report rescuing or assistance activities. In this case, humans also struggle to choose a more suitable label for the tweet. We suggest that using NLP tools to fragment tweets in pre-processing step, and assigning a label for each fragment might help in this case. Besides, many tweets of caution and other useful information are misclassified to each other due to the similarity of information in tweets, i.e., the "other useful information" class gives updates on location or the current status of floods, but some tweets showing caution about the important movement of storm eye are misclassified to this class. This type of error should be solved when there are more labeled data for the observation. The low performance in the "emotional support or irrelevant" category is mainly caused by noisy tweets, i.e., tweets reporting evacuation efforts but talking about another event/situation. Examples of misclassified tweets are illustrated in Table 5.4.

Embedding Representations

In this part, we show that our contrastive learning-based approach is able to bring semantically similar tweets together, and pushes different tweets apart in embedding space. Generally, the recent BERT model and its variants can capture the contextual semantics of sentences. However, these original BERT models are not trained directly on semantic similarity problems. Therefore, the output BERT embedding representa-

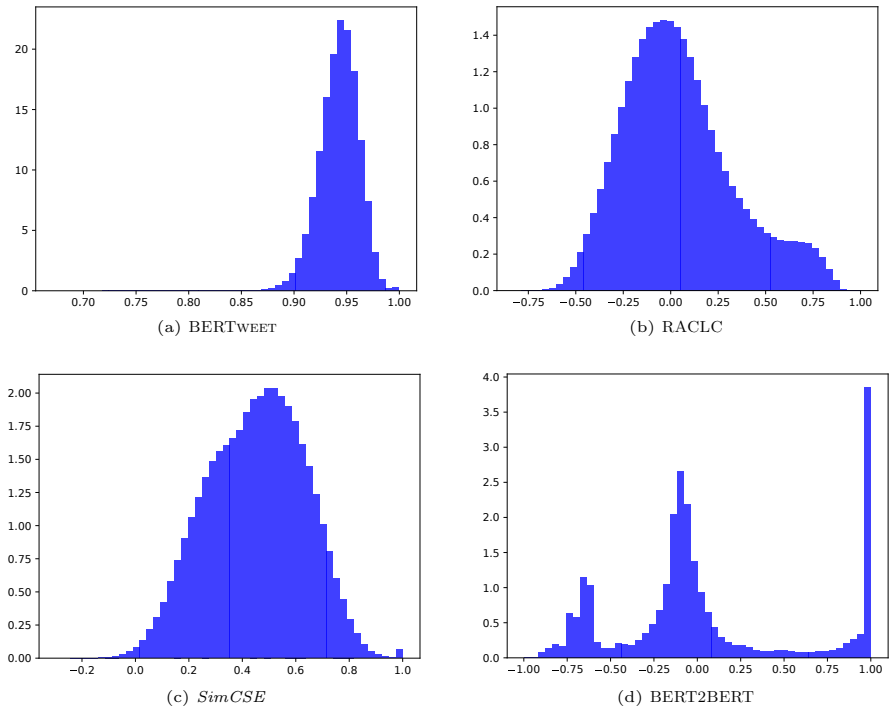


Figure 5.4. Histograms of cosine similarity between 30,000 random tweets in MEQUAKE dataset.

tions are unsuitable for unsupervised tasks such as clustering. Our RACLC model, which is a fine-tuned BERT on small labeled datasets, allows generating semantically meaningful representations of crisis-related tweets. The output embeddings can be compared using similarity metrics such as cosine, Euclidean and are suitable for semantic similarity tasks such as clustering.

Figure 5.4 shows the histogram of the cosine similarities between 30,000 random tweets from unlabeled MEQUAKE dataset using several embedding techniques such as BERTweet [99], SimCSE [42], BERT2BERT[108] and our RACLC in Figure 5.4. For RACLC and BERT2BERT, we use models trained on NEQUAKE dataset to extract representations. The RACLC embeddings are extracted from CL Head. Meanwhile, the BERTweet, SimCSE and BERT2BERT embeddings are output representation of the first [CLS] token. All the histograms tend to have normal distributions. However, BERTweet based similarity scores are skewed towards 1. Many tweets are semantically different, yet have high BERTweet similarity scores. BERT2BERT tends to give high similarity scores to tweets belonging to the same class, even if the reported information is highly different. SimCSE is a contrastive-based embedding trained on normal text datasets, our RACLC embeddings work better than SimCSE on similarity tasks with Twitter texts. Table 5.5 shows that RACLC returns a high cosine similarity score for two tweets with the same meaning,

Tweet 1: *half a million evacuate from #Nepal #earthquake* Tweet 2: *500000 people flee Nepal quake*				
BERTweet	SBERT	SimCSE	BERT2BERT	RACLC
0.969	0.655	0.797	0.998	0.972
Tweet 1: *praying for people in Nepal earthquake* Tweet 2: *When many people cares about Thirst being cancelled, how about Nepal earthquake*				
BERTweet	SBERT	SimCSE	BERT2BERT	RACLC
0.954	0.437	0.574	0.997	0.190

Table 5.5. Examples of tweet cosine similarity with different pre-trained embedding representations

though they have no common words. Besides, it returns a low similarity score for tweets that are not similar.

Figure 5.5 illustrates tweets in our labeled MEquake dataset in 2-D vector space. In Figure 5.5.a, BERTweet embeddings do not show any separation between tweets of different classes. However, when we use RACLC model trained on NEquake and generate embeddings for tweets of MEquake, tweets belonging to the same class tend to move closer to each other.

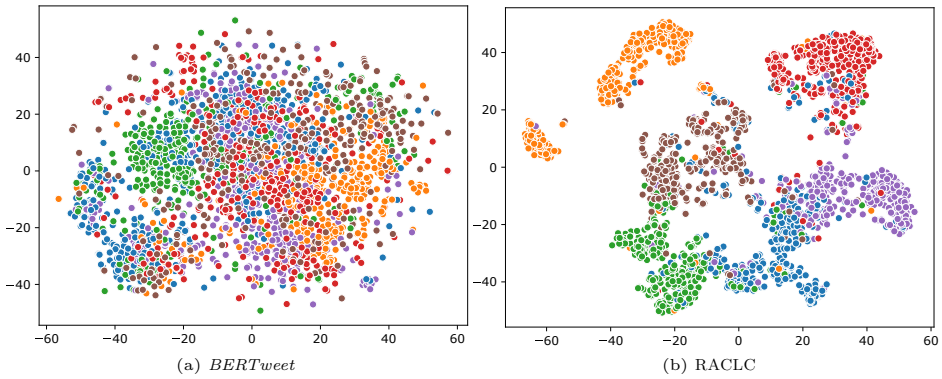


Figure 5.5. Embedding representation of MEquake tweets in 2-D vector space.

We also evaluate our embeddings on a cluster task. First, we extract tweet representations using multiple methods such as Tf-idf, BERTweet, SBERT [123], SimCSE, and RACLC. Then, we employ KMeans model to cluster tweets of MEquake, CPam into six clusters and use labeled data in Section 5.2 for evaluation. *Note that we employ models trained on* NEquake *and* THagupit *to extract embeddings on the two corresponding datasets.* Table 5.6 shows the clustering results. RACLC obtains a good separation of class labels, which is not much worse than the supervised classification results in Table 5.2.

Model	MEQUAKE			CPAM		
	Macro F1	Purity	NMI	Macro F1	Purity	NMI
Tf-idf	0.258	0.352	0.157	0.333	0.379	0.164
BERTWEET	0.257	0.299	0.089	0.372	0.384	0.177
SBERT	0.394	0.408	0.229	0.412	0.446	0.216
SimCSE	0.397	0.415	0.239	0.461	0.465	0.289
RACLC	**0.794**	**0.800**	**0.628**	**0.828**	**0.830**	**0.651**

Table 5.6. Clustering performance

5.5 Summarization experiments and results

5.5.1 Baseline models

The below disaster-specific and deep learning-based methods are considered as our summarization baselines.

- **COWTS** [129]: An unsupervised, ILP-based extractive crisis-related tweet summarization model.
- **APSAL** [63]: A clustering-based extractive summarizer of disaster-related news articles.
- **TSum4Act** [101]: A PageRank-based method for extractive summarization of Twitter disaster events.
- **MOO** [133]: An extractive summarization approach for disaster events on Twitter that jointly optimizes multiple objectives.
- **PACSUM** [176]: A recent unsupervised approach for summarization of news articles.
- **RATSUM** [108]: A crisis-related tweet summarization approach that does not have a contrastive learning setup and uses a word-overlap based deduplication strategy.
- **RATSUM_TG**: A RATSUM variant with temporal graph based deduplication. Tweet representations are fine-tuned embeddings of the first [CLS] token.
- **RACLTS_W**: Our RACLTS variant with rationales are extracted from RACLC model. The temporal-based deduplication is based on word overlap.

Model	ROUGE-1 F-score															
	NEquake				MEquake				THagupit				CPam			
	Infrastructure		Injuries/death		Infrastructure		Injuries/death		Infrastructure		Affected people		Infrastructure		Affected people	
	25/04	26/04	25/04	26/04	20/09	21/09	20/09	21/09	06/12	07/12	06/12	07/12	15/03	16/03	15/03	16/03
APSAL	0.421	0.325	0.368	0.353	0.436	0.414	0.501	0.444	0.381	0.396	0.376	0.359	0.366	0.441	0.459	0.426
TSum4Act	0.446	0.383	0.380	0.273	0.394	0.435	0.478	0.504	0.438	0.300	0.382	0.447	0.422	0.401	0.424	0.389
MOO	0.336	0.239	0.277	0.329	0.389	0.398	0.601	0.358	0.278	0.265	0.310	0.302	0.254	0.396	0.323	0.318
PACSUM	0.447	0.421	0.462	0.418	0.569	0.644	0.52	0.615	0.432	0.422	0.400	0.374	**0.592**	0.475	0.428	0.435
COWTS	0.447	0.387	**0.524**	0.425	0.621	0.631	0.638	0.667	0.419	0.614	0.587	0.484	0.565	0.477	0.472	0.549
RATSUM	0.501	0.431	0.448	0.519	0.602	0.607	0.610	**0.701**	0.514	0.624	0.701	0.585	0.504	**0.487**	0.469	0.539
RATSUM_TG	0.374	0.368	0.444	0.377	0.394	0.434	0.397	0.575	0.407	0.367	0.379	0.38	0.446	0.4	0.334	0.418
RACLTS_W	**0.539**	**0.478**	0.499	**0.525**	**0.629**	**0.720**	0.607	0.660	0.538	**0.639**	0.732	**0.597**	0.513	0.451	**0.504**	0.534
RACLTS	**0.539**	**0.478**	0.504	0.520	**0.629**	**0.720**	**0.646**	0.678	**0.544**	**0.639**	**0.735**	0.592	0.523	0.439	0.494	**0.575**

Table 5.7. Summarization results. The highest and second highest results are in bold and brown, respectively.

5.5.2 Groundtruth summaries

We ask five volunteers to prepare class-level groundtruth summaries. First, we generate 1000-word summaries by our method and all the baseline models. Tweets are pooled together, exact duplicates are removed. Then, we give this set of tweets to users and ask them to pick tweets and prepare 200-word summaries. The final summary is formed by gradually selecting tweets voted by most users until we reach the length limit of 200 words. In total, we have to prepare 8 summaries (4 classes x 2 days) for each event.

5.5.3 Evaluation Metrics

Our summarization results are evaluated using ROUGE toolkit [74]. We report ROUGE-1 F score of all summarization methods.

5.5.4 Results

Table 5.7 shows ROUGE-1 F scores of 8 summary instances extracted by all summarization models. We do not show the results of the other summaries in the table due to the space limit, but they generally obtain similar patterns. It is clearly seen that RACLTS performs the best in the majority of cases. On average RACLTS performs equally well as RACLTS_W and better than RATSUM, COWTS, PACSUM by 1.4%, 4.5%, 6% and the remaining methods by large margin (>8%).

Generally, due to the advances of fine-tuned embeddings, PACSUM can perform well in some cases. The model was proposed for news articles summarization, so it performs worse than RACLTS and some other crisis-related summarizers such as COWTS and RATSUM. MOO tends to return long and redundant summaries. The original MOO paper also points out that drawback. APSAL is designed for

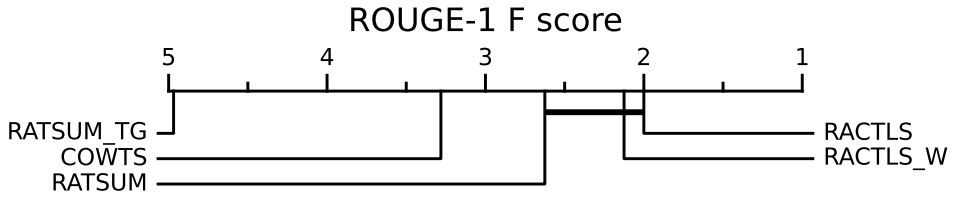


Figure 5.6. Pairwise Friedman test. Models with lower ranks have better performance. Models grouped by a thick horizontal line show insignificantly different (p-value>0.05).

summarization of new articles during crisis events. It relies on some specific features of articles that can be missing in short, noisy tweets. TSum4Act clusters tweets into sub-topics and select the same number of informative tweets in each sub-topic for summaries. However, some sub-topics might be more important than others during crisis events.

For the best models such as COWTS, RATSUM and RACLTS, we compute a pairwise Friedman significance test [38] of ROUGE-1 F scores on all the datasets. The result is illustrated in Figure 5.6. All these models employ the ILP optimization approach, yet in different setups. By using rationales for optimization, RACLTS and RATSUM perform better than COWTS with 95% confidence interval (p-value < 0.05). RACLTS and its variant RACLTS_W outperforms RATSUM, though the statistical test does not show significant difference. It means that our rationales extracted by the contrastive-based approach slightly help improve the performance compared to the rationales extracted by RATSUM. Besides, the use of temporal graph-based deduplication results in slightly better performance of RACLTS compared to its variant RACLTS_W. In contrast to this, RATSUM_TG with temporal graph-based deduplication returns much worse performance than RATSUM. It illustrates the positive influence of our contrastive-based embeddings.

Computational complexity analysis

As shown in the previous part, ILP based optimization approaches such as COWTS, RATSUM, RACLTS obtain the best performance. We now compare the complexity of these methods. All the models run on the same CPU machine under no other load condition. Figure 5.7 shows the average computation time and average number of words to be optimized for each summary. Our RACLTS optimizes significantly less number of words than COWTS, RATSUM and RACLTS_W by 65%, 45% and 38% respectively. Therefore, RACLTS is much faster than the other methods. Specifically, the average running time of COWTS, RATSUM and RACLTS_W are 2.0, 1.6 and 1.5 times faster than RACLTS.

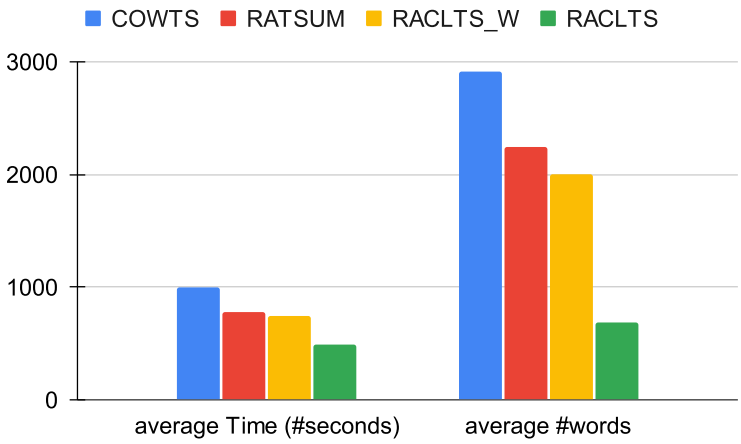


Figure 5.7. Average summarization time and average number of words to be optimized by summarization methods.

5.6 Application of RACLC in detection of actionable phrases

In this section, we observe how well our RACLC can identify actionale phrases in actionale tweets. Recently, TREC-IS track [90] has made available a large set of crisis-related messages. Around 20,000 tweets have been manually annotated into 25 information types and 4 priority levels. Among the labels, six information types are identified as "actionable". We download actionable tweets that are of typhoon and earthquake events. As our objective is to test the zero-shot setup, we remove all the tweets that belong to four events that we have used to train our model and obtain 3466 tweets. Among them, 177 and 1146 tweets are annotated with 'critical' and 'high' priority labels, respectively. We randomly sample 50 tweets with high or critical priority and employ RACLC to predict class labels of tweets and extract rationales. We define the rationales of actionable tweets as *actionable phrases*.

The tweets are given to five users for a study as follows:

- We evaluate **to what extent users agree with our extracted actionable phrases**: For the first twenty-five tweets, users are asked to select the label of each tweet from a checklist and judge whether the extracted rationales report short and actionable information for the chosen label. If the extracted rationales are redundant or too short, users will help us to rewrite them.

- We evaluate **how good users can understand machine behavior in predicting actionable phrases**: For the last twenty-five tweets, we ask users to identify class labels and extract actionable snippets for the chosen labels (this time, machine-extracted actionable snippets are not provided).

Table 5.8 shows the average agreement between user annotations and machine-

	Macro F1	Token-F1
First 25 tweets	0.888	0.906
Last 25 tweets	0.873	0.741

Table 5.8. Agreement between user annotations and machine-generated labels.

generated labels. Overall, in 88% of cases, users and RACLC choose the same class labels. For 25 tweets, when rationales are given, users highly agree that the extracted rationales report all important/actionable information for the class labels. For the last 25 tweets, when users are asked to choose class labels and write the rationales, the average Token-F1 is 74%. We observe that this low agreement is mainly due to the difference in extracted rationales in the case of tweets with multiple information. Users struggle to choose the class label when a tweet reports multiple information. For example, '*New cyclone #kills 3 in #Mozambique; #UN warns of flooding - Apr 26 @ 10:23 AM ET URL*', our machine assign the tweet to 'affected people and evacuation' and extract '*#kills 3*' as the rationales. However, some users select the label 'caution and advice' and extract '*#UN warns of flooding*' as actionable phrases. The mismatch in labels of a tweet leads to completely different extracted rationales and low Token-F1. This issue was also mentioned in Section 5.4.4.

5.7 Chapter Summary

This chapter introduces a rationale-aware contrastive learning-based classification and summarization framework of crisis events from microblogs. Rationales play quite a significant role in both phases. RACLC shows that trustworthiness in prediction may be achieved without significant compromise in performance. Subsequently, our summarizer RACLTS also gets the benefits of the identified rationales to generate informative summaries with low computational complexity. Our RACLC model can be helpful for TREC-IS [90] tracks. The model helps identify meaningful and actionable phrases in tweets and brings more comprehensive research with TREC-IS tasks. In the next chapter, we develop a semi-supervised approach to learning faithful attention-based explanations for the classification of tweets from limited rationale annotations.

6

Semi-Supervised Attention-based Interpretable Classification of Crisis Events

6.1 Introduction

In previous chapters, we employ supervised approaches for interpretable classification of tweets in to into different humanitarian classes during crisis events. Our methods identify the class information and words/phrases responsible for determining that class. We asked humans to provide explanation tokens, so-called rationales, along with class-level annotations for training. For example, the tweet "*RT @USER:* ***Three people from #Taiwan died*** *in #MexicoEarthquake, Chinese embassy in Mexico confirms https://t.co/2Ig19YnCbs*" is labeled as 'injuries or death' and words in bold are annotated as rationales. While these approaches show a promising direction toward interpretable crisis systems, human-level annotation also adds a bottleneck toward the scalability of the method and its application toward new unseen events. On the other hand, some sets of approaches tried to use attention weights as a mode of explanation [12, 44]. However, recent studies pointed out the flaws in considering attention weights as a proxy for explanation [59, 165]. The debate is still ongoing [18]. This brings two open challenges into the framework — (a). How could we learn faithful attention weights that could represent explanations with high confidence, and (ii). How to develop interpretable models under the given human budget, i.e., limited annotated data.

In this chapter, we try to address the above-mentioned challenges and design a two-stage framework that exploits the power of semi-supervised learning. Next, we incorporate the distance between attention weights and predicted probabilities of rationales into our customized loss function to make the attention weights faithful. Evaluation on four different disaster events shows that this would help to alleviate almost 50% human annotation budget and learn faithful attention weights. Further, we extend the idea of zero-shot learning to directly transfer the knowledge acquired in the humanitarian classification model to the actionable tweet detection problem.

Move People
Authorities urge residents to seek higher ground immediately as rains lash the north after Cyclone Kenneth #mozambique #naturaldisaster https://t.co/nSFnEQDdsA https://t.co/3B84ljp8oY

Emerging Threat
READ: Families in #Mozambique face new floods and fight for their lives. Nicholas Finney comments on #CycloneKenneth via @MailOnline: 'we have grave fears for the thousands of families currently taking shelter. https://t.co/V68H9RJixX

Figure 6.1. Example of actionable tweets, actionable labels are in bold, rationales are in blue.

Results suggest that if the source and target tasks are related and from a similar application area (e.g., crisis), such zero-shot learning setup can be directly applied to the target task. This direct transfer helps to identify the actionable classes and related rationale tokens from the tweets. Examples of actionable tweets, class labels, and rationale snippets are illustrated in Figure 6.1.

6.2 Methodology

This section describes the detailed architecture of our faithful attention-based classification model.

6.2.1 Problem Formulation

Given a small set of tweets $T = \{twt_1, twt_2, .., twt_m\}$, along with labels $L = \{l_1, l_2, .., l_m\}$, $l_i \in C$, where C is the set of humanitarian classes (i.e., infrastructure damage, affected people, rescue, etc.). We assume that we also have access to human rationales for a small set of tweets $S = \{twt_i\} \subset T$. Rationales are short snippets from original texts that are marked as having supported the class label. Here, we consider each tweet twt as a list of words $twt = \{w_1, w_2, .., w_k\}$. If the tweet twt is provided with human rationales, we then have labels $y = \{y_1, y_2, .., y_k\}$ assigned for every word, $y_i \in \{0, 1\}$ specifies whether a word is a part of rationales ($y_i = 1$). Our aim is to take the limited human rationales as little supervision to design a Faithful Attention-based Classification model (FAC-BERT) of tweets during crisis events.

6.2.2 Overview

As discussed above, our goal is to develop an interpretable classification approach with little supervision of human rationales. Many previous works [59, 165, 137] have argued that attention is not explanation. Hence, we also aim at finding a way to

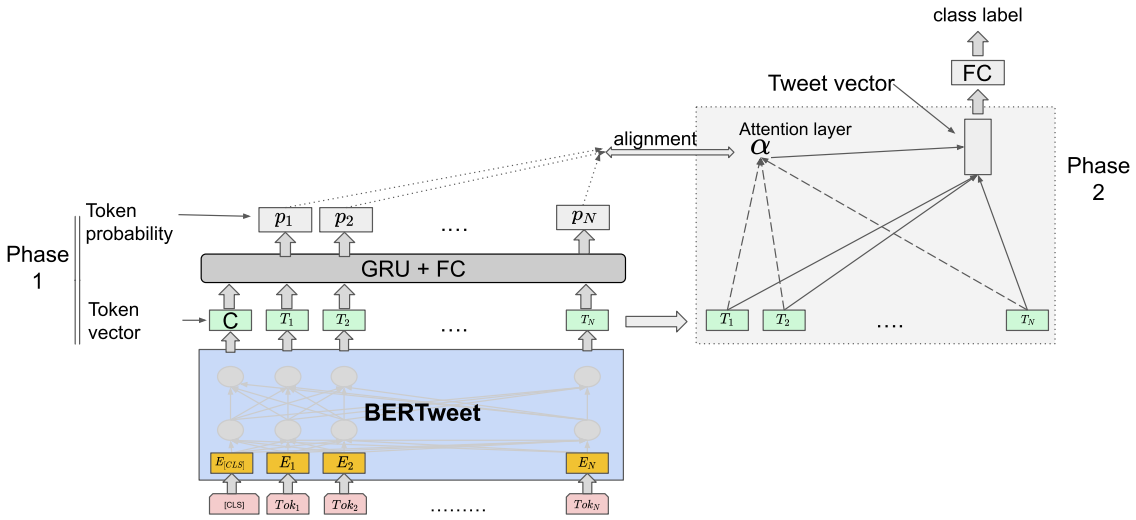


Figure 6.2. FAC-BERT - Our faithful attention-based classification model

make the attention become a faithful explanation. Our model learns a mapping from the annotated rationales to machine attention. We achieve this by proposing a hierarchical learning structure that predicts the probabilities of each word being rationales. Then we align these probabilities with attention weights to inform tweet classification.

As a first step, we apply BERTweet [99] to tokenize input tweets and generate token embeddings. These embeddings are fine-tuned on a token classification task that predicts whether a token is a part of rationales with probabilities. These values are used to guide machine attention. Then, we apply a weighted sum of token vectors to obtain tweet vectors for tweet-level classification. The weights are learned to reflect the importance of each token to the output decision. Our model is able to obtain high classification performance and faithful attention. We refer the model as Faithful Attention-based BERTweet Classification (FAC-BERT). The detailed training process of our FAC-BERT classifier is described below.

6.2.3 Model architecture

Figure 6.2 illustrates the architecture of our FAC-BERT model. It consists of two training phases with a shared BERTweet encoder. Note that our approach is different from muti-task learning setups in some previous work [108, 107]. Our two phases are not trained simultaneously. The second phase takes the information and last checkpoint from the first phase and continues to train its own task.

BERTweet encoder [99]. We use BERTweet as a shared encoder for our learning phases. BERTweet is a language model pre-trained on a large-scale dataset of English tweets. First, each tweet is tokenized into tokens of the form $[CLS]tok_1tok_2..tok_n$,

where $[CLS]$ is a special symbol added in front of every input instance. An unknown word from BERTweet vocabulary can be split into several tokens. Input sequences are padded to the same length, which is the maximum length of tweets in each learning batch. Then, we feed the tokenized data to the BERTweet encoder and obtain token embeddings of size 768 dimensions x_{tok}^{ij} for each token tok_i in tweet twt_j. The token representations are fine-tuned in the first learning phase and then aggregated to form tweet representation for classification in the second training phase.

Token-level training (Phase 1). This step takes token embeddings as inputs and trains a binary classifier to predict which tokens are part of rationales. We append a GRU (Gated Recurrent Unit) followed by a fully connected layer with a Sigmoid function on top of BERTweet token embeddings. Initially, rationale labels are assigned at the word level. To train our model, we map labels to token level, where each tokenized token has the same label as its original word. Later, at the evaluation step, we retrieve word-level labels by applying max pooling on token labels. We employ the binary cross-entropy loss function for token-level classification.

$$Loss_{tok} = BCELoss(y_i, p_i) \tag{6.1}$$

where y_i is the true label, p_i is the predicted probability of token tok_i to be rationale. Recall that we aim at learning with little supervision of human rationales. Hence, the above loss function is only averaged over k% of tweets in training set with human rationales. After the training completes, we obtain fine-tuned token embeddings and the probabilities of tokens to be rationales for all tweets in the training set.

Tweet-level training (Phase 2). This step predicts the class label of input tweets. Token vectors are summed up to obtain tweet representations. In our FAC-BERT, the sum of token vectors is the attention-based weighted sum. Specifically, we apply an attention layer [9] on top of fine-tuned BERTweet token embeddings. The attention weights α_{ij} are computed by a softmax function as follow:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{|N|} exp(e_{ik})} \tag{6.2}$$

Where e_{ij} is the output score of a feedforward neural network model [9], which captures the alignment between input at position j and output i, —N— is the length of the considering tweet. The representation of each tweet twt_i is then the weighted sum over token embeddings:

$$x_{twt}^{i} = \sum_{j=1}^{|N|} \alpha_{ij} x_{tok}^{j} \tag{6.3}$$

The tweet embeddings are fed into a fully connected softmax layer to predict class labels. Besides, we want attention weights to mimic human rationales so that the

attention accurately reflects the true reasoning behind a prediction (i.e., tokens with high attentions highly influence the model decision). Hence, we minimize the distance between attention weights α_j in a tweet twt and probabilities p_j of tokens to be a rationale that is learned in the first phase:

$$d(\alpha_j, p_j) = max(0, 1 - cosine(\alpha_j, p_j)) \tag{6.4}$$

The above distance is interpolated with the weighted cross-entropy classification loss to form the final loss function of the tweet-level training step:

$$Loss_c = -\sum_{l=1}^{|L|} w_j * y_{jl} log(p_{jl}) + \lambda \sum_{i=1}^{|N|} d(\alpha_i, p_i) \tag{6.5}$$

where $|L|$ is the number of unique class labels, y_{jl} and p_{jl} are the true label and predicted value of tweet twt_j having class label l, w_j is the inverse weighted probability of label occurrence in the dataset. In case of a balanced dataset, w_j is set to 1 for all classes. —N— is the token length of the tweet.

When training the class label prediction task, we fix parameters of top layers (GRU+FC) at the token-level training phase.

6.3 Experimental Setup

6.3.1 Datasets

We consider four natural disasters, which are Nepal Earthquake (NEQUAKE), Mexico Earthquake (MEXQUAKE), Typhoon Hagupit (THAGUPIT), Cyclone PAM (CPAM). Each dataset contains about 2000 tweets with humanitarian classes and rationales. The details of datasets and annotations is described in Chapter 5. The size of the datasets is shown in Table 5.1 of the previous chapter.

6.3.2 Baseline methods

We compare our model with the following classification models, which include both typical classification approaches and our proposed interpretable crisis-related classification models in Chapter 4 and Chapter 5.

- SVM: An effective classification baseline for classification of crisis events [23, 56].
- Robust-CNN [98]: A Convolutional Neural Network based approach with pre-trained word embeddings for classification of crisis events.
- BERTweet [99]: BERTweet with a linear classification on top of the first [CLS] token embedding.

- LCL [146]: A classification model that relies on label-aware contrastive loss.
- BERT2BERT [108]: An interpretable by design approach for classification of crisis events. The model employs a multi-task learning strategy to train and predict class labels and rationales simultaneously. BERT2BERT(-2stg) is the variant BERT2BERT, which is not interpretable by design.
- RACLC [107]: A contrastive learning-based approach for classification of crisis events. It applies a contrastive multi-task learning approach to boost the performance of class label and rationale prediction tasks. RACLC(-2stg) is a variant of RACLC, which is not interpretable by design.

6.3.3 Evaluation Metrics

Groundtruth based evaluation

We evaluate how good our predicted class labels and rationales are compared to human annotations. For classification performance, we measure Macro-F1 score. Similarly, we measure the agreement between extracted rationales and human rationales using Token-F1 metric. First, Token-precision is computed to show the fraction of relevant rationale words among all predicted rationales. Next, Token-recall measures the fraction of correctly extracted rationale words among the total number of human rationale words. Then, we combine the two scores by taking their harmonic mean Token-F1.

Model Faithfulness

One might argue that a model can have a high agreement with human rationales (plausibility), but does not reflect the true internal reasoning. Similar to previous chapters, measure to what extent the extracted rationales influence the model decision by using the following metrics.

Comprehensiveness [36]. This metric measures how much the classification performance drops when extracted rationales are removed/masked from the original inputs. Given X, R and $X \backslash R$ are original examples, predicted rationales (non-rationales are marked by '*'), and predicted non-rationales (rationales are marked by '*'), respectively. We compute comprehensiveness score as follows.

$$\textit{Comprehensiveness} = \textit{Macro-F1}(X) - \textit{Macro-F1}(X \setminus R)$$

The higher comprehensiveness shows the high influence of the predicted rationales on the classification performance.

Sufficiency [36]. This metric evaluates performance differences when using only rationales and the original input texts.

Sufficiency = Macro-F1(X) - Macro-F1(R)

The lower sufficiency is better since it shows that only predicted rationales are sufficient for a model to make predictions.

6.3.4 Model Details and Hyperparameters

We evaluate our model and all the baselines using a 5-fold cross-validation setup. At each run, we apply a stratified sampling method to obtain train/valid/test sets with ratios 70%/15%/15% respectively. All the baseline models are run with configurations from original papers. To train our FAC-BERT, we pre-process data by converting tweets to lowercase and removing mentions, URLs. Our method is trained for 10 epochs, and the batch size is 16. The GRU layer has a hidden size of 128. We optimize the model using AdamW optimizer [84] with a learning rate of 2e-5. Besides, we specify a list of candidates for the hyper-parameter λ and select the one that obtains consistently good performance (average Macro-F1 and Token-F1) with a 5-fold setting on validation sets. After fine-tuning, we set $\lambda = 0.5$ for all the datasets since it generally performs the best in the majority of cases across different validation runs. Another hyperparameter is k, i.e., the percentage of human-annotated rationales required to successfully train the model. We set $k = 50\%$ to compare performance with other baseline models. Further, we also observe FAC-BERT performance with varying k.

6.4 Classification Results

This section presents the performance of our proposed approach. We consider both in-domain and cross-domain evaluation. In in-domain classification, training and testing data come from the same event. For cross-domain evaluation, we train the model on one dataset and apply it to another dataset of the same event type. For example, the model trained on NEquake dataset is used to predict class labels and human rationales on MexQuake dataset. Recall that FAC-BERT consists of two-phase learning. In the first phase ($p1$), it predicts binary labels for tokens, i.e., whether a token is a rationale or not. The second phase ($p2$) learns faithful attention weights, it does not predict any binary classification of tokens (rationale/not rationale). To evaluate Token-F1 of the second phase, we extract the same number of rationale tokens as in the first phase prediction. Note that the objective of this paper is not to improve class labels or human rationale prediction tasks. Rather, we want to answer the following two questions:

1. How well our FAC-BERT performs under limited human supervision?
2. How to learn faithful machine attention?

Model	In-domain							
	NEQUAKE		MEXQUAKE		THAGUPIT		CPAM	
	Macro-F1	Token-F1	Macro-F1	Token-F1	Macro-F1	Token-F1	Macro-F1	Token F1
SVM	0.799	-	0.738	-	0.802	-	0.768	-
Robust-CNN	0.833	-	0.787	-	0.817	-	0.843	-
LCL	0.865	-	0.850	-	0.856	-	0.864	-
BERTweet	0.864	-	0.851	-	0.852	-	0.888	-
BERT2BERT(-2stg)	0.874	0.857	0.855	0.826	0.857	0.820	0.891	0.868
BERT2BERT	0.862		0.836		0.847		0.861	
RACLC(-2stg)	**0.890**	**0.868**	**0.869**	**0.874**	**0.865**	**0.847**	**0.896**	**0.893**
RACLC	0.869		0.842		0.845		0.871	
FAC-BERT-50%-$p1$	0.876	0.848	0.851	0.845	0.853	0.826	0.871	0.873
FAC-BERT-50%-$p2$		0.850		0.844		0.822		0.869

Table 6.1. In-domain evaluation. - if a model does not extract rationales

In the following sections, FAC-BERT-k%-$p1$ and FAC-BERT-k%-$p2$ are used to indicate the performance of FAC-BERT with rationales extracted from the *first phase* and *second phase* respectively. The value k specifies the percentage of human rationales used during the training process.

6.4.1 In-domain evaluation

We evaluate the performance of classification models on the test set of the same event on which the models are trained on. Table 6.1 compares the performance between different classification models. We show the prediction results of FAC-BERT using $k = 50\%$ human annotated rationales. FAC-BERT achieves competitively equal Macro-F1 with other baselines such as BERT2BERT or RACLC. Compared to the best Token-F1 returned by RACLC [107] with 100% human rationale supervision, FAC-BERT that uses 50% human rationales only get a slight drop (i.e., $< 3\%$). It is also interesting that adding the regularization part in the loss function of phase 2 helps to align the rationale tokens learned in phase 2 with phase 1. That's why we get almost similar Token-F1 scores between the two phases.

6.4.2 Cross-domain evaluation

This section evaluates the classification performance when the prediction is made on a similar event dataset that was not used for training. We compare the cross-domain performance between FAC-BERT using 50% rationale supervision with baseline methods in Table 6.2. The Macro-F1 is equal to or slightly worse than some other baselines. Both the two phases of FAC-BERT return similar Token-F1 values. It is observed that FAC-BERT-50%-$p2$ got 6%, 1.6%, 2.8% and 2% drops than the best Token-F1 (RACLC) on NEQUAKE, MEXQUAKE, THAGUPIT and CPAM respectively.

	Cross-domain (Train//Test)							
	MexQuake//NEquake		NEquake//MexQuake		CPam//THagupit		THagupit//CPam	
	Macro-F1	Token-F1	Macro-F1	Token-F1	Macro-F1	Token-F1	Macro-F1	Token F1
SVM	0.661	-	0.679	-	0.523	-	0.524	-
Robust-CNN	0.730	-	0.683	-	0.671	-	0.602	-
LCL	0.849	-	0.835	-	0.819	-	0.800	-
BERTweet	0.851	-	0.837	-	0.822	-	0.853	-
BERT2BERT(-2stg)	0.847	0.839	0.841	**0.862**	0.808	0.831	0.851	**0.873**
BERT2BERT	0.842		0.829		0.815		0.818	
RACLC(-2stg)	**0.855**	**0.851**	**0.849**	**0.862**	**0.829**	**0.833**	**0.858**	0.867
RACLC	0.850		0.832		0.819		0.813	
FAC-BERT-50%-$p1$	0.834	0.794	0.826	0.844	0.794	0.806	0.832	0.854
FAC-BERT-50%-$p2$		0.791		0.846		0.805		0.853

Table 6.2. Cross-domain evaluation. - if a model does not extract rationales

6.4.3 How does performance of FAC-BERT vary with budgeted human rationales (k)

Table 6.1 and Table 6.2 show the performance of FAC-BERT with $k = 50\%$ human annotated rationales. In this section, we would like to explore the variation in model performance under different human budgets.

Variation in in-domain scenario: Figure 6.3 shows the Token-F1 scores of our second phase prediction with varying percentages of human rationales. Interestingly, when we use only 10% human rationale labels, we obtain pretty good performance (i.e., 83.1% Token-F1 on CPam dataset). The Token-F1 increases significantly when we vary the percentage of human rationales from 10% to 50%. Then, the Token-F1 gets improved slowly when more human rationales are added. The result shows that by using 10% or around 200 instances with human rationales, our FAC-BERT can obtain more than 75% Token-F1 for all the datasets. Besides, FAC-BERT obtains 80% Token-F1 for all the datasets when 20% or around 400 instances with human rationale supervision are used for training. Unlike previous chapters which we focus on improve the model performance-interpretability tradeoff, this evaluation gives a guideline on how much rationale data is needed to train a good interpretable classification model on crisis domain.

Varying the human rationales doesn't harm the performance of the humanitarian class label prediction task. FAC-BERT obtains similar Macro-F1 score as shown in Table 6.1 with different k% human rationales. Side by side, the Token-F1 performance also reaches a quite stable point with 50% human rationale labels. The gain is not significant beyond this point, and results only slightly improve when more human rationales are added.

Variation in cross-domain scenario: Similar to the in-domain scenario, we feed FAC-BERT with an increasing percentage of human rationales for supervision and observe the performance change in cross-domain. Figure 6.4 illustrates Token-F1

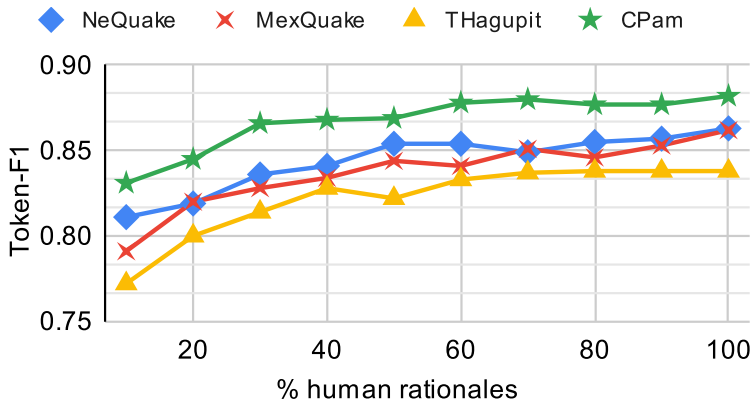


Figure 6.3. In-domain evaluation with various percentages of human rationales

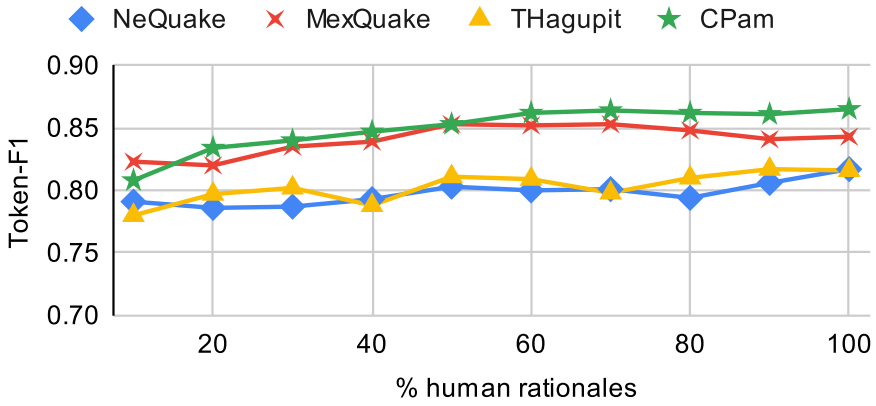


Figure 6.4. Cross-domain evaluation with various percentages of human rationales

values extracted from the second phase of FAC-BERT. Using 10% human rationales archives more than 75% Token-F1. The rationale prediction results improve when more human rationales are added, but not as significant as in case of in-domain evaluation. Starting from 50%, adding more human rationales slightly boost the Token-F1.

6.4.4 Influence of the alignment between human rationales and machine attention

So far, we observe the variation in performance under different annotated rationale budgets. In this section, our objective is to learn the role of alignment regularizer in the loss function (Eqn. 6.5). We observe how our loss function with the alignment between rationale prediction and attention weight helps to improve the faithfulness of machine attention. First, we take predicted rationales from the second phase learning (attention weight-based rationales), namely FAC-BERT-p2, with % human rationales vary in range $k \in [10, 20, .., 100]$. This is done for both cases, which are with

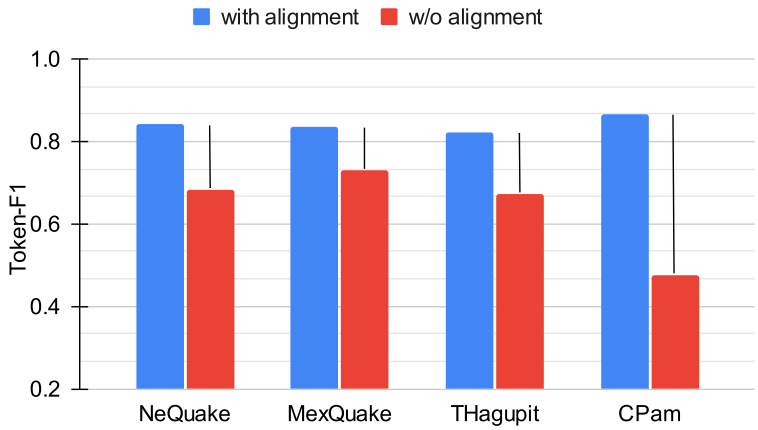


Figure 6.5. In-domain average Token-F1 with/without weights alignment in the loss function. Vertical black lines indicate drops in the performance/Token-F1.

and without distance alignment between attention weight and rationale probability in FAC-BERT loss function. Here, we report the average result over different k values. Figure 6.5 illustrates the impact in in-domain evaluation. We also obtain similar patterns in cross-domain evaluation. When there is no attention alignment in the loss function ($\lambda = 0$ in Section 6.2), the average Token-F1 score returned by FAC-BERT-p2 decreases significantly. More specifically, it drops by 10.9%, 14.5%, 16.2% and 39.3% on MEXQUAKE, THAGUPIT, NEQUAKE and CPAM, respectively, compared to ones using attention alignment. Besides, the prediction of FAC-BERT-p2 without attention alignment varies greatly across datasets. It predicts rationales poorly on CPAM dataset with an average of 47.4% Token-F1. We observe that without alignment, the standard attention might give high attention weights to unimportant words that are not supportive evidence for output labels. As an example, for the tweet "*50k children at risk in #vanuatu* ***after*** *devastation of* ***#cyclonepam .*** *please* ***help*** *respond . . .*", FAC-BERT-p2 *without attention alignment* correctly predicts the tweet as "affected people & evacuation"; however, it assigns the highest weights to words in bold. By using our regularized loss, the model reassigns the highest weights to the following bold words "***50k children at risk*** *in #vanuatu after devastation of #cyclonepam please help respond . . .*".

6.4.5 Model Faithfulness

We evaluate whether the extracted rationales can be seen as explanations for the output decision of FAC-BERT. There is a high overlap between the rationales obtained in two phases ($p1$ and $p2$). For brevity, we only show the comprehensive and sufficiency of predicted rationales from the second phase ($p2$), which is based on attention weights ($\lambda = 0.5, k = 50\%$). The scores are computed when we learn rationales from 50% rationale supervision. Generally, FAC-BERT obtains high comprehensiveness

Dataset	Comprehensiveness ↑		Sufficiency↓	
	RACLC	FAC-BERT	RACLC	FAC-BERT
NEquake	0.352	0.378	-0.005	0.017
MexQuake	0.259	0.365	0.009	0.025
THagupit	0.349	0.265	0.007	0.018
CPam	0.403	0.352	0.017	0.002

Table 6.3. Comprehensiveness and Sufficiency

scores for all datasets. This illustrates the huge drop in Macro-F1 when the predicted rationales are removed from the original inputs. In other words, the predicted rationales are important for FAC-BERT to make decisions. Besides, the low sufficiency of FAC-BERT indicates that the predicted rationales alone are sufficient for FAC-BERT to classify tweets. Compared to the best classification model RACLC, which uses 100% human rationales, FAC-BERT obtains better comprehensiveness on two earthquake datasets. However, FAC-BERT has higher sufficiency than RACLC, but the difference is only less than 2%. By using 50% human rationale supervision, FAC-BERT attention is competitively faithful compared to RACLC using 100% rationale supervision.

6.5 Application of FAC-BERT in detection of actionable tweets

In this section, our objective is to explore the power of transfer learning over the related tasks of the same application area. In Section 6.4, we observed that FAC-BERT gives promising results under a given amount of annotated rationale data. This section tries to answer the question, "what would happen if we deploy the humanitarian classification model over actionable tweet detection?".

6.5.1 Data Collection

The recent Text Retrieval Conference (TREC) Incident Streams track [22] has released datasets for the classification of crisis-related tweets into fine-grained information types. Besides, the track identifies six actionable information types: Requests for Goods/Services, Requests for Search and Rescue, Calls to Action for Moving People, Reports of Emerging Threats, Reports of Significant Event Changes and Reports of Services becoming available. An 'actionable' tweet contains crucial information and immediate alert that might be useful for individuals and stakeholders to pay more attention. Those actionable information types/classes are the most difficult ones for classification models to predict due to the scarcity of labeled data.

We consider all tweets of earthquake or typhoon events from TRECIS 2021 training data [22]. Actionable tweets that belong to no more than one actionable class are selected. This set is quite small in number, which consists of 10% of the collected data. There are six actionable classes in the dataset. Apart from that, we randomly sample 100 tweets that do not contain any actionable labels for each event type and filter out the other tweets from our dataset. The details of the collected dataset are shown in Table 6.4. The last column shows the size of each class in our final actionable dataset. Generally, the dataset is quite imbalanced, classes such as 'EmergingThreats' or 'ServiceAvailable' have more tweets. Meanwhile, only a few tweets report information about 'MovePeople' or 'GoodsService'. This imbalance poses a challenge for classification models.

6.5.2 Actionable tweet classification using FAC-BERT

In this section, we study the application of our proposed model FAC-BERT in two aspects:

1. How well FAC-BERT is able to extract actionable snippets from actionable tweets?
2. How well our proposed FAC-BERT performs on a new dataset with a new problem setup?

The major bottleneck that hinders the direct application of FAC-BERT over actionable tweet classification is the nonavailability of human-annotated rationales. TREC-IS does not have rationale snippets of tweets. Hence, to answer the first question, we apply the idea of transfer learning, i.e., directly apply FAC-BERT-$p1$ on the actionable tweets to gather the rationales. We train the model using the 100% rationale dataset provided for the humanitarian class identification problem. We trained two different models for two different events (earthquake and typhoon), i.e., NEquake and MexQuake datasets are used to train (FAC-BERT-100%-$p1$) and extract the rationale snippets from the actionable tweets of an earthquake event. Similarly, typhoon datasets (THagupit and CPam) are used to train and extract rationale snippets from the actionable tweets of typhoon category. The two phases of FAC-BERT obtain similar rationale prediction performance; hence, we just simply predict rationales on the actionable dataset from the first FAC-BERT learning phase (FAC-BERT-$p1$).

Now, we have the actionable class labels of tweets, and rationale snippets for each of the tweets gathered using FAC-BERT-$p1$ trained on humanitarian class-related rationales. Next, we directly follow the model architecture (described in Section 6.2) to detect the class of actionable tweets and learn faithful attention-based rationale snippets. As we obtained the rationales through transfer learning, we used 100% of the rationale data in phase 1 and tried to align attention weights in phase 2. FAC-BERT will assign to each tweet an actionable class label.

Information Type/Class	Earthquake	Typhoon	Total
ServiceAvailable	747	397	1144
SearchAndRescue	168	4	172
MovePeople	6	66	72
EmergingThreats	545	1632	2177
NewSubEvent	126	561	687
GoodsServices	56	45	101
Others	100	100	200

Table 6.4. A dataset of actionable information types

	X	X\R	R
Macro-F1	0.599	0.353	0.529

Table 6.5. Macro-F1 FAC-BERT - classification of actionable tweets with different input settings.

6.5.3 Results and Evaluations

We evaluate the performance of FAC-BERT on the classification of actionable tweets under the same configuration as in Section 6.3.4. Our FAC-BERT obtains 0.599 Macro-F1 in classification of actionable tweets. This result is significantly better than the leaderboard performance of 0.2784 Macro-F1. Although it is not a fair comparison since the test set is different. However, the results suggest that the task itself is quite difficult, and transfer learning-based applications such as FAC-BERT would help in getting good performance and learning faithful attention-based rationales.

Faithfulness of Actionable Rationales: As we don't have any human annotation for rationales, we used transfer learning to gather the rationale snippets in actionable tweets. As mentioned above, we trained the models based on humanitarian class-based rationales and retrieved the rationales for actionable tweets. Here, we evaluate "how well our FAC-BERT is able to extract actionable snippets from the actionable tweets". For that, we simply consider rationales extracted by the FAC-BERT-$p1$ trained on previous earthquake or typhoon and feed the second learning phase with three different input settings when classifying actionable tweets, which are original texts (X), input texts with rationales marked by '*' (X\ R) and input texts with non-rationales marked by '*'(R). This is similar to comprehensiveness or sufficiency evaluation.

Table 6.5 shows that when we mask out rationales, Macro-F1 score significantly decreases (i.e., from 59.9% to 35.3%). That means the zero-shot predicted rationales cover important content of the original tweets that FAC-BERT relies on to make predictions. Besides, when we replace non-rationales with a wild character '*', FAC-BERT performs slightly worse than the setting with original input texts. Using only rationales is sufficient to obtain decent performance.

6.6 Chapter Summary

This chapter introduces a faithful attention-based classification model. We learn to derive high-quality and faithful attention heatmaps from little human rationale supervision. We conduct experiments on different datasets of short texts from microblogs during crisis events. Experimental results show that our attention heatmaps are highly aligned with human rationales. Besides, the learned attention weights can be considered as faithful explanations, which effectively reflect the reasons for the model's decision. We also vary the size of human rationales supervision to observe the effectiveness of our model in both in-domain and cross-domain classification. Further, we show the application of our proposed model in a new setup, i.e., the detection of actionable tweets and actionable snippets. As the next step, we will evaluate the faithfulness of our attention-based explanations as a gray-scale measure of attention weights using decision flips. Besides, we aim to investigate and improve the faithfulness of attention-based explanation with a zero-shot learning setup (i.e., without human rationale supervision). We believe this kind of zero-shot learning setup helps in contributing data and new problems to TREC-IS [149] and crisisFACTS [150] tracks.

7 Conclusion

7.1 Conclusions and Discussions

This thesis has focused on the three main problems in supporting relevant information from microblooging platforms in crisis situations (1) tweet classification, (2) tweet summarization, (3) model interpretability.

In chapter 3, we propose methods for online tracking and summarization of generic breaking news events from Twitter streams. Our filtering model is a semi-supervised graph-based classification approach, that requires minimal human guidance at early stage of an event. Similarly, we develop an unsupervised extractive graph-based method for summarization of tweets in real time. As events evolve, our models automatically takes into consideration new data from incoming tweets, efficiently filter and summarize relevant information without having to re-run from scratch. The proposed models are, therefore, more scalable than previous approaches, and thus can scale up to evolving large data streams. Experiments reveal that the proposed classifier significantly outperforms other methods in filtering relevant tweets, while being as fast as the most efficient state-of-the-art method. Besides, our summarizing method obtains better performance than baselines qualitatively (in terms of human evaluation) and quantitatively (in terms of groundtruth based evaluation).

In Chapter 4, Chapter 5 and Chapter 6, we specifically focus on tweet classification and summarization challenges in the context of crisis situations. We aim at supporting humanitarian organizations and governmental bodies to obtain crucial information for situational awareness and actions. During crisis events, stakeholders usually request information of various humanitarian classes for efficient emergency assistance purpose. Besides, highly interpretable models are demanded so that the model decisions can be trusted for real-life application scenarios.

In Chapter 4, we provide human annotations of "rationales" on two crisis datasets. Rationales are supporting evidences for tweet class labels. Using the annotated data, we develop an interpretable classification-summarization framework that first clas-

sifies tweets into different humanitarian classes and then summarizes those tweets near real-time. Our classifier is an multi-task learning model which learns to classify incoming tweets from Twitter streams into different humanitarian classes and extract rationale snippets as explanations for the model decisions. In the summarization phase, we employ an Integer Linear Programming (ILP) based optimization technique that jointly optimizes the tweets and extracted rationales to generate summaries for different humanitarian classes. Experiments show that our classifier achieves high performance and interoperability. The generated summaries have 5-25% higher ROUGE-1 F-score than baseline methods and are more informative in terms of human evaluation.

In Chapter 5, we further focus on improving the performance-interpretability tradeoff of the tweet classification model and learning an efficient summarizer with *low computational complexity*. Many recent studies highlighted the importance of learning a good latent representation of tweets for several downstream tasks. Following the idea of previous works, we take advantage of state-of-the-art methods, such as transformers and contrastive learning to build an interpretable classifer. Experiments show the superior performance of our proposed model. Meanwhile, the trained classifier results in better latent representations/embeddings of tweets that help in unsupervised downstream tasks such as clustering, similarity detection, etc. Further, we propose a rationale-aware contrastive learning-based tweet summarization approach. The model utilizes learned embeddings and extracted rationales from the classification phase for an efficient summarization. Our summarizer performs equally well or better than other baseline models in terms of ROUGE-1 F score, while reducing computational complexity to a great extent.

Chapter 6 studies interpretable models under constrained human rationale annotations. The previous chapters rely on human rationales to train and extract short snippets as explanations for model interpretability. However, the rationale annotations are not always available, especially in real-time situations for new tasks and events. In this chapter, we propose a two-stage approach to learn the rationales under minimal human supervision and derive faithful machine attention. In the first stage, our classifier learns to predict rationales under limited human annotation in a semi-supervised setup. Next, the second stage utilizes the extracted rationales for class label prediction. This prediction stage consists of an encoder followed by an attention layer and a classification decoder. We incorporate the attention weights at this stage and predicted probabilities of rationales from the first stage into our customized loss to make attention weights faithful. Experimental results suggest that 40-50% human annotated rationales are good enough to get a performance similar to a fully supervised model (100% annotated rationales). Our customized loss helps learning faithful attention heatmaps. We obtain an average 20% improvement across our four datasets in learning faithful attention weights through the customized loss over the generic cross-entropy-based loss function.

7.2 Open Research Direction

This section discusses some promising future research directions following our studies in this thesis.

Interpretable multimodal classification. In this thesis, we primarily focus on developing interpretable classification models using textual content from Twitter during crisis events. However, a large number of tweets also contain images. Some previous works revealed the usefulness of image data in supporting human organizations during crisis events. For example, Nguyen et al. [100] showed that images posted on social media are helpful for severity assessment of emergency events such as natural disasters. Jing et al. [61] investigated the potential of image features for enhancing the efficiency of emergency management. Recently, Alam et al [3] provided crisis-related datasets with different annotations, such as informativeness vs. non-informative, humanitarian classes, and damage severity labels, for both textual content and images of tweets. Despite extensive research that focuses on the textual content of tweet to identify situational information or different humanitarian information types, works that use image content is limited. Hence, it is a potential future direction to focus both on textual and image content for developing multi-modal interpretable classification approaches. Similar to text data, the interpretability of image classification can be obtained by studying methods that highlight pixels or areas supporting/explaining the model outputs.

Zero-shot transfer learning for interpretable classification and summarization. In this thesis, we extend previously published datasets with humanitarian class labels by giving rationale annotations. Using the labels dataset, we develop supervised interpretable classification approaches. Further, we make attempts to evaluate our models under transfer learning setups. For example, in Chapter 5, we apply the classification model trained on one crisis event (i.e., Nepal earthquake 2015) to classify tweets and extract rationales of another crisis event of the same type (i.e., Mexico earthquake 2017). Chapter 6 studies the effectiveness of the proposed model (FAC-BERT) in (a) extracting actionable snippets from actionable tweets and (b) performing on a new dataset with a new problem setup. Generally, despite promising results, we still use the annotated datasets to some extent. Recent studies on zero-shot transfer learning [171, 122] have shown the success of pre-trained language models in getting a machine to do a task that it was not explicitly trained to do. Radford et al. [122] investigated the capacity of language models in zero-shot task transfer. Specifically, the language model GPT achieves state-of-the-art results on downstream tasks such as machine translation and text summarization without fine-tuning these tasks directly. Inspired by these studies, we would like to investigate the modern zero-shot task transfer in learning interpretable classification and summarization tasks of tweets during crisis events from zero human annotation.

Handling misinformation. Messages posted on social media platforms during crisis events are highly valuable for situational awareness and humanitarian aid. However,

not all the content obtained from these online platforms is correct or trustworthy. Many tweets are misleading or just rumors. User interaction with false content has increased steadily on social media [5], and many people share information without first checking its authenticity. For example, there was a storm of misinformation and fake news during Hurricane Harvey and Hurricane Irma, such as the false rumor "immigration status is checked at shelters", or another rumor about flood waters carrying a plague. Gupta et al. [47] revealed that 29% of the most viral content on Twitter during the Boston Marathon Blasts were rumors and fake content. In this thesis, we propose models to extract and summarize information from various humanitarian classes. It may be interesting to integrate the credibility check into our classification and summarization framework.

Multi-platform classification and summarization. In this thesis, we only employ Twitter data to extract useful information of different categories and generate event summaries. During crisis events, an enormous amount of situational information is also posted on other social media such as Facebook, Instagram, etc [86, 139]. Obtaining data from different sources helps increase the information coverage, i.e., some information about missing people, fatality rates, volunteering efforts, etc., is available on one media channel but not on other platforms. Besides, multi-media usage can be useful in the verification of information consistency/authenticity.

Bibliography

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Firoj Alam, Shafiq Joty, and Muhammad Imran. Graph based semi-supervised learning with convolution neural networks to classify crisis related tweets. In *Proceedings of the international AAAI conference on web and social media*, volume 12, 2018.

[3] Firoj Alam, Ferda Ofli, and Muhammad Imran. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the international AAAI conference on web and social media*, volume 12, 2018.

[4] M-Dyaa Albakour, Craig Macdonald, and Iadh Ounis. On sparsity and drift for effective real-time filtering in microblogs. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 419–428, 2013.

[5] Hunt Allcott, Matthew Gentzkow, and Chuan Yu. Trends in the diffusion of misinformation on social media. *Research & Politics*, 6(2):2053168019848554, 2019.

[6] Karl Appel, Lauren Mathews, Darren Lim, and Sharon Small. Siena's twitter information retrieval system: The 2012 microblog track. Technical report, SIENA COLLEGE LOUDONVILLE NY, 2012.

[7] Salman Aslam. Twitter by the numbers: Stats, demographics & fun facts. http://bit.ly/3nmYfRP, note="[Online available]", 2023.

[8] Konstantin Avrachenkov, Nelly Litvak, Danil Nemirovsky, and Natalia Osipova. Monte carlo methods in pagerank computation: When one iteration is sufficient. *SIAM Journal on Numerical Analysis*, 45(2):890–904, 2007.

[9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[10] Bahman Bahmani, Abdur Chowdhury, and Ashish Goel. Fast incremental and personalized pagerank. *Proc. VLDB Endow.*, 4(3):173–184, 2010.

[11] Ramnath Balasubramanyan and Aleksander Kołcz. " w00t! feeling great today!" chatter in twitter: identification and prevalence. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 306–310, 2013.

[12] Francesco Barbieri, Luis Espinosa Anke, Jose Camacho-Collados, Steven Schockaert, and Horacio Saggion. Interpretable emoji prediction via label-wise attention lstms. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4766–4771, 2018.

[13] Michael Barthel and Elisa Shearer. How do americans use twitter for news? In *Pew Research Center. Available at: goo.gl/Hb7Rqy*, 2015.

[14] Giacomo Berardi, Andrea Esuli, and Diego Marcheggiani. Isti at trec microblog track 2012: Real-time filtering through supervised learning. Technical report, CONSIGLIO NAZIONALE DELLE RICHERCHE PISA (ITALY), 2012.

[15] Kristen Bialik and Katerina Eva Matsa. Key trends in social and digital news media. In *Pew Research Center. Available at: goo.gl/Bt27H5*, 2017.

[16] Jingwen Bian, Yang Yang, and Tat-Seng Chua. Multimedia summarization for trending topics in microblogs. In *Proceedings of the 22nd ACM international Conference on information & knowledge management*, pages 1807–1812, 2013.

[17] Jingwen Bian, Yang Yang, Hanwang Zhang, and Tat-Seng Chua. Multimedia summarization for social events in microblog stream. *IEEE Transactions on multimedia*, 17(2):216–228, 2014.

[18] Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, 2022.

[19] Roi Blanco and Christina Lioma. Graph-based term weighting for information retrieval. *Information retrieval*, 15:54–92, 2012.

[20] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[21] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *WWW*, 1998.

[22] Cody Buntain, Richard McCreadie, and Ian Soboroff. Incident streams 2021 off the deep end: Deeper annotations and evaluations in twitter. 2022.

[23] Mark A Cameron, Robert Power, Bella Robinson, and Jie Yin. Emergency situation awareness from twitter for crisis management. In *Proceedings of the 21st international conference on world wide web*, pages 695–698, 2012.

[24] Enrique Cano-Marin, Marçal Mora-Cantallops, and Salvador Sánchez-Alonso. Twitter as a predictive system: A systematic literature review. *Journal of Business Research*, 157:113561, 2023.

[25] Rich Caruana. *Multitask learning*. Springer, 1998.

[26] Giuseppe Casalicchio, Christoph Molnar, and Bernd Bischl. Visualizing the feature importance for black box models. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 655–670. Springer, 2019.

[27] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Personalized emerging topic detection based on a term aging model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):1–27, 2014.

[28] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[29] Flavio Chierichetti, Ravi Kumar, and Bo Pang. On the power laws of language: Word frequency distributions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 385–394, 2017.

[30] Minsuk Choi, Sungbok Shin, Jinho Choi, Scott Langevin, Christopher Bethune, Philippe Horne, Nathan Kronenfeld, Ramakrishnan Kannan, Barry Drake, Haesun Park, et al. Topicontiles: Tile-based spatio-temporal event analytics via exclusive topic modeling on social media. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2018.

[31] George Chrysostomou and Nikolaos Aletras. Improving the faithfulness of attention-based explanations with task-specific information for text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual*

Event, August 1-6, 2021, pages 477–488. Association for Computational Linguistics, 2021.

[32] Freddy Chua and Sitaram Asur. Automatic summarization of events from social media. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 7, pages 81–90, 2013.

[33] Juan M Cotelo, Fermin L Cruz, and Jose A Troyano. Dynamic topic-related tweet retrieval. *Journal of the Association for Information Science and Technology*, 65(3):513–523, 2014.

[34] Prasanna Desikan, Nishith Pathak, Jaideep Srivastava, and Vipin Kumar. Incremental page rank computation on evolving graphs. In *Special interest tracks and posters of the 14th International Conference on World Wide Web*, pages 1094–1095, 2005.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics, 2019.

[36] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, pages 4443–4458. Association for Computational Linguistics, 2020.

[37] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479, 2004.

[38] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[39] Maria Jose Gacto, Rafael Alcalá, and Francisco Herrera. Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 181(20):4340–4360, 2011.

[40] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, pages 340–348, 2010.

[41] Dehong Gao, Wenjie Li, and Renxian Zhang. Sequential summarization: A new application for timely updated twitter trending topics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–571, 2013.

[42] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 6894–6910. Association for Computational Linguistics, 2021.

[43] Tao Ge, Lei Cui, Baobao Chang, Sujian Li, Ming Zhou, and Zhifang Sui. News stream summarization using burst information networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 784–794, 2016.

[44] Reza Ghaeini, Xiaoli Z. Fern, and Prasad Tadepalli. Interpreting recurrent and attention-based neural models: a case study on natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels*, pages 4952–4957. Association for Computational Linguistics, 2018.

[45] Mehreen Gillani, Muhammad U Ilyas, Saad Saleh, Jalal S Alowibdi, Naif Aljohani, and Fahad S Alotaibi. Post summarization of microblogs of sporting events. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 59–68, 2017.

[46] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU press, 2013.

[47] Aditi Gupta, Hemank Lamba, and Ponnurangam Kumaraguru. $1.00 per rt# bostonmarathon# prayforboston: Analyzing fake content on twitter. In *2013 APWG eCrime researchers summit*, pages 1–12. IEEE, 2013.

[48] gurobi. Gurobi – the overall fastest and best supported solver available, 2015.

[49] Tameru Hailesilassie. Rule extraction algorithm for deep neural networks: A review. *arXiv preprint arXiv:1610.05267*, 2016.

[50] Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage*, 87:96–110, 2014.

[51] Tuan-Anh Hoang and Ee-Peng Lim. Tracking virality and susceptibility in social media. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 1059–1068, 2016.

[52] Tuan-Anh Hoang, Thi-Huyen Nguyen, and Wolfgang Nejdl. Efficient tracking of breaking news in twitter. In *Proceedings of the 10th ACM Conference on Web Science*, pages 135–136, 2019.

[53] Huggingface. Hugging face – the ai community building the future, 2021.

[54] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Processing social media messages in mass emergency: Survey summary. In *Companion Proceedings of the The Web Conference 2018*, pages 507–511, 2018.

[55] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. Aidr: Artificial intelligence for disaster response. In *Proceedings of the 23rd international conference on world wide web*, pages 159–162, 2014.

[56] Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 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*. European Language Resources Association (ELRA), 2016.

[57] Muhammad Imran, Ferda Ofli, Doina Caragea, and Antonio Torralba. Using ai and social media multimodal content for disaster response and management: Opportunities, challenges, and future directions, 2020.

[58] Muhammad Imran, Umair Qazi, and Ferda Ofli. Tbcov: two billion multilingual covid-19 tweets with sentiment, entity, geo, and gender labels. *Data*, 7(1):8, 2022.

[59] Sarthak Jain and Byron C. Wallace. Attention is not explanation. 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, Volume 1 (Long and Short Papers)*, pages 3543–3556. Association for Computational Linguistics, 2019.

[60] Ruipeng Jia, Yanan Cao, Hengzhu Tang, Fang Fang, Cong Cao, and Shi Wang. Neural extractive summarization with hierarchical attentive heterogeneous graph network. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 3622–3631, 2020.

[61] Min Jing, Bryan W Scotney, Sonya A Coleman, Martin T McGinnity, Xiubo Zhang, Stephen Kelly, Khurshid Ahmad, Antje Schlaf, Sabine Gründer-Fahrer, and Gerhard Heyer. Integration of text and image analysis for flood event image recognition. In *2016 27th Irish signals and systems conference (ISSC)*, pages 1–6. IEEE, 2016.

[62] Davinder Kaur, Suleyman Uslu, Kaley J Rittichier, and Arjan Durresi. Trustworthy artificial intelligence: a review. *ACM Computing Surveys (CSUR)*, 55(2):1–38, 2022.

[63] Chris Kedzie, Kathleen McKeown, and Fernando Diaz. Predicting salient updates for disaster summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1608–1617, 2015.

[64] Jens Kersten, Anna Kruspe, Matti Wiegmann, and Friederike Klan. Robust filtering of crisis-related tweets. In *ISCRAM 2019 conference proceedings-16th international conference on information systems for crisis response and management*, 2019.

[65] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[66] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

[67] Onur Küçüktunç, Erik Saule, Kamer Kaya, and Ümit V Çatalyürek. Diversified recommendation on graphs: pitfalls, measures, and algorithms. In *Proceedings of the 22nd international conference on World Wide Web*, pages 715–726, 2013.

[68] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

[69] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138, 2019.

[70] Tai Le Quy, Thi Huyen Nguyen, Gunnar Friege, and Eirini Ntoutsi. Evaluation of group fairness measures in student performance prediction problems. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part I*, pages 119–136. Springer, 2023.

[71] Chaoyang Li, Zhen Yang, and Kefeng Fan. Bjut at trec 2015 microblog track: Real-time filtering using non-negative matrix factorization. In *TREC*, 2015.

[72] Rui Li, Shengjie Wang, and Kevin Chen-Chuan Chang. Towards social data platform: Automatic topic-focused monitor for twitter stream. *Proceedings of the VLDB Endowment*, 6(14):1966–1977, 2013.

[73] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8547–8555, 2021.

[74] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[75] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*, pages 150–157, 2003.

[76] Jimmy Lin, Rion Snow, and William Morgan. Smoothing techniques for adaptive online language models: topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 422–429, 2011.

[77] Nelly Litvak, Werner RW Scheinhardt, and Yana Volkovich. In-degree and pagerank: why do they follow similar power laws? *Internet mathematics*, 4(2-3):175–198, 2007.

[78] Bang Liu, Fred X Han, Di Niu, Linglong Kong, Kunfeng Lai, and Yu Xu. Story forest: Extracting events and telling stories from breaking news. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(3):1–28, 2020.

[79] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil Jain, and Jiliang Tang. Trustworthy ai: A computational perspective. *ACM Transactions on Intelligent Systems and Technology*, 14(1):1–59, 2022.

[80] Junhua Liu, Trisha Singhal, Lucienne TM Blessing, Kristin L Wood, and Kwan Hui Lim. 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, 2021.

[81] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. 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 3728–3738. Association for Computational Linguistics, 2019.

[82] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[83] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 366–376, 2010.

[84] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[85] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[86] Paige Maas, Shankar Iyer, Andreas Gros, Wonhee Park, Laura McGorman, Chaya Nayak, and P Alex Dow. Facebook disaster maps: Aggregate insights for crisis response & recovery. In *KDD*, volume 19, 2019.

[87] Walid Magdy and Tamer Elsayed. Adaptive method for following dynamic topics on twitter. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, pages 335–345, 2014.

[88] Walid Magdy and Tamer Elsayed. Unsupervised adaptive microblog filtering for broad dynamic topics. *Information Processing & Management*, 52(4):513–528, 2016.

[89] Sukanya Manna and Haruto Nakai. Effectiveness of word embeddings on classifiers: A case study with tweets. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 158–161. IEEE, 2019.

[90] Richard McCreadie, Cody Buntain, and Ian Soboroff. 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, 2019.

[91] Richard McCreadie, Cody Buntain, and Ian Soboroff. Incident streams 2019: Actionable insights and how to find them. In *17th International Conference on Information Systems for Crisis Response and Management, ISCRAM 2020, May 2020*, pages 744–760. ISCRAM Digital Library, 2020.

[92] Qiaozhu Mei, Jian Guo, and Dragomir Radev. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1009–1018, 2010.

[93] Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Degeneracy-based real-time sub-event detection in twitter stream. In *Proceedings of the international AAAI conference on web and social media*, volume 9, pages 248–257, 2015.

[94] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411, 2004.

[95] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[96] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617, 2020.

[97] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[98] Dat Nguyen, Kamela Ali Al Mannai, Shafiq Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. Robust classification of crisis-related data on social networks using convolutional neural networks. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 632–635, 2017.

[99] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. Bertweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP*, pages 9–14. Association for Computational Linguistics, 2020.

[100] Dat T Nguyen, Ferda Ofli, Muhammad Imran, and Prasenjit Mitra. 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, 2017.

[101] Minh-Tien Nguyen, Asanobu Kitamoto, and Tri-Thanh Nguyen. Tsum4act: A framework for retrieving and summarizing actionable tweets during a disaster for reaction. In *Advances in Knowledge Discovery and Data Mining: 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part II 19*, pages 64–75. Springer, 2015.

[102] Thi Huyen Nguyen, Marco Fisichella, and Koustav Rudra. A trustworthy approach to classify and analyze epidemic-related information from microblogs. *IEEE Transactions on Computational Social Systems*, 2024.

[103] Thi-Huyen Nguyen, Tuan-Anh Hoang, and Wolfgang Nejdl. Efficient summarizing of evolving events from twitter streams. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 226–234. SIAM, 2019.

[104] Thi Huyen Nguyen, Hoang H Nguyen, Zahra Ahmadi, Tuan-Anh Hoang, and Thanh-Nam Doan. On the impact of dataset size: A twitter classification case study. In *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 210–217, 2021.

[105] Thi Huyen Nguyen, Tu Nguyen, Tuan-Anh Hoang, and Claudia Niederée. On the feasibility of predicting questions being forgotten in stack overflow. *arXiv preprint arXiv:2110.15789*, 2021.

[106] Thi Huyen Nguyen and Koustav Rudra. L3s at the trec 2022 crisisfacts track. In *Proceedings of the 31st Text REtrieval Conference (TREC)*, 2022.

[107] Thi Huyen Nguyen and Koustav Rudra. Rationale aware contrastive learning based approach to classify and summarize crisis-related microblogs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1552–1562, 2022.

[108] Thi Huyen Nguyen and Koustav Rudra. Towards an interpretable approach to classify and summarize crisis events from microblogs. In *Proceedings of the ACM Web Conference 2022*, pages 3641–3650, 2022.

[109] Thi Huyen Nguyen and Koustav Rudra. Learning faithful attention for interpretable classification of crisis-related microblogs under constrained human budget. In *Proceedings of the ACM Web Conference 2023*, 2023.

[110] Thi Huyen Nguyen and Koustav Rudra. Human vs chatgpt: Effect of data annotation in interpretable crisis-related microblog classification. In *Proceedings of the ACM on Web Conference 2024*, pages 4534–4543, 2024.

[111] Thi Huyen Nguyen, Miroslav Shaltev, and Koustav Rudra. Crisicsum: Interpretable classification and summarization platform for crisis events from microblogs. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4941–4945, 2022.

[112] Jeffrey Nichols, Jalal Mahmud, and Clemens Drews. Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 189–198, 2012.

[113] Andrei Olariu. Efficient online summarization of microblogging streams. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 236–240, 2014.

[114] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 376–385, 2014.

[115] Petra Perner. How to interpret decision trees? In *Advances in Data Mining. Applications and Theoretical Aspects: 11th Industrial Conference, ICDM 2011, New York, NY, USA, August 30–September 3, 2011. Proceedings 11*, pages 40–55. Springer, 2011.

[116] Mark Edward Phillips. 2016 democratic national convention in philadelphia twitter dataset. University of North Texas Libraries, Digital Library, digital.library.unt.edu, 2017.

[117] Mark Edward Phillips. Hurricane harvey twitter dataset. University of North Texas Libraries, Digital Library, digital.library.unt.edu, 2017.

[118] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

[119] Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4782–4793. Association for Computational Linguistics, 2020.

[120] Umair Qazi, Muhammad Imran, and Ferda Ofli. Geocov19: a dataset of hundreds of millions of multilingual covid-19 tweets with location information. *SIGSPATIAL Special*, 12(1):6–15, 2020.

[121] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[122] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[123] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. 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*, pages 3980–3990. Association for Computational Linguistics, 2019.

[124] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[125] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, page 2662–2670. AAAI Press, 2017.

[126] François Rousseau and Michalis Vazirgiannis. Graph-of-word and tw-idf: new approach to ad hoc ir. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 59–68, 2013.

[127] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[128] Koustav Rudra, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh. Extracting and summarizing situational information from the twitter social media during disasters. *ACM Transactions on the Web (TWEB)*, 12(3):1–35, 2018.

[129] Koustav Rudra, Subham Ghosh, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh. Extracting situational information from microblogs during disaster events: a classification-summarization approach. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 583–592, 2015.

[130] Koustav Rudra, Pawan Goyal, Niloy Ganguly, Muhammad Imran, and Prasenjit Mitra. Summarizing situational tweets in crisis scenarios: An extractive-abstractive approach. *IEEE Transactions on Computational Social Systems*, 6(5):981–993, 2019.

[131] Koustav Rudra, Pawan Goyal, Niloy Ganguly, Prasenjit Mitra, and Muhammad Imran. Identifying sub-events and summarizing disaster-related information from microblogs. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 265–274, 2018.

[132] Ahmed Saad, El Din, and Walid Magdy. Web-based pseudo relevance feedback for microblog retrieval. In *Proceedings of The Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012*, 2012.

[133] Naveen Saini, Sriparna Saha, and Pushpak Bhattacharyya. Multiobjective-based approach for microblog summarization. *IEEE Transactions on Computational Social Systems*, 6(6):1219–1231, 2019.

[134] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 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, 2010.

[135] Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22, 2019.

[136] Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. Event detection and tracking in social streams. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 3, pages 311–314, 2009.

[137] Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 2931–2951. Association for Computational Linguistics, 2019.

[138] Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Summarizing microblogs automatically. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 685–688, 2010.

[139] Wanita Sherchan, Shaila Pervin, Christopher J Butler, Jennifer C Lai, L Ghahremanlou, and B Han. Harnessing twitter and instagram for disaster management. *IBM Journal of Research and Development*, 61(6):8–1, 2017.

[140] Lidan Shou, Zhenhua Wang, Ke Chen, and Gang Chen. Sumblr: continuous summarization of evolving tweet streams. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 533–542, 2013.

[141] Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. pages 62–75, 2021.

[142] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.

[143] Ian Soboroff, Iadh Ounis, Craig Macdonald, and Jimmy Lin. Overview of the trec-2012 microblog track. In *TREC*, volume 2012, page 20, 2012.

[144] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[145] Robert J Steed, Amaya Fuenzalida, Rémy Bossu, István Bondár, Andres Heinloo, Aurelien Dupont, Joachim Saul, and Angelo Strollo. Crowdsourcing triggers rapid, reliable earthquake locations. *Science advances*, 5(4):eaau9824, 2019.

[146] Varsha Suresh and Desmond C. Ong. Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification. 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 4381–4394. Association for Computational Linguistics, 2021.

[147] Yoshimi Suzuki and Fumiyo Fukumoto. Detection of topic and its extrinsic evaluation through multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 241–246, 2014.

[148] Sarvnaz Karimi Jie Yin Paul Thomas. Searching and filtering tweets: Csiro at the trec 2012 microblog track. *TREC Microblog*, 27, 2012.

[149] TREC. Trec incident streams: Enabling emergency services with social media data. https://www.dcs.gla.ac.uk/~richardm/TREC_IS/, 2020.

[150] TREC. Crisis facts, 2022.

[151] Martin Tutek and Jan Snajder. Staying true to your word: (how) can attention become explanation? In *Proceedings of the 5th Workshop on Representation Learning for NLP, RepL4NLP@ACL 2020, Online, July 9, 2020*, pages 131–142. Association for Computational Linguistics, 2020.

[152] Twitter. Twitter api. https://developer.twitter.com/en/products/twitter-api.

[153] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[154] István Varga, Motoki Sano, Kentaro Torisawa, Chikara Hashimoto, Kiyonori Ohtake, Takao Kawai, Jong-Hoon Oh, and Stijn De Saeger. Aid is out there: Looking for help from tweets during a large scale disaster. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1619–1629, 2013.

[155] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[156] Sudha Verma, Sarah Vieweg, William Corvey, Leysia Palen, James Martin, Martha Palmer, Aaron Schram, and Kenneth Anderson. Natural language processing to the rescue? extracting" situational awareness" tweets during mass emergency. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 5, pages 385–392, 2011.

[157] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[158] Eric Wallace, Tony Z. Zhao, Shi Feng, and Sameer Singh. Concealed data poisoning attacks on NLP models. 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 139–150. Association for Computational Linguistics, 2021.

[159] Bairong Wang, Bin Liu, and Qi Zhang. An empirical study on twitter's use and crisis retweeting dynamics amid covid-19. *Natural Hazards*, 107(3):2319–2336, 2021.

[160] Xinyue Wang, Laurissa Tokarchuk, Félix Cuadrado, and Stefan Poslad. Exploiting hashtags for adaptive microblog crawling. In *Proceedings of the 2013 ieee/acm international conference on advances in social networks analysis and mining*, pages 311–315, 2013.

[161] Xinyue Wang, Laurissa Tokarchuk, and Stefan Poslad. Identifying relevant event content for real-time event detection. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pages 395–398. IEEE, 2014.

[162] Zhenhua Wang, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. On summarization and timeline generation for evolutionary tweet streams. *IEEE Transactions on Knowledge and Data Engineering*, 27(5):1301–1315, 2014.

[163] Zhenhua Wang, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. On summarization and timeline generation for evolutionary tweet streams. *TKDD*, 2015.

[164] BP Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

[165] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. 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 11–20. Association for Computational Linguistics, 2019.

[166] Wikipedia. Wilcoxon signed-rank test, 2021.

[167] Wikipedia. Automatic summarization. https://en.wikipedia.org/wiki/Automatic_summarization, 2023. [Online; last edited 07-March-2023].

[168] Wikipedia. Right to explanation. https://en.wikipedia.org/wiki/Right_to_explanation, 2023. [Online available; last edited on 16 March 2023].

[169] Wikipedia. Twitter. https://en.wikipedia.org/wiki/Twitter, 2023. [Online available; last edited 14-March-2023].

[170] Shuang-Hong Yang, Alek Kolcz, Andy Schlaikjer, and Pankaj Gupta. Large-scale high-precision topic modeling on twitter. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1907–1916, 2014.

[171] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. 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*, pages 3912–3921. Association for Computational Linguistics, 2019.

[172] Zhonggen Yu and Liheng Yu. A meta-analytical review on the effect of twitter use in education. *International Journal of e-Collaboration (IJeC)*, 18(1):1–20, 2022.

[173] Yiming Zhang, Ke Chen, Ying Weng, Zhuo Chen, Juntao Zhang, and Richard Hubbard. An intelligent early warning system of analyzing twitter data using machine learning on covid-19 surveillance in the us. *Expert Systems with Applications*, 198:116882, 2022.

[174] Zijian Zhang, Koustav Rudra, and Avishek Anand. Explain and predict, and then predict again. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 418–426, 2021.

[175] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval: 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings 33*, pages 338–349. Springer, 2011.

[176] Hao Zheng and Mirella Lapata. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6236–6247. Association for Computational Linguistics, 2019.

[177] Deyu Zhou, Liangyu Chen, and Yulan He. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

[178] Bolong Zhu, Jinghua Gao, Xiao Han, Cunhui Shi, Shenghua Liu, Yue Liu, and Xueqi Cheng. Ictnet at microblog track trec 2012. Technical report, CHINESE ACADEMY OF SCIENCES BEIJING INST OF COMPUTING TECHNOLOGY, 2012.

[179] Arkaitz Zubiaga, Damiano Spina, Enrique Amigó, and Julio Gonzalo. Towards real-time summarization of scheduled events from twitter streams. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 319–320, 2012.