A Neural Network-Based Situational Awareness Approach for Emergency Response

Mehdi Ben Lazreg

A Neural Network-Based Situational Awareness Approach for Emergency Response

Doctoral Dissertation for the Degree *Philosophiae Doctor (PhD)* at the Faculty of Engineering and Science, Specialization in Information and Communication Technology

University of Agder
Faculty of Engineering and Science
2020

Doctoral Dissertations at the University of Agder 264
ISBN: 978-82-7117-965-6
ISSN: 1504-9272

Printed by 07 Media
Kristiansand

Dedicated to

My parents

Abstract

In the past few years, social media has become an open medium for a variety of emergency communications. First, emergency management services use social media to inform the public about the status of an emergency and what precaution the public needs to take. Second, the citizens use these online mediums to check on the safety of loved ones. Finally, eyewitness and affected individuals share their observations, concerns, and challenges they face during an emergency. Yet, despite the apparent potential benefit from collecting critical information from social media to improve situational awareness, emergency management services are still reluctant to use social media as a source of information. The main reasons for this reluctance are trustworthiness and information overload.

This thesis studies, from a technical perspective, the available social media analysis platforms and the problems faced by emergency management services when it comes to leveraging social media. We identify two main challenges. The first challenge is the nature of the language used on social media with all the misspellings, leetspeak, and abbreviations, which requires a specific and adaptive normalization technique. The second challenge is that social media analytics platforms often do not provide emergency management services with the information they need during an emergency, and thus do little to increase the situational awareness. Most of the dominant platforms try to find accurate ways of extracting as much information related to the crisis as possible, which does not explicitly address the specific information-requirements of the time-constrained emergency personnel. An automated approach for determining the demanded information in emergency response situations would be advantageous. However, automatically detecting the information-needs is particularly challenging since what information is required varies from an emergency to another and usually depends on what stage the crisis is in and what information has already been made available.

To address the first challenge, we propose a string metric that embraces similarities between strings based on (1) the character similarities between the words, including non-standard and standard spellings of the same words, and (2) the context of these words. For the second challenge, we propose an intelligent information re-

trieval framework for social media that, given the status of the emergency, provides the information most likely needed by the emergency management services. This framework combines two components. The first component classifies social media messages into separate topics representing information required by emergency services during a specific situation. The second component decides which information to retrieve by learning what the emergency services need, based on the information available and the status of the emergency. This component is implemented with the introduction of the Conditional Neural Turing Machine.

Our results show that the proposed string metric algorithm finds the correct version of a non-standard spelling 85.4% of the cases on a data set composed of the 1 051 most frequently used words on Twitter. We also show that the words used in a similar context, in contrast to different, are learned to be more alike. The Conditional Neural Turing Machine is verified both in a theoretical randomly generated sparse graph, and a realistic graph of information-requirements in emergencies. The proposed algorithm is in the latter able to predict what information will be required in 78.59% of cases. The framework as a whole is evaluated through a survey of eleven emergency management experts. The automated framework and the personnel agree on information that is needed in 70.09% of the cases.

Acknowledgments

I would like to thank my supervisor, Associate professor Morten Goodwin, and my co-supervisor, Professor Ole-Christoffer Granmo. Your knowledge and thrive towards research excellence not only shaped this Ph.D. but also me as a researcher and scientist. Thank you for your guidance, comments, and support.

I would like to thank all the staff and the research centers at the University of Agder: The head of the ICT department, Folke Haugland, for his support, the administrative staff Tonje, Emma, and Kristine for their help with almost everything, my colleagues at the Center for Integrated Emergency Management for their enthusiasm and assistance –special thanks to Jaziar and Tim. Collaborating with you was an enriching chance–, my colleagues at the Centre for Artificial Intelligence Research for their shared knowledge. Accomplishing my Ph.D. at the University of Agder is an honor and a privilege.

I would like to thank my colleagues from the iTrack project, Professor Tina Comes, my friend Ahmed, and Nadia. iTrack was a big part of my Ph.D. journey. Working on the project was an ascending learning curve for me. It would not have been possible without your directions. Special thanks to Ahmed, who took the time to review my dissertation.

I would like to thank my colleagues at the University of Duisburg-Essen, Tobias, Bjrn, and Stefan for their hospitality during my visit to Duisburg and the insightful work we did then. I would like to thank my colleagues at the University of Pittsburgh, my host Professor Vladimir and his team Usman, Yang, and Danchen. I am grateful for the opportunity to stay with your research group and looking forward to visiting you again.

I would like to thank the emergency personnel and experts, who took part in this research and answered my surveys. Special thanks to Bente Skattr form the Oslo police for her help in that phase.

I would like to thank my colleagues Lei and Ali, who reviewed my dissertation and help shape its consistency. I would like to thank my colleagues of years past, Huihiu, Lashmikanth, Mohamed Hamid, Mohamed Talab. Our memories will ever last. I would like to thank my current colleagues, Rym, Leila, Mohmed Norani,

Ahmed Salem, Indika, and Ahmed Abdelgawwad, for our lengthy discussion and the good times we had. I would like to thank my office mates past and present, Hossein, Joel, and Ahmed, for our conversations and ideas ever potent.

To my friend and mate, Mohamed Abomhara, for the encouragement and the moments we shared. To my friend and companion Ahmed Radman and his family, for the support and the roads we traveled. To my friend, colleague, and office mate Vimala, for the feedback and the kindness you carried. To my friend Basel and his family, for the aid and the hospitality you showed. To Grimstad and the souls I met along the way, who either helped or eased my stay, thank you.

Mehdi Ben Lazreg
March, 2020
Grimstad, Norway

List of Publications

All the papers listed below are an outcome of the research work carried out by the author of this dissertation, including one submitted and five published papers.

Papers Included in the Dissertation

Paper A: Mehdi Ben Lazreg , Narayan Ranjan Chakraborty, Stefan Stieglitz, Tobias Potthoff, Bjrn Ross, and Tim A. Majchrzak. "Social Media Analysis in Crisis Situations: Can Social Media be a Reliable Information Source for Emergency Management Services?." *In Designing Digitalization (ISD2018 Proceedings).* Lund, Sweden: Lund University. ISBN: 978-91-7753-876-9.

Paper B: Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo. "Combining a context aware neural network with a denoising autoencoder for measuring string similarities." *Computer Speech & Language* 60 (2020): 101028.

Paper C: Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo. "A Neural Turing Machine for Conditional Transition Graph Modeling." Submitted to to *IEEE Transactions on Neural Networks and Learning Systems.*

Paper D: Mehdi Ben Lazreg, and Morten Goodwin, and Ole-Christoffer Granmo. "An Iterative Information Retrieval Approach from Social Media in Crisis Situations Situations." *In International Conference on Information and Communication Technologies for Disaster Management (ICT-DM 2019 Proceedings).* Paris, France.

Paper E: Mehdi Ben Lazreg, Nadia Noori, Tina Comes, and Morten Goodwin. "Not a Target. A Deep Learning Approach for a Warning and Decision Support System to Improve Safety and Security of Humanitarian Aid Workers." *In IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 378-382. ACM, 2019.

Paper F: Mehdi Ben Lazreg, Usman Anjum, Vladimir Zadorozhny, and Morten Goodwin. "Semantic Decay Filter for Event Detection." *In the* 17th *International Conference on Information Systems for Crisis Response and Management ISCRAM 2020.* Virginia, USA.

Other Publications Not Included in the Dissertation

Paper 1: Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo. "Deep learning for social media analysis in crises situations." *In The* 29th *Annual Workshop of the Swedish Artificial Intelligence Society (SAIS)* 23 June 2016, Malmö, Sweden, p. 31. 2016.

Paper 2: Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo. "Information abstraction from crises related tweets using recurrent neural network." *In IFIP International Conference on Artificial Intelligence Applications and Innovations*, pp. 441-452. Springer, Cham, 2016.

Paper 3: Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo. "Vector representation of non-standard spellings using dynamic time warping and a denoising autoencoder." *In 2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1444-1450. IEEE, 2017.

Contents

List of Figures

List of Tables

List of Abbreviations

CNTM	Conditional Neural Turing Machine
EMS	Emergency Management Service
FSM	Finite State Machine
LDA	Latent Dirichlet Allocation
LSH	Locality-Sensitive Hashing
LSTM	Long Short-Term Memory
NER	Named-Entity Recognition
NLP	Natural Language Processing
NTM	Neural Turing Machine
QG	Query Generator
QoI	Quality of Information
SDF	Semantic Decay Filter
SVM	Support Vector Machine
TD	Topic Detector

PART I

Chapter 1

Introduction

> *In the world of emergency management, we often see a demand for science that "proves" behaviours. For example, does the use of social media save lives or improve emergency management? One wonders if the same questions were asked about the first landline telephones.*
> ***–Leysia Palen in "Nature"***

1.1 Scope

During the past decade, the impact of natural and human-made disasters have been growing steadily. In 2017, more than 11 000 persons lost their lives or went missing, while millions were left homeless due to disasters [1]. These disasters caused a total economic losses of USD 337 billion [1]. To efficiently cope with an emergency, and minimize both human and financial loss, Emergency Management Services (EMS) create emergency management plans [2]. Such plans have three phases: response, rehabilitation, and recovery. The response phase is the most crucial part of emergency management. In this phase, vital activities and operations are laid out, such as search & rescue, damage and needs assessments, provision of first aid, and humanitarian assistance to those affected. An active response phase requires the EMS to perform a rapid assessment of any damage caused by the emergency. This assessment is commonly referred to as situation awareness and is based on gathering and compiling information from different sources including, weather forecast agencies, EMS personnel on the ground, drones, etc.

In addition to the traditional information sources, EMSs must face up to the challenges and opportunities of the information age, in which social media plays a pivotal role as a means of communication. Social media is used to warn people,

collect information from the field, manage public responses, answer the publics questions [3, 4]. As an example, during the Nepal earthquake in 2015, 106 380 tweets were shared about the disaster following the week of the disaster [5]. More recently, 1 653 tweets were shared about the London Bridge attack in 2017, 1 024 tweets about the shooting in STEM school highlands ranch in 2019, and 1409 about the Virginia beach shooting all in a radius of 800 meters around the location of the events and in less than 12 hours following the incidents (see **Paper F**).

Social media also enables a wide intentional or unintentional propagation of false information (fake news). During an emergency, fake news can have significant adverse effects on both the EMSs social media-based assessment of the situation and the accuracy of automated information extraction methods. Therefore, fake news detection on social media has recently become an important research area. Shu et al. [6] present a comprehensive overview of this research field.

Purely textual information, beyond fake news [1], is at the heart of most social media platforms. However, processing and inferring valuable knowledge from textual information publicly available on social media platforms is complicated for several reasons. First and foremost, the huge burst in social media posts during an emergency requires massive resources from the EMSs to read and analyze the posts. Most EMSs do not prioritize such analysis during an emergency. Furthermore, most of the automatic analysis tools struggle in extracting emergency related posts because the social media posts are typically brief, informal, and heterogeneous (e.g., a mix of languages, acronyms, and misspellings) with varying levels of quality. Moreover, to understand the meaning of a message, the reader often has to also understand the context.

There is no doubt that valuable, high throughput data is produced on social media only seconds after an emergency occurs. However, the data throughput on Twitter is 7GB/min [7], consequently, an emergency-related tweet is drowned in the vast majority of tweets about other mundane events. Even among the emergency-related tweets, the topics discussed by those tweets are often irrelevant to the EMSs. For example, by performing keyword analysis, Radianti et al. [5] found that during the 2015 Nepal earthquake the most discussed topics were on monetary support, followed by injuries, deaths, and missing people. Nevertheless, when users post about financial aid, mostly they call for donation in contrast with monetary needs which is more relevant for the EMSs.

The non-standard nature of the social media text coupled with the need to retrieve relevant information for EMSs are the two main challenges facing social me-

[1] Even though fake news detection is out of the scope of this research, it is worth mentioning that its results can have a significant impact on the social media analysis field in general.

dia analysis during emergencies. These challenges are addressed by two different research fields, respectively: text normalization, and classification/clustering of social media text.

1.2 Text normalization

Text normalization is the method of transforming texts, typically with misspellings or written in a specific jargon, into a standard canonical form [8]. The normalization of text usually takes place before any additional processing, such as categorization. In social media, this task is particularly challenging because of the nature of social media posts. The writing on social media is, in most cases, brief and informal. This property has made social media a place where all types of spellings are accepted, including sociolects, dialects, leetspeak, and phonetic spelling. The latter is particularly common – a type of grammar based on the way the word sounds rather than following the correct orthography. To top it off, the language used in social media is also usually fragmented, containing typographical errors, and incoherent [9].

The analysis of social media posts, particularly in an emergency management setting, has an additional level of complexity, namely, the source of the post. When collecting social media texts, the standard approach is to retrieve as many messages as possible, regardless of their origins. However, messages posted by an eyewitness present very different properties than messages posted by a traditional media sources (e.g., news anchor or journalist). As a consequence, there is, in most cases, a high discrepancy in the messages and information quality (e.g., clarity, readability, and conciseness) suffers [10]. Additionally, a single message can have text snippets with entirely different languages in it – a phenomena known as borrowing [10]. Because of these constraints, the word's context in a social media post becomes crucial to understand its meaning.

Text normalization traditionally follows three separate paradigms [11,12]. First, it has been seen as a spell correction problem. Here, a model, often statistically based, is used to transform a non-standard spelling to its standard form on a word-per-word basis. Second, normalizing a non-standard text can be done as if it was a foreign language in need of translation. In this setting, the text normalization task becomes a machine translation task. Finally, non-standard texts has been handled as if it was written in a phonetic spelling. The normalization task, then, becomes a type of speech recognition problem, which consists of decoding a word sequence in a phonetic framework.

A completely different way to solve the text normalization problem is to trans-

form every word into a vector of real numbers. Such vector should encode all semantics and synthetics influencing the word spelling. Various methods have been used following the general principle of text encoding, including the word co-occurrence matrix [13, 14], probabilistic models [15], and explainable knowledge methods [16]. With the advances in neural networks, this way of text normalization is gaining more traction. A neural network can learn the vector representation of a word form the data, based on the surrounding words [17]. Most often, we call this representation-approach for word embedding.

1.3 Social media analytics

Social media analysis platforms have tried to address some of the difficulties mentioned in Section 1.1 through a mix of different approaches. Two main objectives are addressed by the analysis platforms: Event detection and tracking, and classification and clustering [18].

In social media, an event is defined as "an occurrence causing a change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time and often associated with entities such as people and location" [19]. The event detection task in social media is rather difficult since data emerge quicker and in larger volumes than traditional document streams. For emergency management, the research on event detection is grouped into two separate divisions: Offline and online event detection.

Offline event detection is the practice of identifying events from the past, usually from stored messages. Methods for this type of event detection frequently involve clustering the messages based on predefined similarities between them, which may include diverse dimensions like common words, group of senders, time, and geolocation of the sender [20–22].

Online event detection tries to recognize new events in real-time, often in a stream of data and typically without any prior knowledge about the event. Methods to achieve this detection are mainly based on message and keyword bursts [23–26], keyword frequency [27], and wavelet signals [28].

When crises related social media messages are detected, the next objective, in most social media analytical systems, is to categorize the text in the post. Again, this can be separated into two categories of techniques. Some approaches do classification and clustering of the data, while others only extract information from them. The latter is referred to as information extraction.

Information extraction can be further defined as extracting structured informa-

tion from unstructured or semi-structured data [29]. The most common information extraction task is entity extraction or recognition. Here, the methods carry out text transformation from natural language (e.g., five people were injured in Grimstad) to a structured, machine-readable record (e.g., <Number-of-injured=5, location=Grimstad>). Most notably, Imran et al. [30, 31] used this technique to transform a text in natural language, then further employed conditional random fields to extract infrastructure damage and donations from twitter messages.

The classification and clustering of social media messages attributes a topic or a set of topics to a message or a cluster of messages with varying human intervention. Although there is no standard way of categorizing social media messages, most literature tries to classify data items using the following dimensions [18]:

- Factual, subjective, or emotional content;
- Information provided by the messages;
- Information source (select information coming from a particular group of peoples);
- Credibility of the information;
- Time and stages of the event;
- Location from which the message originates.

An obstacle with such classifications is that due to the highly dynamic nature of emergencies, we need to redefine the labels (topics) of the classification for each emergency and sometimes during the same emergency. Automating the process of inferring the information (topics) to be extracted from social media, given an emergency context, would optimize the information gathering process. In the next section, we will give an insight into how such automation can be performed.

1.4 Link prediction

All large-scale emergencies are complex events in which many variables change over time. The information needed by EMSs widely varies from an emergency to another. The required information changes even during the same crisis. A typical situation is that providing a piece of new information will trigger an additional need for even more information. An example is when receiving information that a fire has started in a specific location. The event-information, naturally, leads to

an inquiry about the number of people living in that location. Such an information gathering process can be modeled as a graph, and this graph will directly influence the decisions and interventions to take depending on the status of the emergency.

The information-gathering process in social media analytics for emergency management is also highly complex and depends on the status of the emergency, and the information gathered so far. It could be possible, one might argue, to organize this process as a simple Finite State Machine (FSM). In this FSM, each state would represent the required information, and the transitions would be inputs of the emergency management status.

On the other hand, the share number of types of emergencies, ranging from natural, human-made to technological, is continually growing [32]. Each crisis is dynamic and continuously evolving, which would make any FSM-based approach vast and exhausting to maintain and update. Furthermore, in such an FSM graph, it would be challenging to represent any collective knowledge between the crises. Such an FSM, would not have any properties beyond what is modeled, making new emergencies harder to model.

As a counter-argument, let us assume that there exists an FSM that represents the information graph of some generic crisis. Can this graph be generalized to other disasters, and the different situations in one particular emergency? If so, the problem then boils down to that of link prediction in the sense of inferring missing links from an observed graph.

To exemplify, let us assume a state S for emergency A require information Q. A could represent the a fire emergency, and the state S could then represent that we need to know the location of the fire, and Q is the actual location. Upon obtaining Q, we move to a new state S_1 which could represent the need to know if there are any victims on sight.

In similar state $S^{'}$ of a similar emergency $A^{'}$, if we acquire the same information Q, then we are likely to move to the state S_1. $A^{'}$ could represent a flood, while the state $S^{'}$ still represents not knowing the location and Q is the actual location. In this case, we predicted a link between $S^{'}$ and S_1 (See figure 1.1).

The overall objective, seen from the emergency personnel point of view, is to know what information they should gather. With this in mind, situations $S^{'}$ and S are very similar. We can argue that there is a general knowledge in an emergency S which could be transferred to $A^{'}$.

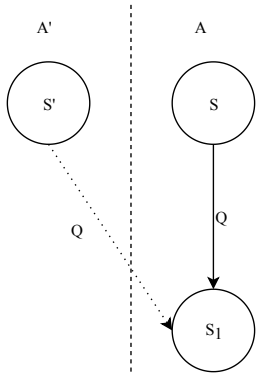


Figure 1.1: Example of link prediction in a conditional graph.

1.5 Motivation

During an emergency, social media has been used mainly under three separate communication channels. One usage is from an EMS-to-public, the EMSs relay situation updates, evacuation orders, and possible dangers. Another usage is as a public-to-public channel; the public uses social media to maintain contact with relatives, friends, and loved ones, and to support the community. Finally, social media is used from the public back to the EMS; the public uses social media to report problems, needs, calls for help, and provide information during the emergency.

The first two channels (EMS-to-public and public-to-public) are becoming well established. The EMSs obtain training on how to effectively communicate information to the public [33, 34], and many social media platforms provide ways for people in the affected area to report their safety status. Nonetheless, the third usage, from the public back to the EMS, still has many open challenges.

There is no doubt that social media already plays a pivotal role in helping the public inform the EMS as of today. Using social media, EMS can assess the emergency by, for example, getting updates from people in the affected areas. However, most EMSs are still skeptical about using that information [3, 35]. There might be various reasons for this. The most pivotal reason is the rumors and misinformation spread on social media [36]. Additionally, the lack of needed functionality in social media analytics platforms can also be a reason of skepticism among EMSs personnel. As an example, most of the research on Twitter data during an emergency follows a data-driven approach. The data is analyzed first by finding ways to accurately extract as much information related to the crisis as possible. Unfortunately, the EMSs requirements are rarely taken as an input of the collection process. [18]. Such an approach may result in an overload of information extracted during the analysis that are not useful to the EMS. Automating the process of inferring the

required information given the emergency context will reduce such overload.

Most of the Twitter analysis platforms try to use state-of-the-art machine learning techniques and apply them to social media data. During this process, they neglect the specificity of social media posts [31]. In many cases, such posts use language that differs from any standard natural language. This language used often contains leetspeak, abbreviations, and misspellings, among other things. The same sentence can be written in many ways, which makes it practically impossible to analysis by an automatic technique that does not take into account this constraint. Consequently, text analysis for social media will specifically require an approach of text prepossessing that takes into consideration the constraints of the social media language. A valuable analytical tool should handle the language of social media, such as words normalization used therein to facilitate the classification of the message.

1.6 Research questions

This thesis focuses on the automatic extraction of textual information from the social media public to EMS channels to improve the perception element of the situation awareness process during the response phase of an emergency. To achieve this objective, we investigate five research questions (RQ):

- RQ1: What needs to be present in a social media analysis platform for it to be accepted and used by EMS in their situation awareness process?

- RQ2: How can we identify and model the dynamics of the information required by EMS during an emergency?

- RQ3: How can we automate the extraction of the information required by EMS identified in RQ2?

- RQ4: How can we cope with the specific language used on social media platforms in order to improve the classification of messages posted on the platforms?

- RQ5: How can we combine the finding of RQ3 and RQ4 into an automated information retrieval framework from social media?

1.7 Thesis outline

This dissertation is composed of two parts. **Part I** summarizes the research carried out throughout the Ph.D. and presents the main findings. **Part II** contains the collection of six research papers representing the main contribution of this thesis.

The remaining Chapters in **Part I** are as follows:

- Chapter 2 positions this work with respect to the existing literature and identifies the gaps that this thesis addresses. It further presents the finding of Paper A and answers RQs1-2.
- Chapter 3 presents the algorithm developed in Papers B-F to answer RQs3-5.
- Chapter 4 evaluates the algorithm illustrated in Chapter 3, and gives the main results of the evaluation.
- Chapter 5 concludes the dissertation, present the research limitation, and points to future research directions.

Part II contains the following research papers:

- **Paper A** identifies the information needs, the problem faced by EMSs, and the ways they can use social media. We particularly address the case of extreme weather. The paper further pinpoints what different social media analysis platforms can provide in this type of emergency. The results of the research are criteria that social media analysis should follow to address EMSs' concerns. The output of this work can be used to more precisely describe social media communication for crises and to design more efficient platforms for information retrieval from social media.
- **Paper B** proposes a string metric that encompasses similarities between strings based on (1) the character similarities between the words, including non-standard and standard spellings of the same words, and (2) the context of these words. We propose a neural network model composed of a denoising autoencoder and what we call a context encoder, both specifically designed to find similarities between the words based on their context.
- **Paper C** proposes extensions of memory based Neural Turing Machine (NTM) with two novel additions. We allow for transitions between nodes to be influenced by information received from external environments, and we let the NTM learn the context of those transitions. We refer to this extension as the

Conditional Neural Turing Machine (CNTM). The CNTM is used to predict links in a graph in which an external input conditions the transitions (edges).

- **Paper D** proposes an intelligent information retrieval framework for social media in crises. The developed framework combines two components. The first component classifies social media messages into separate topics representing information or a question asked by emergency services during a specific situation. The second component decides which information to retrieve based on the information available and the status of the emergency using the algorithm developed in **Paper C**.

- **Paper E** presents a threat detection and decision support system that combines knowledge and information from a network of responders with automated and modular threat detection. The system consists of three parts. It first collects textual information from social media, and online news reports. Second, the system automatically prepossesses the data using the algorithm developed in **Paper B** and detects a threat or incident and extracts information, including location, threat category, and casualties. Third, given the type of threat and the information extracted, the system proposes a mitigation plan based on humanitarian standard operating procedures.

- **Paper F** proposes an event detection method based on peaks in the number of tweet that also uses the string metric introduced in **Paper B**. Using peaks in the number of posts and keyword bursts has become the go-to method for event detection from social media. However, those methods suffer from the random peaks in posts attributed to the regular daily use of social media. This paper proposes a novel approach to remedy that problem by introducing a low pass filter. The filter's role is the eliminate the random peaks and preserve the peaks related to an event.

The papers in **Part II** can be divided into three main topics: text analysis, emergency management, and link prediction. They can be read following the flow diagram in Figure 1.2 without loss of consistency.

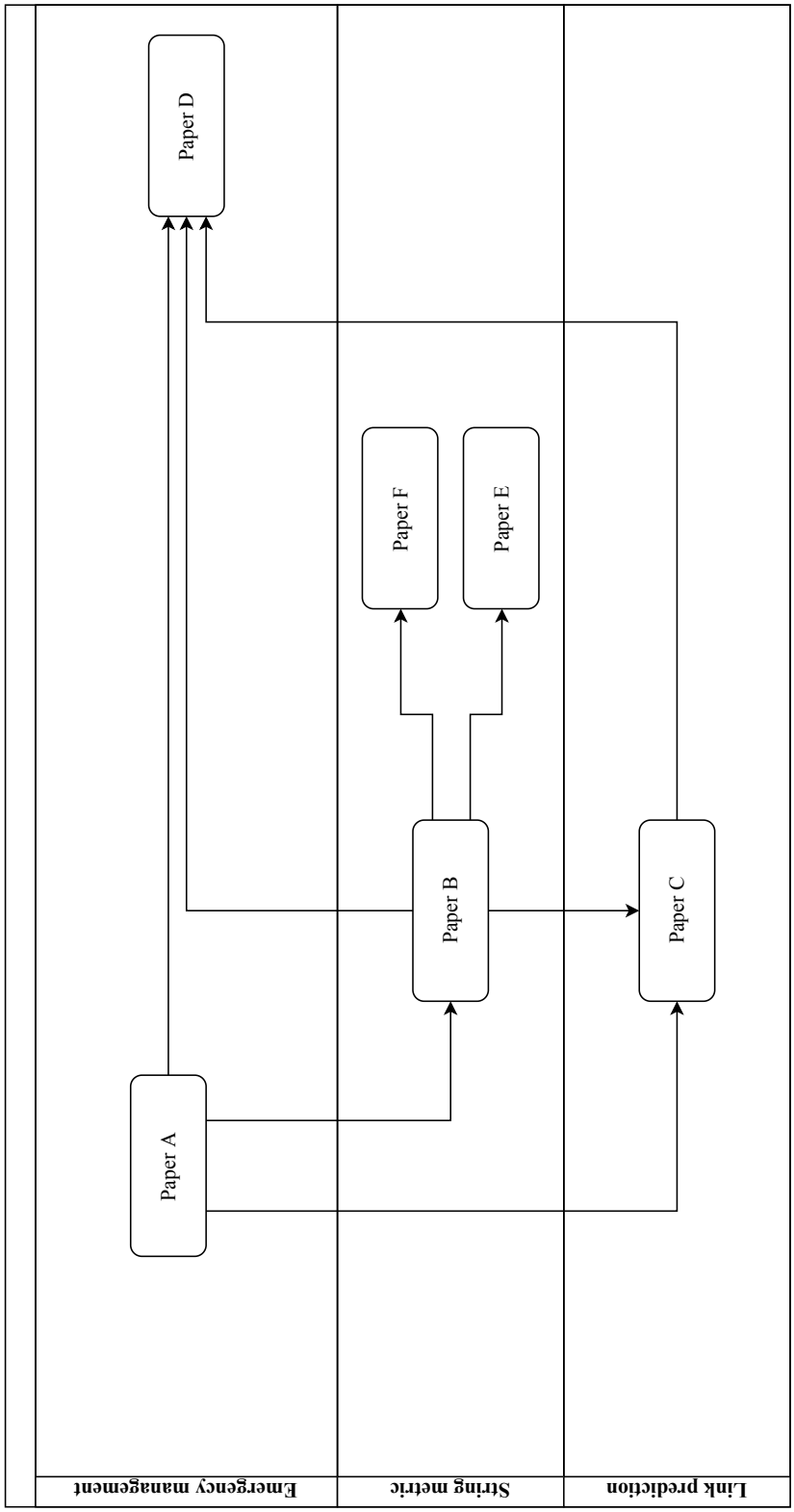


Figure 1.2: Thesis map

Chapter 2

Background and Research Gaps

Even Mombi was not without a curious interest in the man her magic had brought to life; for, after staring at him intently, she presently asked:
"What do you know?"
"Well, that is hard to tell," replied Jack. "For although I feel that I know a tremendous lot, I am not yet aware how much there is in the world to find out about. It will take me a little time to discover whether I am very wise or very foolish."
–L. Frank Baum in "The Treasury of Oz"

Social media analysis in emergencies is a multidisciplinary field combining computing and knowledge of emergencies. Social media research focuses mostly on the analysis of large-volume data by extracting as much information about an emergency as possible. The usefulness of that information to EMSs is often an afterthought, which explains some of the doubts about social media's value by EMSs. Besides, the intricacies of the social media posts need to be studied in more depth to develop more efficient information extraction tools.

In Chapter 1, we argued that social media could contain useful information to EMS personnel for their situational awareness that can add to what is being processed during an emergency today. In Section 2.1, we dive more into the type of information available on social media with more concrete examples. Section 2.2 gives an overview of the research literature and identifies the gaps related to the first topic of this thesis addresses: the non-standard spelling issues in social media posts. Further, Section 2.3 presents an overview of the automated techniques and platforms used to extract information from social media. In Section 2.4, we summarize the results of **Paper A**. We reveal why EMSs are still skeptical about the use

of social media, and we identify the gaps between the EMSs requirements and what social media analytics platforms provide. One of the EMSs requirements is an analysis platform that addresses their information requirements. In Section 2.5, we also study the dynamics of those requirements and show that they can be modeled using a conditional graph. Further, we show that the problem of automatically identifying information requirements by EMS is a link prediction problem in an incomplete graph. Finally, Section 2.5 also presents a literature overview regarding the link prediction problem and motivates the need for a new method to predict links in a conditional graph.

2.1 Information available on social media during an emergency

Since 2007, numerous researchers have shown that in time of emergencies the public converge to the social media platforms. During an emergency, users typically manage the social media life around contacting loved ones, getting updates on the status of the crisis, show support, and report on the crisis development [4, 37–40]. In this Section, we compile a comprehensive overview of the research on social media content analysis, which illustrates the type of information shared during an emergency. This overview aims to demonstrate that social media contains information that can help EMSs in their situation awareness process as well as irrelevant information to EMS even among emergency related posts.

Vieweg et al. [41] investigated the tweets shared during four separate crises: The 2009 Oklahoma fire, the 2009 and 2010 Red river flood, and the 2010 Haiti earthquake. They filtered 37 802 tweets about each crisis and categorized them by information-type. They found that, on average, only 29.32% of the tweets discuss the crisis to some degree, and 15.12% of those tweets provide updates and information about the statue of the crisis.

Radianti et al. [5] analyzed 430 708 tweets about the Nepal earthquake in 2015. They discovered that 24.7% of tweets discuss public concerns during the crisis, and only 30.36% of the information present in those discussions is about reported casualty and infrastructure damage. Monetary support is also one of the most discussed topics, accounting for 11.66% of the total number of tweets. However, the majority of those messages are just appeal for donations posted by ordinary people or organizations outside the affected areas.

Olteanu et al. [9] analyzed 29 4200 tweets from 26 distinct crises in 15 countries between 2012 and 2013. They found that, on average, 59% of the data can

Table 2.1: Overview of the information discussed in crises-related posts

Emergency	Affected people	Infrastructure	Physical environment
2009 Oklahoma fire	37.0%	13.7%	18.9%
2010 Red river flood	54.6%	3.5%	31.8%
2010 Haiti earthquake	83.7%	2.5%	6.7%
2013 Alberta flood	7.8%	19.8%	10.4%
2013 Queensland flood	10.66%	10.08%	18.25%
2015 Nepal earthquake	16.88%	13.48%	0.12%

contribute to the situation awareness process of the crisis. That data groups as follows: 32% of the tweets provide a description of the crisis status, 20% report on affected individuals, and 7% report on infrastructure status.

Table 2.1 presents an overview of the percentage of tweets on different topics from the tweets related to the crisis.

2.2 Non-standard spelling issue in social media text

One characteristic of many social media messages is the irregular vocabulary usage. The same phrase – and even the same word – can be expressed in a variety of ways within the same language. This evolution presents a challenge to natural language processing (NLP), such as translation and classification, as any data handling has become more complex. It is particularly challenging in emergency management where EMS rely on both accurate and up-to-date information. One mitigation of this situation is to normalize non-standard words to a more standard format that is easier to handle.

A typical approach to normalize misspellings, abbreviations, dialects, sociolects, and other text variations (referred to here as non-standard spelling and non-standard words) is the use of string metrics. The approach measures the distance between two text strings and in this way find the closest matching string (see **Paper B** for a comprehensive background on string metrics). Other approaches in the literature can be grouped into three main categories.

The first method is to view the normalization of any non-standard spelling as a translation problem [11]. One example uses statistical tools that map the non-standard words with their English counterparts based on a probability distribution [42]. Indeed, the translation method is a promising strategy, however, using a technique designed to capture the complicated relationship between two different

languages in the context of word normalization is an "overreach" given the strong relationship between the English words and their non-standard forms.

A second approach for solving the non-standard spelling challenge is to consider it initially as conventional spell-checking. The method tries to correct the misspelled words based on a probability model [11]. The obstacle with the latter is that the difference between non-standard and standard spelling can be substantial. As example, the number "4" is often used instead of the preposition "for". Hence, words far from the correct normalization may be viewed as corrections.

Third, normalizing non-standard spelling can be treated as a speech recognition problem [11, 43]. In this approach, the texts are regarded as phonetic approximations of the correctly spelled message. The factor in inspiring this view is that many non-standard spellings are written based on their phonetic rather than normative spelling. However, this view is also an oversimplification of the texts' nature, as it contains non-standard spellings that are not merely phonetic spellings of the correct word. For example, writings with abbreviations (e.g., lol for a laugh out loud), truncated words (e.g., n for and), and this approach can not handle leetspeak (e.g., 4ever for forever).

The use of machine learning techniques for vector representations of words has been around since 1986, much thanks to the work of Rumelhart, Hinton, and Williams [17]. Vector representations can be used as features in supervised, natural language processing tasks to increase the performance of the classifier. More recently, Roweis et al. introduced a method called local linear embedding. This method computes low-dimensional, neighborhood-preserving embedding of high dimensional input. The approach is applied to generate a two-dimensional embedding of words that preserves their semantics [44].

Bengio et al. [45] used feedforward neural networks to generate a distributed vector representation of words. By predicting the next word giving the previous words in the context, the neural network learns vector representation of each word in its hidden layer. The method is extended by Mikolov et al. [46] to take into consideration the surrounding words, not only the previous words.

In a very similar context, Mnih et al. replaced the feedforward neural network with a restricted Boltzmann machine to produce the vector representations [47]. Dahl et al. and others continued this work by introducing a word vector representation variant that learns, for each word, a low dimensional linear projection of the one-hot encoding of a word. They did so by incorporating the projection in the energy function of a restricted Boltzmann machine [48, 49].

Finally, GloVe is one of the most successful attempts at producing vector repre-

sentations of words for string comparisons [50]. GloVe learns a log-bi-linear model that combines the advantages of global matrix factorization and local context window to provide a vector representation of words based on the word count. A vector similarity measure, for instance, the Euclidean distance, cosine similarity, or L_1 measure, may then be used to measure the similarity between two strings.

Traditional methods to normalize non-standard spellings only look at a way to retrieve the correct spelling and disregard the textual context of the word, such as words and phrases around the word. This method misses a central aspect of text understanding since the textual context definitely matters. As an example, if we consider a misspelling "brak". This non-standard word may refer to break, as in a pause, or brake, as in braking a car. Whether it relates to the first or the latter all depends on the textual context. In the sentence "I want a brak 4life", it probably refers to break as in pause, but in the sentence "My bike braks stopped working", it probably refers to bike brakes. The context matters, but is not built into most methods to normalize non-standard spellings.

On the other hand, word embedding methods respect the context. In fact, words sharing the same context (i.e., similar in meaning) will have a similar representation. We can see that by plotting the embeddings of related, but distinct, words, such as "king" and "queen", and both will in an embedding plot be located nearby. In the same plot, completely unrelated words, such as "king" and "food" will appear far away. However, word embedding methods do not take into account a second constraint, namely that non-standard spellings of a word should also have similar vector representations. They calculate the vector representation based on the context of a word (surrounding words) so that only word sharing the same context (i.e., similar in meaning) will have similar representation. However, the phrase "laughing out loud" rarely appears in any social media post. Instead, we often see "lol". A traditional word embedding approach will not make the connection between the two. Thus the need to answer RQ4.

2.3 Social media analysis in crisis situations

2.3.1 Machine learning for topic detection in social media

In this Section, we summarize the main literature on machine learning algorithms for classifying social media messages, specifically for emergencies. For a broader overview of machine learning applied to social media analysis in general, we refer the reader to the survey by Ifran et al. [51]. Machine learning approaches for social

media analysis during emergencies can be divided into supervised, unsupervised, and semi-supervised.

The supervised learning algorithm learns a predictive function that represents the relation between the features and particular crises. In this way, the algorithm can classify any new unknown message to be part of one of the predefined crises-categories. Several approaches are available for the classification of crisis-related social media with a different order of computational complexity including Naïve Bayes, Support Vector Machine, Neural network etc.

Naïve Bayes is a family of probabilistic classifiers. When used for text classification, it proposes a probabilistic model based on assumptions about how the words in a text are distributed[1]. Then it uses the training data to estimate the parameters of the model. Naïve Bayes is a relatively simple classification method but has shown remarkably good performance in text classification, especially when there is a limited training data. In social media, it was used for topic detection [26] and sentiment analysis [52, 53].

A Support Vector Machine (SVM) is a kernel-based algorithm that draws a hyperplane dividing the space into subspaces. When used for text classification, each subspace typically contains vector representations of text belong to the same category. Like Naïve Bayes, SVM does not require a big training set, but it requires more computational resources than Naïve Bayes. SVM is one of the most experimented techniques for text classification. In social media, SVM is applied in message classification into crisis an non-crisis related topics [26, 54].

Neural networks are techniques inspired by the the human brain. Text classification has benefited from the recent development of deep learning architectures and their potential to reach high accuracy with less need for engineered features. Nguyen et al. [55] used a Convolutional Neural Network to classify social media posts by the type of information they provide (e.g., affected people, infrastructure damage, sympathy and support). Rosenthal et al. [56] used Long-Short Term Memory (LSTM) in a neural network for sentiment analysis on Twitter during a crisis. Their goal was to classify posts with regard to the sentiment they express (positive, negative, or neutral)

Unsupervised methods are used to identify patterns in unlabelled data. They are most useful when the information seekers do not have a labeled data, or if obtaining such data would prove to be costly.

Nearest neighbor algorithms are regularly used in an unsupervised setting for clustering, i.e., to group a set of data points in different clusters based on a similarity

[1] It typically uses individual words as the basis, but other variants are also common such as n-gram.

measure. In this context, Locality Sensitive Hashing (LSH) is the nearest neighbor algorithm designed to hash data points into so-called buckets so that data points sharing similar patterns will, with high probability, be located in the same bucket. Rogstadius et al. used it to cluster social media messages into crises-stories based on the keywords used in the message [57].

Another popular unsupervised method is soft-clustering methods that allows an item to belong to several clusters with varying degrees simultaneously. In the social media domain, Kireyev et al. used Latent Dirichlet Allocation (LDA), including a weighting scheme that accounts for the document of the words in the tweet, as well as for the length of the tweet for crisis-related topics extraction [58].

Semi-supervised or hybrid methods combines both supervised and unsupervised techniques. They are most useful during the early stages of an event where supervised techniques are challenged by the lack of labeled data for that particular event. As an example, Mazloom et al. [59,60] combined matrix factorization and k-nearest neighbors with Naïve Bayes and Alam et al. [61] used a graph-based deep learning framework to learn a semi-supervised model for tweet classification during an emergency.

2.3.2 Social media analysis platforms in crisis situations

Automated and semi-automated categorizations of social media face several challenges. Among those challenges is the data size, the diversity of sources, and the unstructured nature of social media. However, the overload of information in social media makes manual analysis cumbersome at best and urges the use of automated analysis platform. Tweak-the-Tweet [62], for example, is a social media analysis tool that tries to overcome these challenges by proposing a new syntax to writing tweets in time of crisis. The syntax has a specific predefined keyword set that can be used to identify categories of the message. Ushahidi [63] and CrisisTracker [57] also try to avoid hurdles of social media analytics but through a crowdsourcing platform. In these platforms, local volunteers and observers report the needs and risks they witness/face during an emergency. CrisisTracker has additional functionality to cluster the collected messages onto different discussion topics using LSH. SensePlace2 is yet another platform that extracts geographical and temporal features from tweets to present them in a layered map [64].

Some platforms have fully automated analytical functionality. AIDR [65] uses Random Forest to classify social media posts into a set of predefined crisis related topics such as donation, damage, and casualties. Tweedr [66] uses Logistic Regression for the same purpose. EMRSE [67] uses SVM, and ESA [26] uses a combina-

tion of Naïve Bayes and SVM – all for automatic categorisation of tweets.

Tweedr goes a step further and in addition uses conditional random fields to extract text tokens from the social media messages that can be related to numerical value, building, and transportation. The token are used to report specific information about different classes of infrastructure damage, damage types, and casualties. ESA provides a keyword burst detector that generates alerts when the frequency of a specific word increases. It can be used to look for a burst of words such as "shooting" without human intervention. TweetCred [54] uses Ranking SVM to classify social media posts by credibility to favor credible posts in a time of emergency. Twitris [68] is more of a sentiment analysis platform for public sentiment perception during an emergency. It uses lexicon-based classification to classify social media post into factual, subjective, or emotional.

Table 2.2 summarizes the most notable social media analysis platforms dedicated to crisis management found in the scientific literature.

Table 2.2: Social media analysis platforms in the academic literature

Analysis platform	**Data**	**Dimension of categorization**	**Machine learning approach**	**Reference**
AIDR	Twitter	Information provided	Random forest	[65]
Tweak-the-Tweet	Twitter	Information provided	Does not use machine learning	[62]
Ushahidi	Twitter, RSS feed, Email, SMS	Information source	Does not use machine learning	[63]
CrisisTracker	Twitter	Information source and Information provided	LSH	[57]
TweetCred	Twitter	Credibility of the information	Ranking SVM	[54]
Twitris	Twitter	Factual, subjective, or emotional content	Lexicon-based classification	[68]
SensePlace2	Twitter	Time and location	Does not use machine learning	[64]
EMERSE	Twitter, SMS	Information provided	SVM	[67]
ESA	Twitter	Information provided	Naïve Bayes and SVM	[26]
Tweedr	Twitter	Information provided	Logistic regression	[66]
Twitcident	Twitter, TwitPic	Information provided	Does not use machine learning	[69]

2.4 Addressing the emergency manager's requirements

This section summarizes the findings from **Paper A** in which we answer RQ1. The paper presents the results of a two-hour semistructured personal interview we conducted with representatives from four prominent local authorities on May 23, 2017: The chief of staff at Agder police district, the crisis preparedness leader at Kristiansand municipality, head of a unit at the Grimstad fire brigade, and a volunteer in Grimstad Red Cross. The interview was oriented to discover the current crisis response procedure, information needed during this process, the practice in information gathering, and how the EMSs currently use social media and their opinions about its potential.

From this interview, we realized that several problems need to be solved for social media analysis platforms to become an efficient tool for EMS personnel. The most prominent are:

- Credibility: Social media analysis platforms should establish a trusted network of people that delivers the information.
- Quality of information (QoI): Social media analysis platforms should ensure that the platform treats only high-quality messages shared by the network.
- Information requirement: Social media analysis platforms should analyze social media data in ways that provide relevant answers to the question asked by the EMS.

The first two points are in line with another research performed by Vieweg et al. [3] where they interviewed personnel from the United Nations Office for the Coordination of Humanitarian Affairs (UN OCHA) regarding their use of social media. The third point was also deduced by Palen et al. [70] in their review of the use of social media in crisis informatics.

This thesis main focus is on the third point: Information requirement. More precisely it focus on how to automate the information requirement process. Most of the tools available today, introduced in Section 2.3.2, analyze the data by extracting as much information about as many crisis-related topics as possible. Social media data should, obviously, be analyzed in ways that provide relevant answers to questions asked by the experienced EMS. Such inquiries usually trigger new data collection steps, which again yield further questions that need to be answered. A hypothetical example is a tool that finds out that there is a flood in an area. The natural step for the EMS is to ask the location of the disaster. If the flood has happened in a densely populated area, the next question might be to ask how many people are affected and

Table 2.3: Compliance of social media analysis platforms to the criteria specified by EMSs

Analysis platform	**Credibility**	**QoI**	**Information requirements**
AIDR	×	×	√
Tweak-the-Tweet	×	√	×
Ushahidi	√	√	×
TweetTracker	×	×	×
TweetCred	√	×	×
Twitris	×	×	×
SensePlace2	×	×	×
EMERSE	×	×	×
ESA	×	×	×
Tweedr	×	×	×
Twitcident	×	×	×

how seriously. Simply bombarding EMS with all information at once causes noise, and might even be one of the reasons social media analytics is not used to the extent that it can be. For a social media analysis platform to be efficient in a crisis, it needs to focus on answering questions and requirements raised by the EMS personnel.

Table 2.3 shows how the current social media analysis platforms, discussed in Section 2.3.2, comply to the criteria discussed in this section – the emergency manager's needs. Many of the available tools are solely data-driven classification of social media messages. The table shows the gap that still exists between what the social media analysis platform can provide and what the EMSs require. Regarding the information requirement, a platform like AIDR gives the EMS the option to define the label or topic of interest for the classification. However, this process is manual and not automated, which means that for each emergency, the EMSs need to define the set of topics they require. Besides, Rudar et al. [71, 72] proposed a method that helps EMS fulfill their requirements with different levels of granularity. The method is based on identifying clusters of tweets describing a sub-event of an emergency and generate summaries of those tweets. Nevertheless, the method does not ensure that the clusters of sub-event represent the EMS requirement given a specific emergency context. These findings trigger RQs2-3.

2.5 Link prediction problem

2.5.1 Automated information requirement identification during an emergency

In Section 2.4, we noted that the information gathering process is iterative, i.e., knowing a particular piece of information usually triggers new questions and the requirement for additional data collection. In this section we propose an a graph-based approach to answer RQ2.

EMSs rely on correct information in dynamic and chaotic situations. For example, in an extreme weather crisis, the knowledge that there are trapped persons triggers an enquiry about evacuation possibilities, and if the evacuated persons have what they need. The required knowledge can be modeled as a discrete graph showing that any information gathered leads to the need for further details and additional information. The graph is highly conditioned because much of the emergency response depends on the information at hand. The availability of information conditions the transition from a node to another. Therefore, the edges of the graph need to be weighted by that answer. To be more concrete, the requirement to obtain information on a location is typical in many crises. Knowing that a certain building is the location of an emergency, triggers collecting information such as the number of occupants of the building. Not knowing that location, will trigger the collection of a different set of information.

We compiled a graph representing this information gathering process, particularly in three areas: Fire, extreme weather, and public disturbance. The nodes in the graph present the type of information needed by emergency manger during a crisis. For a fire emergency, the sub-graph is extracted from the work of Nunvath et al. [73] with the collaboration of the main author who did an extensive interview of firefighters about the type of information they need during an indoor fire crisis. The sub-graph for extreme weather and public disturbance, as well as the rest of the graphs, were vetted by two policemen from Oslo police station. Figures 2.1 to 2.3 show the graphs for indoor fire, extreme weather, and public disturbance respectively. The graphs are connected to a big graph where many of the nodes overlap (e.g., location, people affected). The graph is conditioned because the steps taken from a node to another depend on the information available up to the current node.

It is nearly impossible to model a graph for each type of crisis and each possible situation in every emergency. Such a model will be immensely huge and include all possible conditions. Then, the question becomes twofold: Can this process be automated? In other words, can we design a system that, given, the available infor-

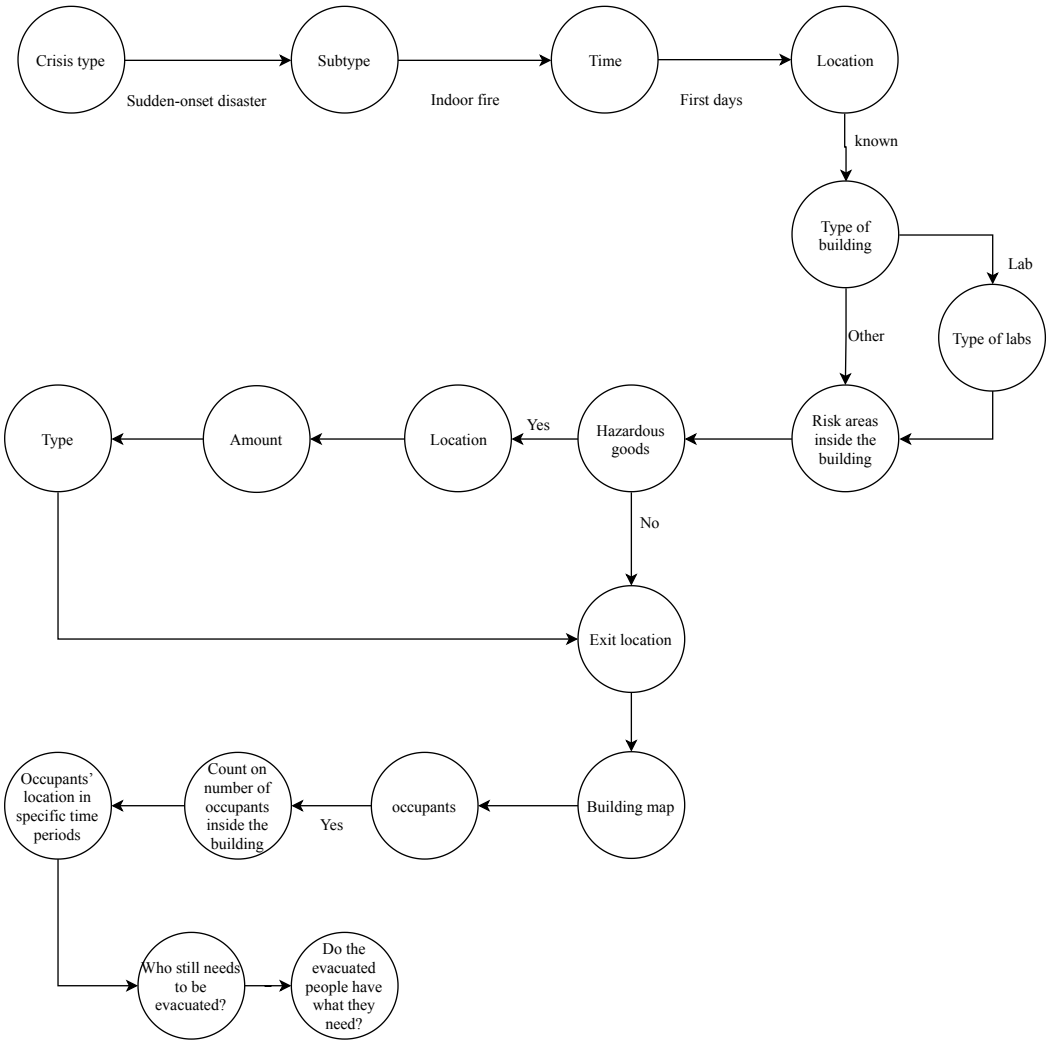


Figure 2.1: Example of an information requirement graph during an indoor fire emergency.

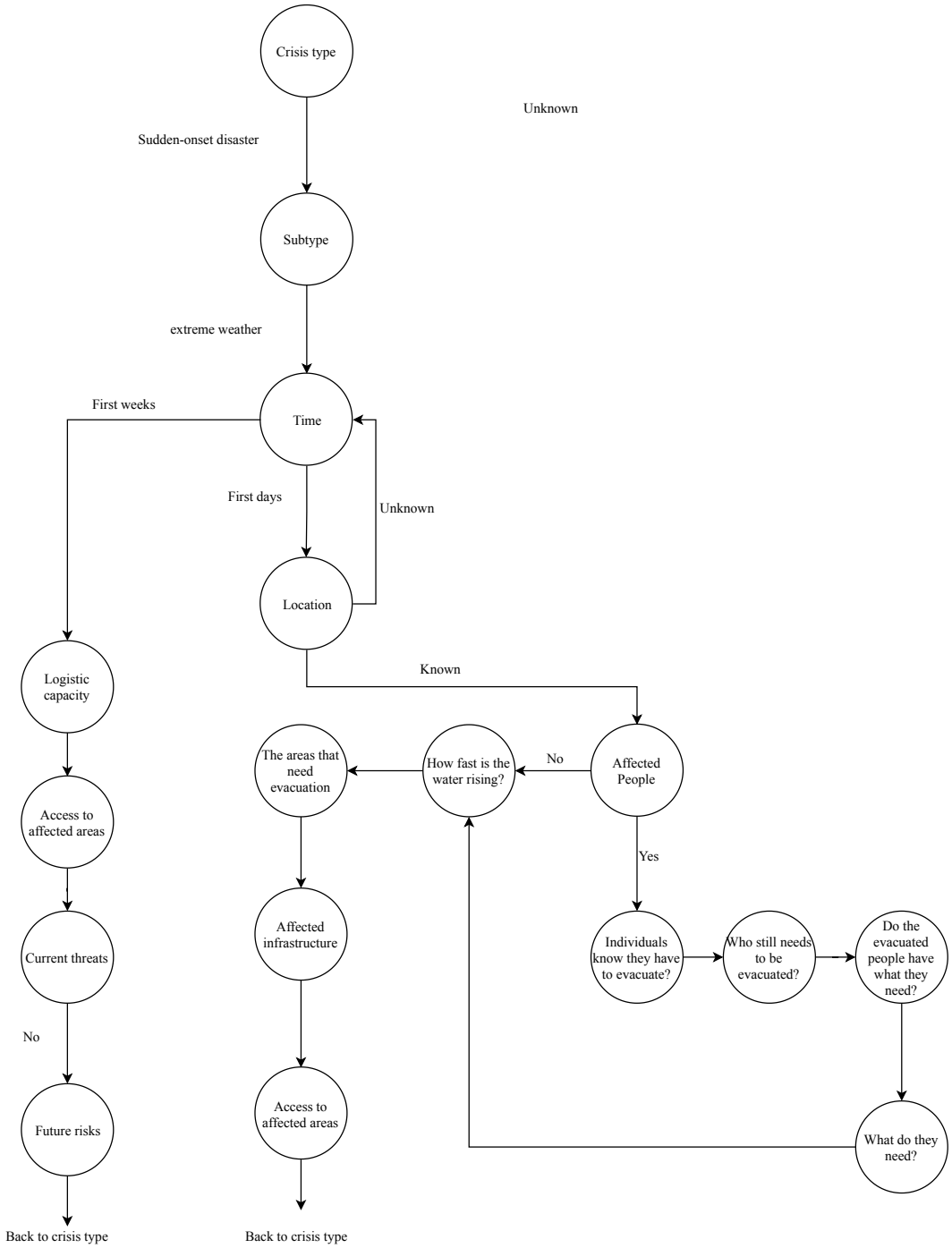


Figure 2.2: Example of an information requirement graph during an extreme weather emergency.

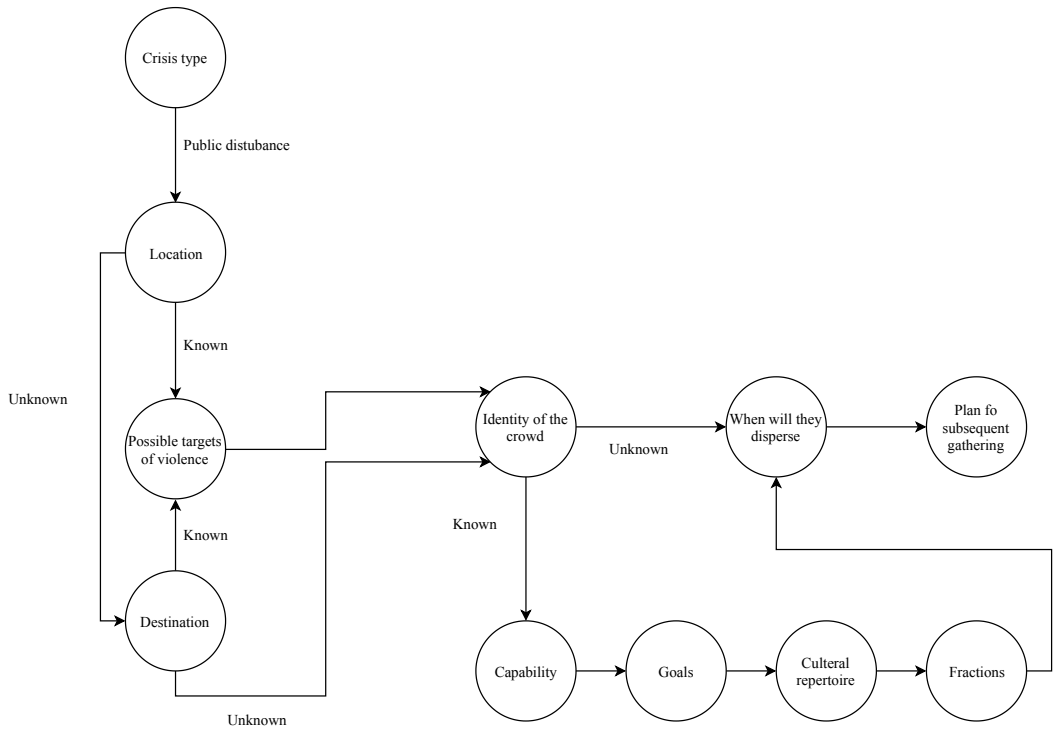


Figure 2.3: Example of an information requirement graph during a public disturbance emergency.

mation, will infer the following needed information. If so, what happens if we only have a graph for a limited set of situations: can that generalize to other crises and other conditions in the specific crisis?

The assumption is that if we know the following required information Q given the available information for a crisis A, then, for a similar crisis A', the information needed next is similar to the information required in Q. In this case, the problem becomes that of link prediction in the sense of inferring missing links from an observed graph. The expectation is that it is possible to learn general crisis trends from a few emergencies.

2.5.2 Background on link prediction

Consider a graph $G = (V, E)$ where V is the set of nodes and E the set of links. Let U be the collection of all possible connections in G ($|U| = |V|(|V| - 1)/2$). In terms of crisis management, G can be a specific type of crisis, V can be nodes of needed information for that crisis, and E can be information that leads to the next information need. We assume that there are missing links, or links that might appear in the future in G from the set of non-observed links $U - E$. Link prediction is the task of finding these links. Three main families of approaches have been

used to solve the link prediction problem [74]: The similarity-based algorithms, the maximum likelihood algorithms, and the probabilistic algorithms.

The simplest algorithm for link prediction is the similarity-based algorithms. In this approach, each possible pair of nodes have a similarity score. The likelihood of linking a pair of nodes is ranked according to their similarity scores. The links between nodes with the highest similarities are supposed to have a high likelihood of existence. Despite its simplicity, the definition of node similarity is not a trivial issue. Similarity measures can be very simple or very complex without any guarantee that it will work or fail for a certain graph [75]. Often used similarly measures include common neighbours [76], Jaccard index [77], Katz index [78], and superposed random walk [79].

The maximum likelihood algorithms presuppose a particular organization of the network structure. This organization is modeled with a set of parameters obtained by maximizing the likelihood of the observed structure. After that, the possibility of non-observed links can be calculated using those parameters. Even though the maximum likelihood algorithms give some insights into the network organization, the empirical results show that there is no gain in accuracy in comparison with the similarity-based algorithms [75]. The maximum likelihood algorithms include the hierarchical structure model [80], and the stochastic block model [81].

Finally, the probabilistic algorithms intend to abstract the structure of the observed graph and then predict missing links using the learned model. The probabilistic algorithm optimizes a target function to learn a model composed of a set of parameters θ, which fits an observed graph G. The probability of existence of link between two nodes i and j ($A_{i,j}$) in graph G is estimated by the conditional probability $P(A_{i,j}|\theta)$. Probabilistic algorithms include the relational Bayesian network [82], the probabilistic entity-relationship model [83], and the stochastic relational model [84].

All of the previously cited link prediction applications do not consider the case in which the edges depend on, or are conditioned by, an external input: so-called conditional graphs. A typical example of a conditional graph is the graph represented by an FSM. An FSM has a structure that exhibits a syntactic and semantic meaning, which often is cyclic, and with transitions between nodes dependent on external input. If we only have an FSM that represents a part of the system, the problem becomes challenging to model. We want to complete this FSM by inferring new links and this way making it fully descriptive of the system. To use the traditional link prediction solution is challenging because the complete graph depends on the external input. A graph where some links are missing, or not known

to exist, is a model of crisis information retrieval.

Furthermore, the process of inferring links in a conditional graph depends on a series of past nodes. For example, in the case of a crisis, the next information to be gathered depends on the series of previous details available. Such a series of past details often have a variable length, which makes the problem a variable-length input problem. The already collected information puts some constraints on any link prediction algorithm for a conditional graph. First, it needs to be a variable input algorithm. In the case of neural networks, the recurrent neural network is designed to handle such input. Second, it needs to keep a memory of the past series of nodes. Thus, the recurrent neural network has to preserve "a certain kind of memory", such as an LSTM or a neural Turing machine (NTM).

2.5.3 Neural Turing machine (NTM)

An NTM has a neural network, called the controller, and a two-dimensional matrix referred to as the memory (see Figure 2.4). The controller is a feedforward or recurrent neural network that can read from and write to selected memory locations using read and write heads [85]. The read head $w^r(t)$ and the write head $w^w(t)$ have the property described in equation 2.1.

$$\sum_i w_i^r(t) = \sum_i w_i^w(t) = 1. \tag{2.1}$$

Let $M(t)$ be the $n \times m$ memory matrix at time t. To read values from M, we need an addressing mechanism that dictates from where the head should read. A read operation is defined as the weighted sum over the memory rows $M_i(t)$:

$$r(t) = M(t)^T w^r(t). \tag{2.2}$$

The write-operation is composed of an erase-operation and an add-operation. The erase-operation deletes particular elements from the memory $M(t-1)$ using an erase-vector $e(t) \in [0,1]^m$. The add-operation replaces the deleted values with elements from an add-vector $a(t)$. Thus, the write-operation can be expressed by the following equation where $\circ$ is the element-wise multiplication:

$$M(t) = M(t-1) \circ [1 - w^w(t)e(t)^T] + w^w(t)a(t)^T. \tag{2.3}$$

There are two types of addressing methods used to calculate the weight-vectors $w^r(t)$ and $w^w(t)$: content-based and location-based addressing. The content-based addressing selects the weight-vectors based on the similarity between a row in the

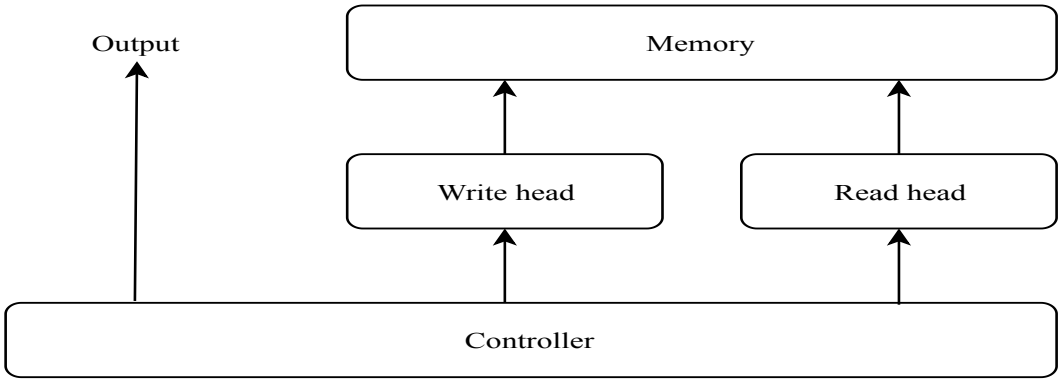


Figure 2.4: An NTM block

memory-matrix and a given query $k(t)$ generated by the controller. The location-based addressing rotates through the element of the memory.

The NTM is designed after the Turing machine introduced by Alan Turing [86][2]. In the previous Section, we noted that a conditional graph could be used to describe an FSM. On the other hand, an FSM is a spatial case of the Turing machine. The NTM, on the other hand, is a recurrent neural network with external memory. It seems only evident that our approach will lead to tuning the NTM to solve the link prediction problem in a conditional graph.

To summarize, in this chapter, we identified two main research gaps. First, the non-standard spelling issue in social media which triggers RQ4. Second, how to best address the EMSs information requirements from social media and enhance their situation awareness which triggers RQ3 and RQ5.

[2]Alan Turing called it the automatic machine or the a-machine

Chapter 3

Approach

> *There is scarcely a subject that cannot be mathematically treated and the effects calculated or the results determined beforehand from the available theoretical and practical data.*
> ***– Nikola Tesla in "My Inventions"***

This Chapter presents the approach and innovations to address the research gaps identified in Chapter 2. Section 3.1 presents our approach to solve the non-standard spelling issue in social media. It summarizes the approach used in **Paper B**, also applied in **Paper F**, and answers RQ4. Section 3.2 continues by presenting the classification method of social media post. The method was used to classify social media posts in both **Paper D** and **Paper E**. The social media posts are classified into topics related to information needed by EMSs. The same topics are used as nodes in the conditional graph. Section 3.3 describes our link prediction algorithm to for conditional graphs. It summarizes the method used in **Paper C** to answers RQ3. Finally, Section 3.4 describes how all the components are integrated to produce an information retrieval framework useful for EMS during a crisis. This section summarizes the approach used in **Paper D** and answers RQ5. For a more detailed description, we refer the reader to the papers mentioned above.

3.1 String metric over word space

To address the non-standard spelling gap described in Section 2.2, we developed the function F that maps words into real vector space $\mathbb{R}^n$. The function is designed so that two similar words (i.e., non-standard spellings of the same word or words used in the same context) will be the shortest distance between the corresponding

mapping in the real vector space. F obeys two constraints. First, the distance in real vector space between the mapping of a word and its non-standard versions must be shorter than the distance between that word and non-standard versions of other words. Second, the distance in real vector space between the mapping of words with similar meanings must be shorter than the distance between words with different meanings.

For a vocabulary of words A, our approach starts by initializing the embedding of each word with a function $u : A \longrightarrow \mathbb{R}^n$. u produces an initialization matrix U. Then, to implement the first constraint, we use a denoising autoencoder. A denoising autoencoder is a neural network that takes as input a vector with added noise and tries to reconstruct the original vector. By doing so, it captures features and patterns in the vectors. Here we treat the non-standard version of the word (e.g., misspelled version of the word) as a noisy variant of the standard spelling of the word. The denoising autoencoder should be able to capture relational patterns of the non-standard and standard versions of the same word in its hidden layer. Hence, it should obtain how non-standard and standard writing of the same word is related.

For the second constraint, we use a neural network to predict a word in the sentence given its context (i.e., surrounding words). The word embeddings are then the weight matrix in the first layer of the neural network. These embeddings are learned to maximizes the log-likelihood of predicting the correct words, which will ensure that they contain patterns about the relationship of the word and its context. We call this part a context encoder. Figure 3.1 shows the overall neural network architecture of the model.

The function F allows the definition of pseudometric D_c over a word space. The distance between two words a_i and a_j using D_c is defined by:

$$D_c(a_i, a_j) = d(F(a_i), F(a_j)), \tag{3.1}$$

where d is a metric in $\mathbb{R}^n$.

In **Paper F**, we use the string metric D_c to detect peaks of tweets related to an event and filter out randomly occurring peaks caused by activities that stand out from normal daily usage of social media. The random peak problem affects event detection methods from social media based on peak detection.

The paper assumes that if an event occurs, the number of similar posts in the cluster of posts following the event is higher than in the case of a non-event. Here, we use the string metric D_c and propose a similarity threshold. Two tweets that have a similarity measure higher than the defined limit are considered similar. The proportional decrease in the number of similar posts in a specific cluster as the

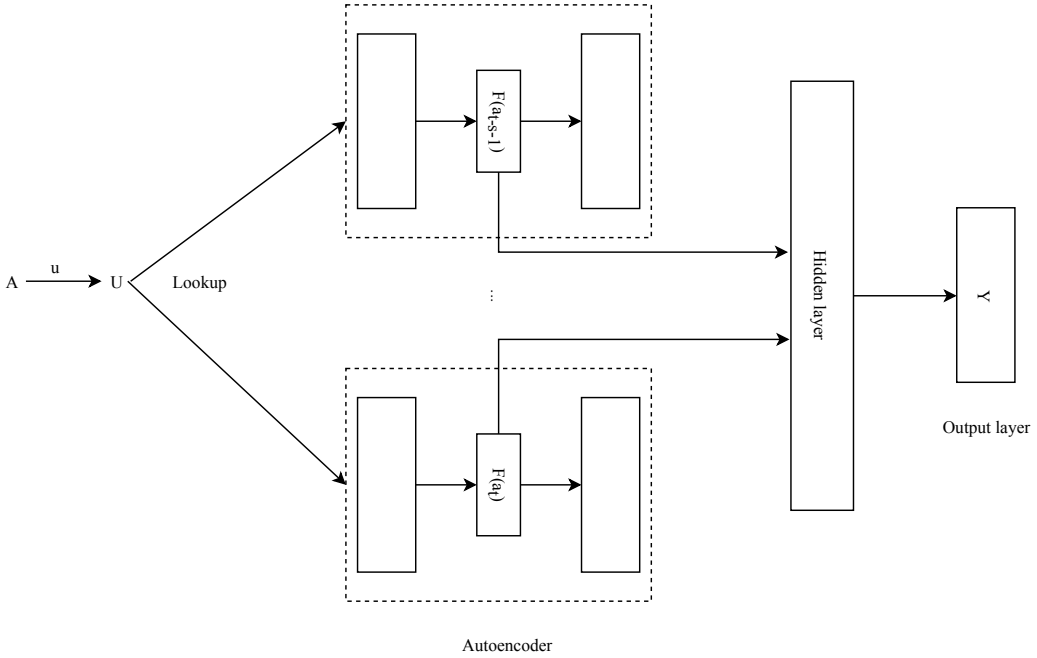


Figure 3.1: Neural network architecture of the autoencoder in combination with the context encoder to find the similarity between the words

similarity threshold increases is referred to as decay. We use that decay as a part of a low pass filter to detect which cluster corresponds to an event. The low pass filter is a function SDF (Semantic Decay Filter) that eliminates a random peak in the number of tweets and preserves significant peaks. Therefore, the SDF function should only peak around the time of a significant peak. The SDF function is defined in Equation 3.2:

$$SDF_t = \alpha Norm(|T_t|) + \beta Norm(\lambda_t^k) \tag{3.2}$$

where λ_t^k is the decay at a threshold a_k for a cluster of tweets collected during a time window t, T_t is the set of tweets in the cluster of collected during a time window t, $Norm$ is a normalization function, and α and β are real number.

3.2 Classification of social media messages

This section presents our approach for the classification of social media messages into the different topics that EMSs require during a crisis. We call this classification "topic detection" (TD), which is composed of three sub-components:

- A named entity recognizer (NER), which extracts metadata from the text it receives. The information is the location, number of injured, killed, and missing persons.

- A word embedding that transforms the text into a vector.
- A neural-network-based detection layer that combines the word embedding and the information extracted by the NER to produce the probability distribution over topics.

The NER uses a maximum entropy classifier developed by [87]. It is a statistical classifier created to identify named entities by examining each word in a sentence. It subsequently decides whether this word is the start of a named entity, the continuation of an already started entity, or not part of any named entity at all. The NER is responsible for detecting the presence of words and phrases such as the number of injured, killed, or missing persons. For the word embedding, we used the function F introduced in Section 3.1.

Finally, a feedforward neural network takes as input the information extracted by the NER and the word embedding, and outputs the probability of the message belonging to a specific topic,

$$o = p(topic|(a, F(m))) = \frac{e^{z_i}}{\sum_j e^{z_j}}, \tag{3.3}$$

where a is the information extracted by the NER, $F(m)$ is the embedding of message m, and z_i is the weighted sum of the output of the last hidden layer. Hence, the overall output is the probability that a message is related to a specific topic given information a and function of the message $F(m)$. The output of the topic detection is composed of the tuple <*topic, number of injured, number of killed, number of missing*>. Figure 3.2 illustrates the neural network architecture of the classifier.

It is important to stress that the objective of this work is not to design a state-of-the-art classifier, but rather to construct an automated information retrieval framework that predicts the information required by the EMSs, which is the topics of the classification. We used off-the-shelf components including our developed word embedding to construct a robust classifier that compares with other classical classifiers.

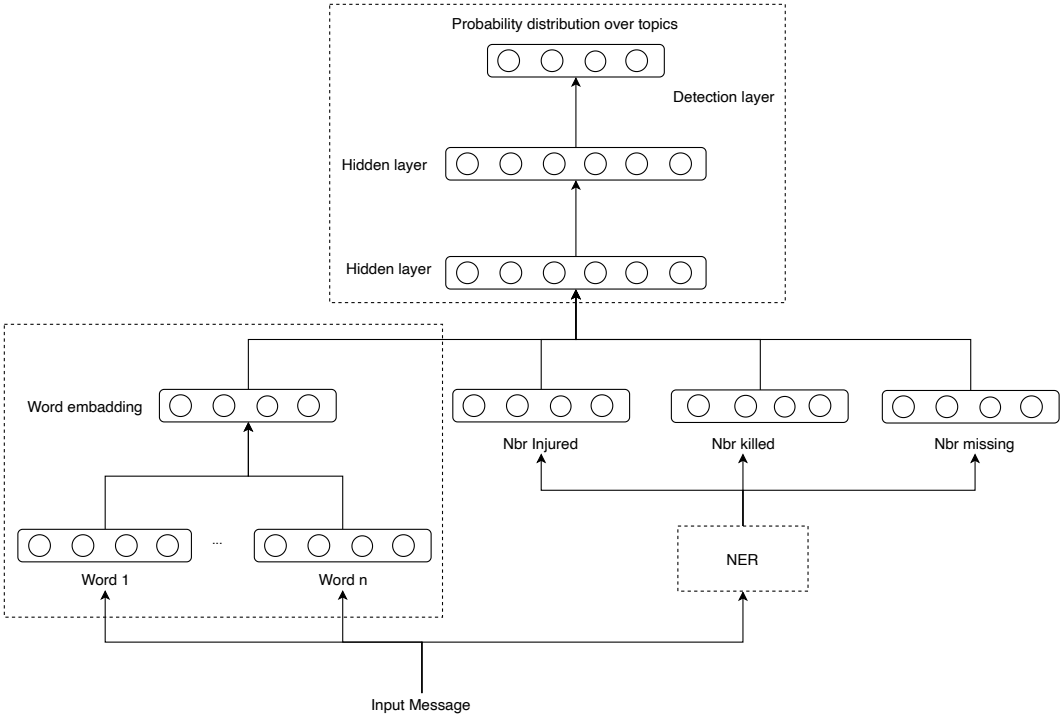


Figure 3.2: Neural network architecture for message classification

3.3 The link prediction problem

This section extends the existing NTM for conditional graphs. We call this unique algorithm the Conditional Neural Turing Machine (CNTM). The overall objective is to compose a neural Turing machine that predicts links in graphs with the conditional transition. In a conditional transition graph, transitions from one node to another is conditioned by an external component, such as assembled knowledge from social media. Such a graph is composed of:

- A finite set Q of nodes or states.
- A finite set C of input. C can be a set of logical propositions p_i that can be true or false, or a vector of logical propositions. In the context of this paper, C is a set of variables $c_i \in [0, 1]^n$.
- A transition $\delta : Q \times C \longrightarrow Q$ from a node to the next.
- A final node or state F.

The rest of this section focuses on how to model the transition δ using the CNTM. The first step of the CNTM is to produce a coding U given the current context v and the sequence of previous contexts. The context v is a combination

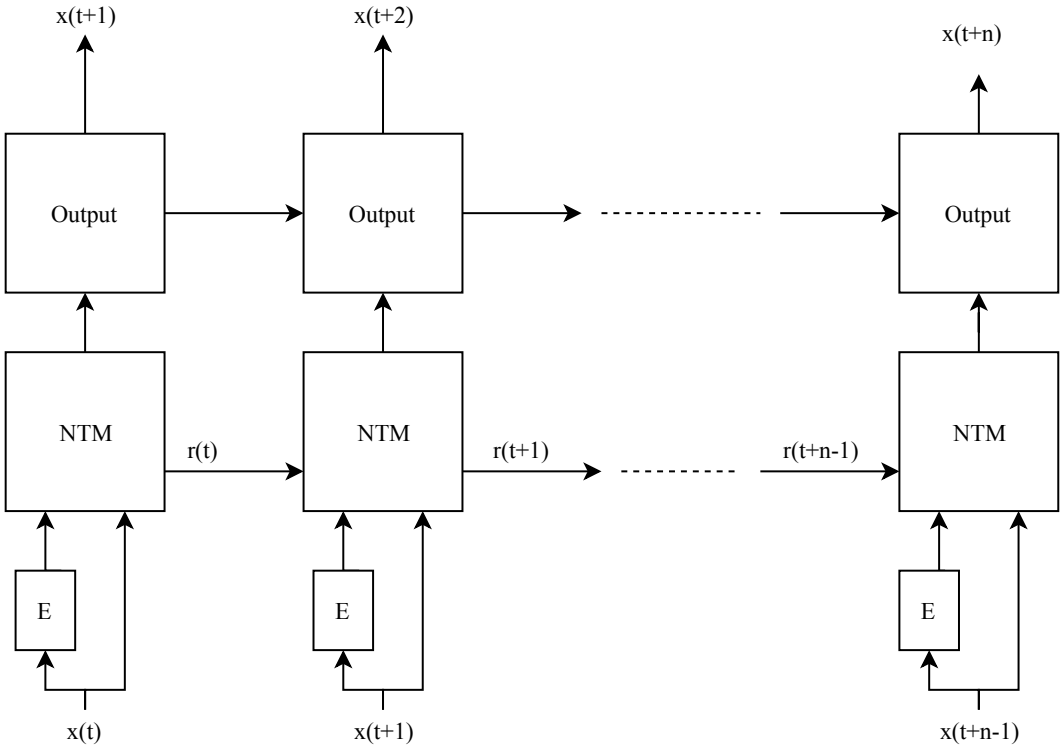


Figure 3.3: Architecture of the CNTM

of the preceding node in the graph and the external input $c \in C$. The output of this block is implemented using a neural layer. As input, it takes the output of the controller $h(t)$, and the information read from the memory $r(t)$, and calculates a linear combination between them. Thus the activation function for that output layer is a linear activation:

$$U = W_1 * h(t) + W_2 r(t) + b. \tag{3.4}$$

In the second step in the CNTM, the transition δ from a node to the other is implemented using the output layer. The output layer's role is to produce the next node, $x(t+1)$, in the graph given the previous set of coding of the context produced by the NTM block. At each time step t, the output layer takes as input U. The output at time t is consequently a probability distribution over the nodes of the graph $P(y|U, \beta)$, where β is the parameters of the output layer. In our proof of concept implementation, the output layer is implemented using an LSTM [88] with a Softmax output layer. Figure 3.3 shows the architecture of the CNTM.

3.4 Iterative information retrieval from social media

Let us consider an incomplete graph in which the nodes represent an information query needed by an EMS. The availability of the information required by query a in social media conditions the transition for the query a to a query b (or node a to node b). As an example, query a could be "is there human injuries", and query b could be "what is the severity of the injuries?" Whether or not the EMS should go from the state of asking if there are injuries, to the state of collecting injury statistics, depends on (or is conditioned by) the answer to query a.

This graph is similar to the model introduced in Section 3.3, where the set of inputs C is the output of the topic classifier introduced in Section 3.2. The CNTM can be understood as a query generator (QG), and the classifier is a topic detector (TD). The next query produced by the QG depends on the current query, and the output of the TD. Thus, this information gathering process is iterative, and the framework integrates both components in an iterative information retrieval system.

The communication between the components of the framework is shown in Figure 3.4: The QG queries the TD via its interface, and gives the ID of the requested topic as input to that query. The TD sends a feed request to the social media engine (SME). The SME's role is to connect to a social media platform and produce a feed of social media messages. The TD obtains the feed returned and classifies each message in the feed into the topics. If a message belongs to the topic of the request sent by the QG, the TD queries the QG via its interface and gives *<TRUE, number of injured, number of killed, number of missing>* as input. The input of the query is *<FALSE, number of injured, number of killed, number of missing>* otherwise. The QG will then generate a new query. All the messages belonging to the query topic, along with the metadata extracted by the NER will be sent to the EMS. Note that, in our current approach, we limited the feedback provided by the TD to a Boolean variable, which indicates if the required topic was found in the social media. This procedure offers proof of the concept of the framework, and this constraint can be lifted to include other attributes in the future.

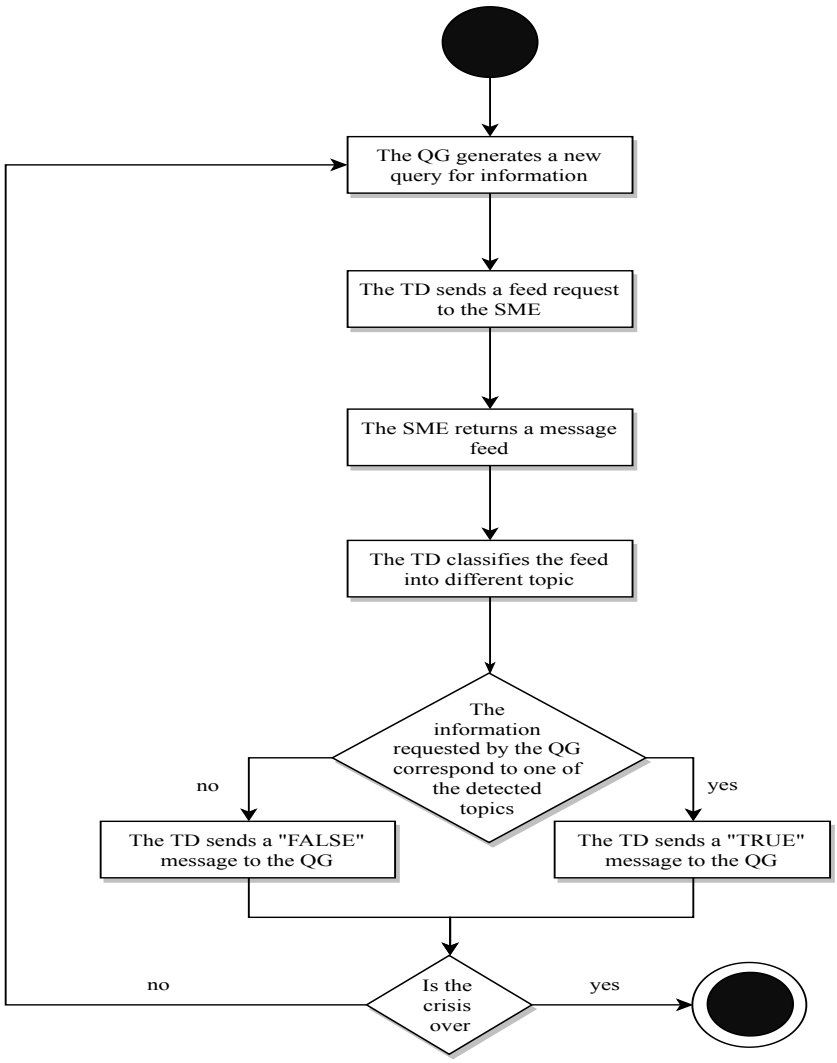


Figure 3.4: Flow diagram of the framework

Chapter 4

Evaluation and Discussion

> *We have trained them. Their knowledge of their tools is purely empirical; and they have a firm belief in the mummery that surrounds them.*
> ***–Isaac Asimov in "Foundation"***

In previous chapter, we introduced three components to address the identified research gaps, namely a word embedding approach, the CNTM, and the overall framework. In this chapter, we use two evaluation techniques to test these components. First, we perform an empirical analysis using diverse data sets to evaluate the word embedding and the CNTM separately (Sections 4.1, 4.2, and 4.3). Then, we perform a qualitative analysis to evaluate the overall framework with a survey of EMS personnel (Section 4.4)[1].

4.1 Evaluation of the string metric

To verify the string metric approach (see Section 2.2), we used a data set composed of the 1 051 most frequently used words on Twitter paired with their misspellings. The original usage of the data was in an IBM data normalization challenge in 2015.

For the context-based encoder, we used a data set with 97 191 unique sentences, including the vocabulary in their standard and non-standard form. The data is imbalanced which means that some words have only one non-standard equivalent, while other words have multiple non-standard ways of writing the same word. This imbalance could introduce challenges since an autoencoder might not recognize features in words with only a few non-standard versions.

[1]The data and codes used in this chapter can be found here for the string metric, and here for the CNTM.

Table 4.1: Performance comparison including our approach: D_c the combination of autoencoder and context encoder

Distance	**Closest word**	**5th closest word**
Cosine Similarity	46.33%	60.22%
Q-Gram	47.57%	62.41%
Srensen-Dice coefficient	47.85%	60.03%
Edit distance	55.75%	68.22%
Weighted-Levenshtein	55.85%	67.93%
Damerau-Levenshtein distance	56.51%	68.03%
N-Gram	58.23%	76.49%
Metric-Longest Common Subsequence	60.89%	75.73%
Longest Common Subsequence	61.37%	74.31%
Normalised-Levenshtein	63.17%	78.30%
D_c with L_1 distance	76.37%	81.53%
D_c with Euclidean distance	82.71%	87.35%
D_c with Cosine similarity	**85.37%**	**89.61%**

Table 4.1 compares the results produced by the distance metrics D_c (the distance produced by our embedding – Section 3.1), with the string metrics present in the literature. The intent is to find the correct version of a non-standard spelling. The table shows an increase in accuracy from 63.17% for the best metric available in the literature (Normalised-Levenshtein) to 85.37% when using our proposed D_c from the context-aware autoencoder. The reason is that unlike the state-of-the-art metric, D_c captures stochastic word patterns shared between the non-standard words and their correct form.

Figure 4.1 shows a scatter plot of the vector representation of the words after a dimension reduction of D_c reduced to two dimensions with T-SNE [2]. [89].

To test the low pass filter introduced in Section 3.1, we used three different Twitter data sets related to the STEM school shooting, London bride attacks, and Virginia beach attacks. We divided the time over which the data collection was made into windows t of length 1 hours for the Virginia beach attacks and the London bride attacks, and 5 hours for the STEM school shooting. We chose those windows for each data set based on the limitation of the required minimum number of tweets (no less than 2) present in each time window. Figure 4.2 shows the SDF function (Equation 3.2) as a function of time for the considered event. We chose the parameters $\alpha = 1$ and $\beta = -1$ in Equation 3.2. Those parameters guarantee that the SDF peaks around the significant peak. As the Figures show, SDF reaches a value higher than zero only around the significant peak for all the events and thus

[2]For further details on the results of string metric see **Paper B**.

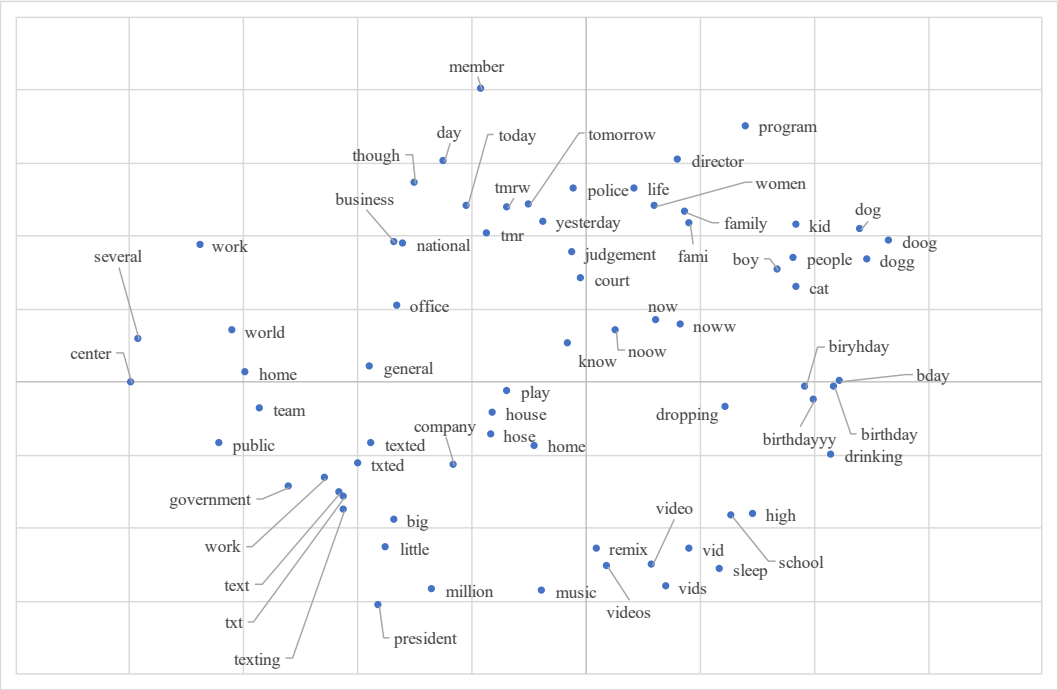


Figure 4.1: 2D plot of some example of vector representation of words using T-SNE

can be viewed as a low pass filter.[3]

[3]For further details on the results of the SDF see **Paper F**.

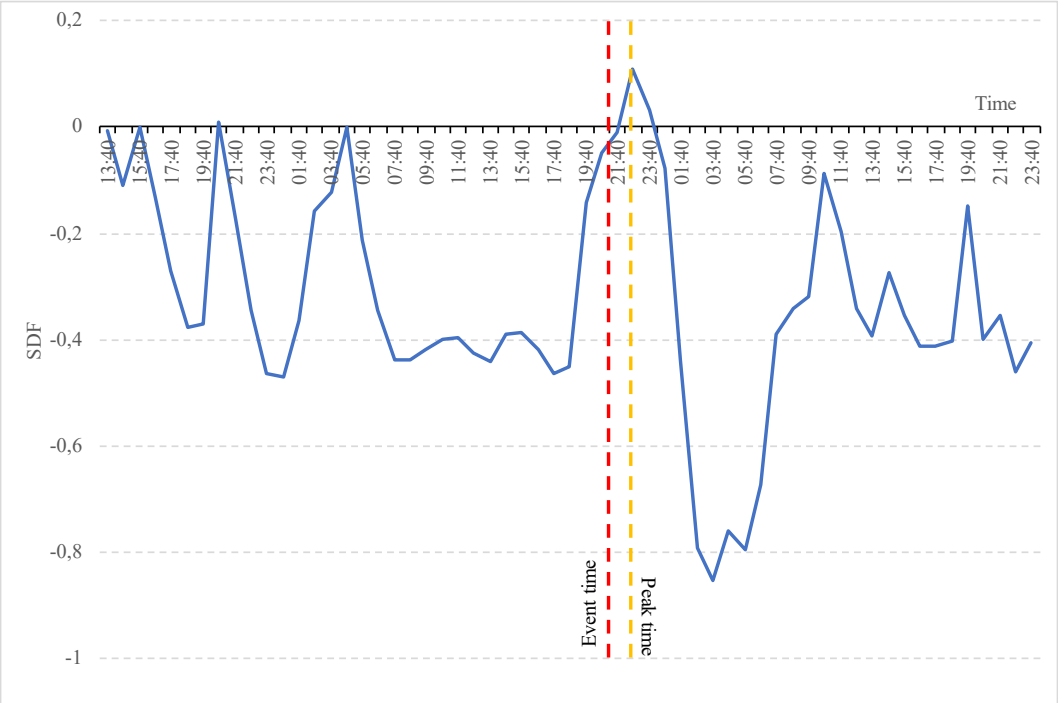


Figure 4.2: SDF as a function of time for the London bridge attacks: The SDF function reaches a peak (a value above 0) during the period of the peak containing event related tweets (22h40)

4.2 Evaluation of the social media classification

The string metric approach was further used as part of Twitter text classification. To verify the social media text classification approach (see Section 3.1 for details), we used a Twitter data set from *CrisisLex* [90] and *CrisisNLP* [91]. Both are platforms for collecting and filtering communications during a crisis. We selected data on extreme weather crises: The Alberta, Queensland crises, Typhoon Hagupit, and Cyclone Pam. The tweets are a mix of crisis-related tweets and tweets, which are in no way related to the ongoing crises. The percentage of unrelated tweets is 44% for the Alberta flood and 43% for the Queensland flood. In the first case, the Alberta flood, only 30% out of the related tweets, gave accurate, valuable information about the crisis. In the Queensland flood data, 48% of the related tweets had useful information. Such a mix is particularly challenging as both the unrelated tweets and the tweets without useful information, become noise to the classifier. The data was further labeled into the topics in the nodes presented in the sub-graph related to extreme weather (Figure 2.2). We used 70% of the data for training and 30% for validation. Table 4.3 compares the results of our classifier with other baseline classifiers, and Table 4.2 compares the results of the neural network classification

Table 4.2: Results of the neural network classification with a changing word embedding approach

	Alberta flood	**Queensland flood**	**Typhoon Hagupit**	**Cyclone PAM**
		Word2Vec		
Precision	0.82	0.83	0.86	0.90
Recall	0.77	0.70	0.81	0.82
F-measure	0.79	0.76	0.83	0.86
		Glove		
Precision	0.80	0.85	0.83	0.91
Recall	0.76	0.70	0.82	0.83
F-measure	0.78	**0.77**	0.82	0.87
		Our approach		
Precision	0.85	0.84	0.87	0.93
Recall	0.78	0.71	0.82	0.85
F-measure	**0.81**	**0.77**	**0.84**	**0.89**

presented in Section 3.1 with including context as part of the word embedding approach. Even though the objective of this work is not to design a state-of-the-art classifier, the off the shelf classifier we used can compete with basic and classical classifiers delivering satisfactory results. However, more advanced classifiers are already developed which show more robust results. For example, Nguyen et al. developed a convolutional neural network based classifier and tested on the same data set [55]. They reported, on average, a 0.88 F-measure but using different labels. Besides, the main obstacle we found in our classifier is the lower recall, which indicates that the developed approach fails to identify a proportion of the tweets related to the topic. [4]

[4]For further detailed for social media text classification see **Paper D** and **E**.

Table 4.3: Performance of the topic classification

	Alberta flood	**Queensland flood**	**Typhoon Hagupit**	**Cyclone PAM**
		SVM		
Precision	0.85	0.78	0.78	0.90
Recall	0.78	0.60	0.71	0.80
F-measure	**0.81**	0.68	0.74	0.85
		Logistic regression		
Precision	0.85	0.79	0.82	0.90
Recall	0.78	0.60	0.71	0.79
F-measure	**0.81**	0.68	0.76	0.84
		Random forest		
Precision	0.82	0.75	0.82	0.90
Recall	0.74	0.66	0.73	0.80
F-measure	0.78	0.70	0.77	0.85
		Our approach		
Precision	0.85	0.84	0.87	0.93
Recall	0.78	0.71	0.82	0.85
F-measure	**0.81**	**0.77**	**0.84**	**0.89**

4.3 Evaluation of the link prediction

Our information retrieval framework from social media becomes much more useful for emergency personnel when it used to guide responders to achieve a better situation awareness. For this to be possible, we need to verify the link prediction mechanism. We investigate the validity of the link prediction algorithm empirically in two stages, first in the theoretical and randomly generated model and later in with real models from emergency personnel.

We trained and validated our link prediction model with seven distinct data sets. The first six sets had 1000 randomly generated sparse conditional graphs with 10, 20, 40, 60, 80, 100 nodes each. The seventh data set is the graph presented in Section 2.5 and is collected from the information gathering process, particularly in three areas: Fire, extreme weather, and public disturbance. During the training phase, we exclusively trained the algorithm on graphs explicitly containing 70% of the links in the randomly generated graphs. During the test phase, starting from the initial node, the algorithm creates a path in the graph ending by the final node. The algorithm is deemed successful if the path it generates is available in the full graph. Hence, it is a success if it has learned a path that correctly is in the full graph. The accuracy is measured by the percentage of success the algorithm achieves over 100 tries, each with a random starting node and input from the external environment.

This might at first seem trivial but is genuinely far from it. Considering an initial condition, which in real life could be collected knowledge from social media, the model should be able to predict the most likely information to be gathered next; even if that link is not present in its training set for that specific emergency but rather in a similar situation in another emergency. Hence, it shows that the model is able to learn general trends in the graph, rather than memorizing a particular graph. This approach is a step towards learning overall crisis trends and transferring the learned steps from one emergency to the next.

Figure 4.3 shows the accuracy of the CNTM compared to the vanilla graph distance [92], and the LSTM in inferring the correct links for randomly generated conditional transition graphs. The table shows a clear advantage of using the CNTM over the other approaches. As can be expected, the larger the graph (i.e., the more nodes), the less accurate the predictions. For a graph with 100 nodes, the accuracy is 65.25%. However, as the number of nodes grows, the gap in performance between the CNTM and the other approaches grows exponentially: The difference between the CNTM and the LSTM starts with 2.6% for ten nodes graph and reaches approximately 23% for 100 nodes graphs. For the emergency information requirement graph (Section 2.5.1), The accuracy of the CNTM in inferring the correct links

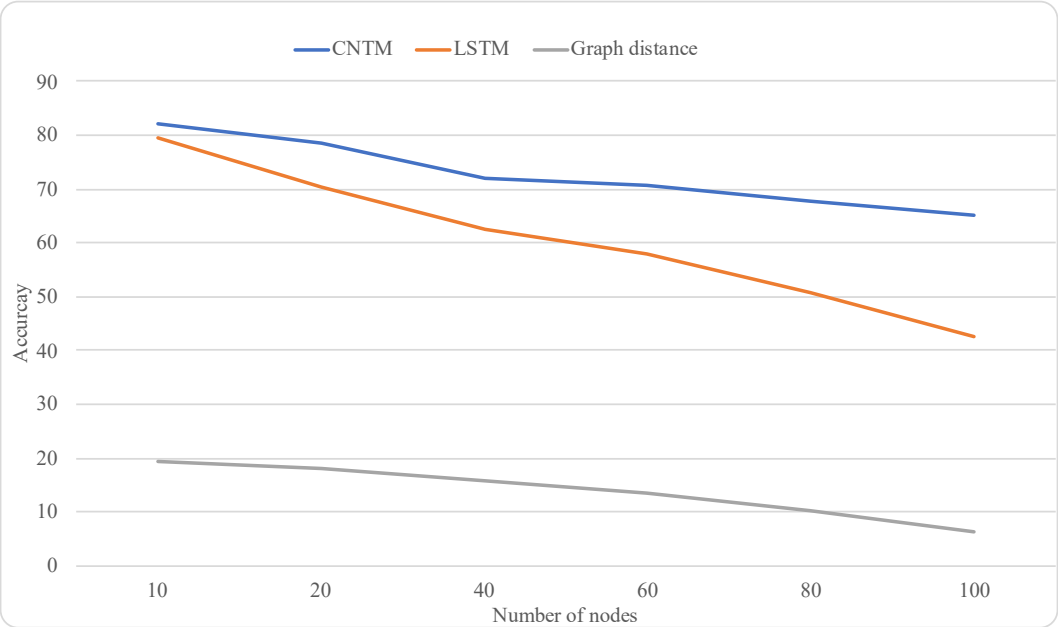


Figure 4.3: Comparison of different link predictors.

for the crisis graphs is 78,59% compared with 67.29% for the LSTM and 16.46% for the graph distance. It is in the same range of the accuracy obtained using a randomly generated graph of 20 nodes. This increase in accuracy might be because, in randomly generated graphs, we average the results over 100 different graphs. Some of the graphs might perform worst or better than the average depending on randomly generated edges. The crisis graph, on the other hand, is a well-defined graph presenting logical edges and connections.

Figure 4.4 show box-plots comparing the CNTM and the LSTM with the graph distance as a baseline. It illustrates that both these approaches perform, on average, 42% better than the graph distance. Finally, it is necessary to point here that the variance in the performance of the CNTM is much lower than the other approaches. Consequently, the CNTM algorithm is not only more accurate but also produces more stable results. [5]

[5]For further details on the results of the CNTM see **Paper C**.

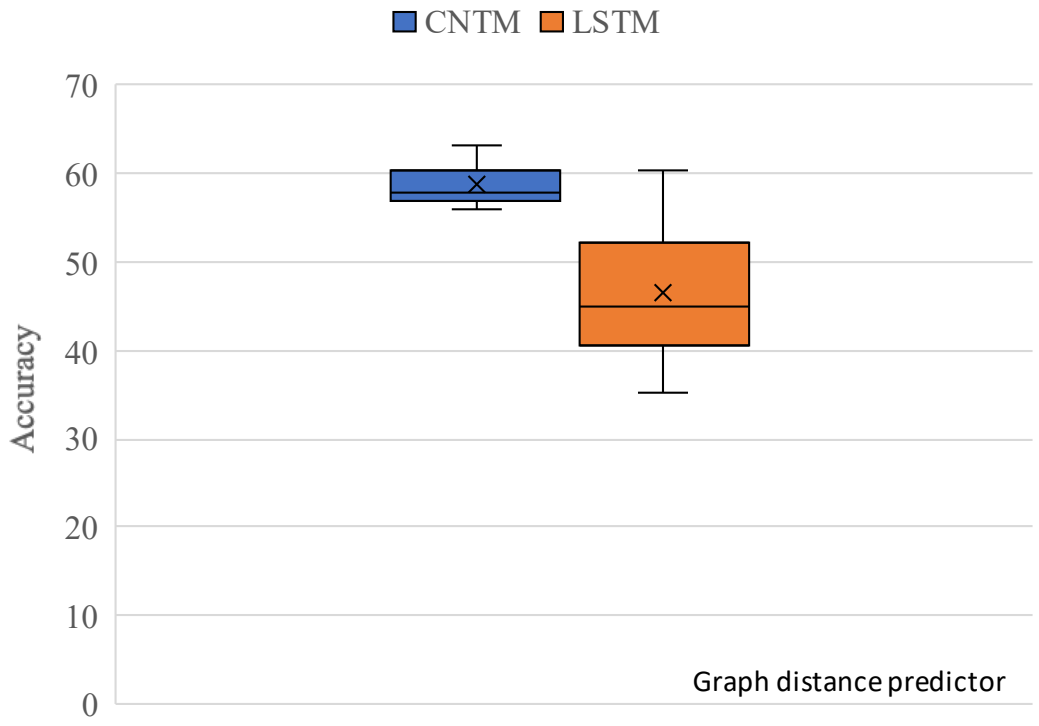


Figure 4.4: Comparison of different link predictor with the graph distance predictor as the baseline.

4.4 Evaluation of the framework

In addition to the single component evaluation, we wanted to verify the value of the information retrieval framework as a whole for EMS experts. When the context-aware autoencoder is combined with the social media text classifier and the link predictor, does it at all give any value to the EMS personnel?

To verify the usefulness of the whole information retrieval framework to experts in emergency management, we designed a survey as follows.[6] We simulated a run of the framework following the same scenario given to the experts, and we retrieved the tweets judged by the framework to be most relevant to the situation. Those tweets were given a grade of 4, and the rest of the tweets a grade of 1. Let us call this grade $E_{framework}$. A total of 20 tweets were provided to each expert containing tweets picked randomly from the data set along with the tweets extracted by the framework. The experts were presented with a crisis scenario, which included the status of the crisis and information gathered so far. They were then asked to evaluate the usefulness of the tweet from a scale of one to four. An evaluation of four represents useful information, and one represents a completely useless tweet.

[6]The survey is available at this link.

Let call this grade E_{expert}. Then, we study the correlation between what the expert evaluation judges as useful and with what the framework retrieves for the same situation by examining the distribution of the difference E between $E_{framework}$ and E_{expert} (Equation 4.1).

$$E = E_{framework} - E_{expert}. \tag{4.1}$$

Two Red Cross (RC) workers, a policeman, a Word Food Program (WFP) field worker, and seven academical experts answered the survey. The survey provided each responder with 20 randomly chosen tweets. Some of the tweets were categorized as relevant by our framework, while others were not. Whether a tweet had automatically been categorized as relevant or not was concealed to the participants. The categorization mechanisms were evaluated as successful when both the responders and the automatic algorithm picked the same tweets as relevant. Concretely, the experts were presented with a crisis scenario that contained the status of the crisis and information gathered so far and asked to evaluate the usefulness of the tweet from a scale of 1 to 4. The actual topics produced by the QG were also not visible to the experts. Only the results of the predictions (i.e., the tweets) were evaluated, which provided a "black box" verification of the predictions. [7]

The results show that the experts and the framework "agree" on the usefulness of 70.09% of the tweets. By "agree" we mean that the experts give a grade of 3 or 4 for a tweet retrieved by the framework, or a grade of 1 or 2 to the information disregarded by the framework in the same situation. Figure 4.5 shows a box-plot comparing the expert opinion with the information retrieved by the framework (E). It reveals that, on average, most of the expert opinion on the tweets varies within 1 point (in the survey scale) of the framework judgment. Figure 4.5 shows a box-plot comparing the EMSs opinion (RC workers, policeman, WFP field worker) with the opinions of the academics. The figure reveals that the distribution of E does not vary much between the two types of experts. However, E tends to have negative values, which means that there is a trend of the expert judging tweets as useful while the framework did not extract those tweets. These results are justified by what we discussed in Section 4.2, where we mentioned that our classifier suffers from low recall, indicating that it fails to identify a proportion of the tweets related to the topic.

[7] For further details on the results of the framework see **Paper D**.

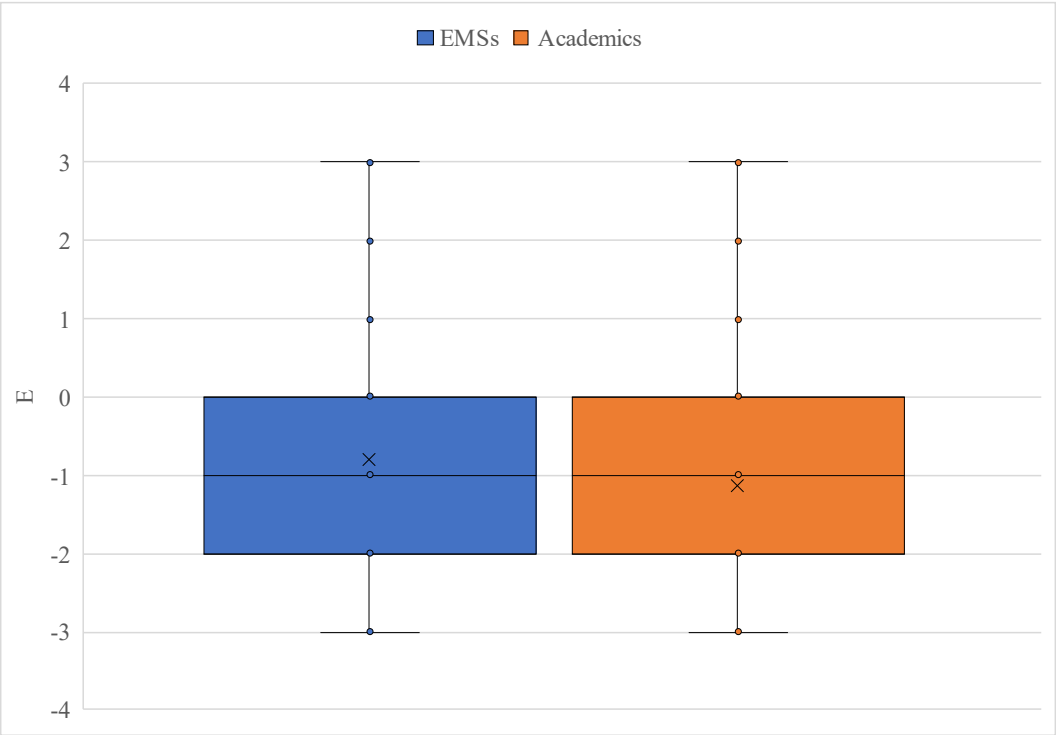


Figure 4.5: Comparison of the EMSs and academic experts opinions with the framework assessment

Chapter 5

Conclusions and Future Work

> *To live is to war with trolls in heart and soul. To write is to sit in judgement on oneself.*
> ***– Henrik Ibsen***

5.1 Summary of contributions

Despite the recent advances in applying machine learning to social media analysis in general and in crises in particular, there are still two significant gaps that have not been addressed. The first gap is categorizing social media texts with non-standard spelling issues. In this area, word embeddings have been the go-to approach to normalize and represent words in a machine-readable form. However, the traditional word embedding approaches do not integrate the non-standard spelling issue as part of their model. The second gap is the failure of most social media analysis platforms to provide emergency personnel with the information they need at the time they need it and considering the context of the ongoing crisis. Many EMSs deem social media analytics platforms as a tool that increases their already vast amount of information to an overload. The information overload is a significant factor behind their reluctance to use it. In this thesis, we tried to address these gaps by proposing the following contributions:

- We have proposed combining a denoising autoencoder with a context encoder to determine a mapping from the vocabulary-space and real vector space. This mapping allows us, in turn, to define a metric in word-space that encompasses non-standard spelling variations of words, such as misspellings, as well as words used in similar contexts. This work is a first attempt at defining a met-

ric deliberately, including non-standard spelling in word-space using neural networks. (**Paper B**)

- We presented a neural network that is able to model conditional graphs. A conditional graph has contexts that condition the transitions from one node another. This context is typically external and out of control for the learning mechanism. We showed that such graphs could divide into two parts: an environment and transition. We introduced the Conditional Neural Turing Machine (CNTM), which learns the transitions dependant on this context. The CNTM's role is to predict the next node in the graph, both given the previous set of nodes and input from the environment. We showed that, if trained on an incomplete graph, the CNTM becomes a link prediction algorithm capable of inferring relationships between nodes in a graph and complete the missing links in the training graph. (**Papers C**)

- We proposed an intelligent information retrieval framework from social media in crises. The developed combines two components. The first component classifies social media messages into distinct and separate topics representing a piece of information or a question asked by EMSs during a specific situation. It plays the roles of the environment in the conditional graph described earlier. The second component is a link prediction component that decides which information (topic) to retrieve based on the information available so far and the status of the crisis. Since crises situations typically result in highly complex scenarios, information overload and erroneous information is an important problem. The framework tackles this problem by only providing the information that the EMS personnel need in the context in which they need it. (**Paper D**)

5.2 Limitations and future work

The contributions and findings of this work can suffer from a few limitations. First, despite the improvements the string metric and embedding approach proposed in **Paper B** can provide to multiple NLP tasks, the resulting metric does not satisfy all the theoretical properties of a string metric [93]: the metric does not guarantee the identity of indiscernibles. In our context, this means that the distance between the embedding of two words should be zero if and only if they are the same word. In our approach, two different words can have the same embedding, and thus have a distance between them of zero. A range close to zero might be desirable for two

variant spellings of the same word. However, this is also possible for words used frequently in the same context but does not have the same meanings. Theoretically, there is a fine line between a metric that merges two closely related words of different meaning, and two misspelling of the same word. This distinction is not yet thoroughly studied.

We tried to make an effort to construct graphs encapsulating the complex information need flow during various crisis scenarios. However, the graphs created from the crisis scenario we considered are limited compared to the countless number of scenarios that can occur during a single crisis. Training our link prediction algorithm on such limited graphs should be seen as an early proof of concept. Even though the performance of the CNTM decreases as the graph becomes bigger, it has shown that it is by far better than all the other tested approaches, and can perform well on more massive random graphs (up to 80 nodes). This observation leads us to assume that, as we build bigger graphs, describing more complex crisis scenarios in the future, the loss of performance will be comparable to the decline observed in random graphs. A future direction in this topic is to increase the performance of the CNTM beyond 80 nodes graph.

Furthermore, we believe that the CNTM can be used to learn more general crisis trends, which would mean that it absorbs the information requirements from few emergency scenarios, and applies this to an entirely new emergency. Since the learning algorithm can learn from incomplete graphs, and this way becomes a link prediction algorithm, it implies that it can be applied to other link prediction problems in social networks, NLP, and product recommendation. This direction is yet to be investigated.

To evaluate both the framework and the TD, we chose an extreme weather case with data from Twitter. The reason behind this limitation is two fold. First, extreme weather is the most common emergency in Norway, which puts the experts we want to survey later in familiar territory. Second, the *CrisisLex* [90] and *CrisisNLP* [91] platforms provide an appropriately labeled data from Twitter. This labeling not only includes if the tweets are related to a crisis or not but also the type of information it provides, which facilitates further labeling into the topics needed by EMS. Unfortunately, even thought the QG support (indoor fire, extreme weather, and public disturbance), *CrisisLex* and *CrisisNLP* only contain data about extreme weather.

Eleven experts evaluated the framework, and they were asked to assess only 20 tweets each. A noticeable improvement in the validation results is to increase the number of experts, and the number of tweets evaluated. Concerning the low

number of tweets, we tried to choose that number in a way that balances between two constraints: The first is the minimum number of tweets needed to have an acceptable correlation analysis. The second is not to take more than 20 minutes of the participants' time so that they are not discouraged by a long and tedious survey. Concerning the number of participants, we tried to propose a survey with all the expertise in our contact group. The study is voluntary, and eleven experts were able to answer. Despite the relatively low number, we thought it is essential to show their opinions about the information extracted by the framework as it was designed primarily for them.

REFERENCES

[1] S. Re, "Natural catastrophes and man-made disasters in 2017: A year of record-breaking losses," *Sigma Report*, no. 1, 2018.

[2] D. Alexander, "Disaster and emergency planning for preparedness, response, and recovery." Oxford University Press, 2015.

[3] S. Vieweg, C. Castillo, and M. Imran, "Integrating social media communications into the rapid assessment of sudden onset disasters," in *International Conference on Social Informatics*. Springer, 2014, pp. 444–461.

[4] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: what twitter may contribute to situational awareness," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2010, pp. 1079–1088.

[5] J. Radianti, S. R. Hiltz, and L. Labaka, "An overview of public concerns during the recovery period after a major earthquake: Nepal twitter analysis," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 136–145.

[6] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[7] G. Valkanas, I. Katakis, D. Gunopulos, and A. Stefanidis, "Mining twitter data with resource constraints," in *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 01*. IEEE Computer Society, 2014, pp. 157–164.

[8] H. Zhang, R. Sproat, A. H. Ng, F. Stahlberg, X. Peng, K. Gorman, and B. Roark, "Neural models of text normalization for speech applications," *Computational Linguistics*, vol. 45, no. 2, pp. 293–337, 2019.

[9] A. Olteanu, S. Vieweg, and C. Castillo, "What to expect when the unexpected happens: Social media communications across crises," in *Proceedings of the 18th ACM conference on computer supported cooperative work & social computing*. ACM, 2015, pp. 994–1009.

[10] T. Friberg, S. Prödel, and R. Koch, "Information quality criteria and their importance for experts in crisis situations," in *Proceedings of the 8th International ISCRAM Conference*, 2011, pp. 145–149.

[11] A. Aw, M. Zhang, J. Xiao, and J. Su, "A phrase-based statistical model for sms text normalization," in *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, 2006, pp. 33–40.

[12] E. Cambria and B. White, "Jumping nlp curves: A review of natural language processing research," *IEEE Computational intelligence magazine*, vol. 9, no. 2, pp. 48–57, 2014.

[13] Y. Li, L. Xu, F. Tian, L. Jiang, X. Zhong, and E. Chen, "Word embedding revisited: A new representation learning and explicit matrix factorization perspective," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[14] R. Lebret and R. Collobert, "Word emdeddings through hellinger pca," *arXiv preprint arXiv:1312.5542*, 2013.

[15] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," *Journal of Machine Learning Research*, vol. 8, no. Oct, pp. 2265–2295, 2007.

[16] M. A. Qureshi and D. Greene, "Eve: explainable vector based embedding technique using wikipedia," *Journal of Intelligent Information Systems*, pp. 1–29, 2018.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[18] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, p. 67, 2015.

[19] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. X. Zhou, "Leadline: Interactive visual analysis of text data through event identification and exploration," in *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2012, pp. 93–102.

[20] D. Pohl, A. Bouchachia, and H. Hellwagner, "Automatic identification of crisis-related sub-events using clustering," in *2012 11th International Conference on Machine Learning and Applications*, vol. 2. IEEE, 2012, pp. 333–338.

[21] A. Ritter, O. Etzioni, S. Clark *et al.*, "Open domain event extraction from twitter," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 1104–1112.

[22] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 155–164.

[23] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 1155–1158.

[24] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: aggregating and visualizing microblogs for event exploration," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2011, pp. 227–236.

[25] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*. Association for Computational Linguistics, 2010, pp. 181–189.

[26] R. Power, B. Robinson, J. Colton, and M. Cameron, "Emergency situation awareness: Twitter case studies," in *International conference on information systems for crisis response and management in mediterranean countries*. Springer, 2014, pp. 218–231.

[27] J. Weng and B.-S. Lee, "Event detection in twitter," in *Fifth international AAAI conference on weblogs and social media*, 2011.

[28] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," in *Fifth international AAAI conference on weblogs and social media*, 2011.

[29] D. Freitag, "Machine learning for information extraction in informal domains," *Machine learning*, vol. 39, no. 2-3, pp. 169–202, 2000.

[30] M. Imran, S. Elbassuoni, C. Castillo, F. Diaz, and P. Meier, "Practical extraction of disaster-relevant information from social media," in *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 2013, pp. 1021–1024.

[31] ——, "Extracting information nuggets from disaster-related messages in social media." in *Iscram*, 2013.

[32] M. K. Lindell, R. W. Perry, C. Prater, and W. C. Nicholson, *Fundamentals of emergency management*. FEMA Washington, DC, USA:, 2006.

[33] V. S. M. W. Group *et al.*, "Lessons learned: Social media and hurricane sandy," *Homeland security science and technology*, pp. 1–39, 2013.

[34] ——, "Next steps: Social media for emergency response," *Homeland security science and technology*, 2012.

[35] F. Aligne and J. Mattioli, "The role of context for crisis management cycle," in *Supporting Real Time Decision-Making*. Springer, 2011, pp. 113–132.

[36] R. M. Tripathy, A. Bagchi, and S. Mehta, "A study of rumor control strategies on social networks," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1817–1820.

[37] A. Sarcevic, L. Palen, J. White, K. Starbird, M. Bagdouri, and K. Anderson, "Beacons of hope in decentralized coordination: Learning from on-the-ground medical twitterers during the 2010 haiti earthquake," in *Proceedings of the ACM 2012 conference on computer supported cooperative work*. ACM, 2012, pp. 47–56.

[38] K. Starbird, L. Palen, A. L. Hughes, and S. Vieweg, "Chatter on the red: what hazards threat reveals about the social life of microblogged information," in *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. ACM, 2010, pp. 241–250.

[39] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: Can we trust what we rt?" in *Proceedings of the first workshop on social media analytics*. ACM, 2010, pp. 71–79.

[40] L. Palen, S. Vieweg, J. Sutton, S. B. Liu, and A. Hughes, "Crisis informatics: Studying crisis in a networked world," in *Third International Conference on e-Social Science*, vol. 7, no. 10. Citeseer, 2007.

[41] S. E. Vieweg, "Situational awareness in mass emergency: A behavioral and linguistic analysis of microblogged communications," Ph.D. dissertation, University of Colorado at Boulder, 2012.

[42] C. Kobus, F. Yvon, and G. Damnati, "Normalizing sms: are two metaphors better than one?" in *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2008, pp. 441–448.

[43] D. Pennell and Y. Liu, "Toward text message normalization: Modeling abbreviation generation," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5364–5367.

[44] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[45] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[46] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[47] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 641–648.

[48] G. E. Dahl, R. P. Adams, and H. Larochelle, "Training restricted boltzmann machines on word observations," *arXiv preprint arXiv:1202.5695*, 2012.

[49] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in neural information processing systems*, 2009, pp. 1607–1614.

[50] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–43.

[51] R. Irfan, C. K. King, D. Grages, S. Ewen, S. U. Khan, S. A. Madani, J. Kolodziej, L. Wang, D. Chen, A. Rayes *et al.*, "A survey on text mining in social networks," *The Knowledge Engineering Review*, vol. 30, no. 2, pp. 157–170, 2015.

[52] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[53] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual web texts," *Information retrieval*, vol. 12, no. 5, pp. 526–558, 2009.

[54] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "Tweetcred: Real-time credibility assessment of content on twitter," in *International Conference on Social Informatics*. Springer, 2014, pp. 228–243.

[55] D. T. Nguyen, K. A. Al Mannai, S. Joty, H. Sajjad, M. Imran, and P. Mitra, "Robust classification of crisis-related data on social networks using convolutional neural networks," in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[56] S. Rosenthal, N. Farra, and P. Nakov, "Semeval-2017 task 4: Sentiment analysis in twitter," in *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, 2017, pp. 502–518.

[57] J. Rogstadius, M. Vukovic, C. Teixeira, V. Kostakos, E. Karapanos, and J. A. Laredo, "Crisistracker: Crowdsourced social media curation for disaster awareness," *IBM Journal of Research and Development*, vol. 57, no. 5, 2013.

[58] K. Kireyev, L. Palen, and K. Anderson, "Applications of topics models to analysis of disaster-related twitter data," in *NIPS Workshop on Applications for Topic Models: Text and Beyond*, vol. 1. Canada: Whistler, 2009.

[59] R. Mazloom, H. Li, D. Caragea, M. Imran, and C. Caragea, "Classification of twitter disaster data using a hybrid feature-instance adaptation approach," *Proc. of ISCRAM*, 2018.

[60] R. Mazloom, H. Li, D. Caragea, C. Caragea, and M. Imran, "A hybrid domain adaptation approach for identifying crisis-relevant tweets," *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, vol. 11, no. 2, pp. 1–19, 2019.

[61] F. Alam, S. Joty, and M. Imran, "Graph based semi-supervised learning with convolution neural networks to classify crisis related tweets," in *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[62] K. Starbird and J. Stamberger, "Tweak the tweet: Leveraging microblogging proliferation with a prescriptive syntax to support citizen reporting," 2010.

[63] O. Okolloh, "Ushahidi, or testimony: Web 2.0 tools for crowdsourcing crisis information," *Participatory learning and action*, vol. 59, no. 1, pp. 65–70, 2009.

[64] A. M. MacEachren, A. Jaiswal, A. C. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford, "Senseplace2: Geotwitter analytics support for situational awareness," in *2011 IEEE conference on visual analytics science and technology (VAST)*. IEEE, 2011, pp. 181–190.

[65] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "Aidr: Artificial intelligence for disaster response," in *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, 2014.

[66] Z. Ashktorab, C. Brown, M. Nandi, and A. Culotta, "Tweedr: Mining twitter to inform disaster response," *Proc. of ISCRAM*, 2014.

[67] C. Caragea, N. McNeese, A. Jaiswal, G. Traylor, H.-W. Kim, P. Mitra, D. Wu, A. H. Tapia, L. Giles, B. J. Jansen *et al.*, "Classifying text messages for the haiti earthquake," in *Proceedings of the 8th international conference on information systems for crisis response and management (ISCRAM2011)*. Citeseer, 2011.

[68] A. S. Jadhav, H. Purohit, P. Kapanipathi, P. Anantharam, A. H. Ranabahu, V. Nguyen, P. N. Mendes, A. G. Smith, M. Cooney, and A. P. Sheth, "Twitris 2.0: Semantically empowered system for understanding perceptions from social data," 2010.

[69] F. Abel, C. Hauff, G.-J. Houben, R. Stronkman, and K. Tao, "Semantics+ filtering+ search= twitcident. exploring information in social web streams," in *Proceedings of the 23rd ACM conference on Hypertext and social media*. ACM, 2012, pp. 285–294.

[70] L. Palen and K. M. Anderson, "Crisis informaticsnew data for extraordinary times," *Science*, vol. 353, no. 6296, pp. 224–225, 2016.

[71] K. Rudra, P. Goyal, N. Ganguly, P. Mitra, and M. Imran, "Identifying subevents and summarizing disaster-related information from microblogs," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 265–274.

[72] K. Rudra, P. Goyal, N. Ganguly, M. Imran, and P. Mitra, "Summarizing situational tweets in crisis scenarios: An extractive-abstractive approach," *IEEE*

Transactions on Computational Social Systems, vol. 6, no. 5, pp. 981–993, 2019.

[73] V. Nunavath, A. Prinz, and T. Comes, "Identifying first responders information needs: supporting search and rescue operations for fire emergency response," *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, vol. 8, no. 1, pp. 25–46, 2016.

[74] M. Al Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social network data analytics*. Springer, 2011, pp. 243–275.

[75] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[76] M. E. Newman, "Clustering and preferential attachment in growing networks," *Physical review E*, vol. 64, no. 2, p. 025102, 2001.

[77] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 547–579, 1901.

[78] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[79] W. Liu and L. Lü, "Link prediction based on local random walk," *EPL (Europhysics Letters)*, vol. 89, no. 5, p. 58007, 2010.

[80] A. Clauset, C. Moore, and M. E. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, no. 7191, p. 98, 2008.

[81] H. C. White, S. A. Boorman, and R. L. Breiger, "Social structure from multiple networks. i. blockmodels of roles and positions," *American journal of sociology*, vol. 81, no. 4, pp. 730–780, 1976.

[82] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI*, vol. 99, 1999, pp. 1300–1309.

[83] D. Heckerman, C. Meek, and D. Koller, "Probabilistic entity-relationship models, prms, and plate models," *Introduction to statistical relational learning*, pp. 201–238, 2007.

[84] K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu, "Stochastic relational models for discriminative link prediction," in *Advances in neural information processing systems*, 2007, pp. 1553–1560.

[85] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[86] A. M. Turing, "On computable numbers, with an application to the entscheidungsproblem," *Proceedings of the London mathematical society*, vol. 2, no. 1, pp. 230–265, 1937.

[87] J. Baldridge, "The opennlp project," *URL: http://opennlp. apache. org/index. html,(accessed 2 February 2012)*, p. 1, 2005.

[88] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[89] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[90] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, "Crisislex: A lexicon for collecting and filtering microblogged communications in crises." in *ICWSM*, 2014.

[91] M. Imran, P. Mitra, and C. Castillo, "Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages," *arXiv preprint arXiv:1605.05894*, 2016.

[92] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 2009, pp. 322–335.

[93] M. Hazewinkel, *Encyclopaedia of Mathematics*. Springer Science & Business Media, 2013.

PART II

A

Paper A

Title:	Social Media Analysis in Crisis Situations: Can Social Media be a Reliable Information Source for Emergency Management Services?
Authors:	Mehdi Ben Lazreg[†], Narayan Ranjan Chakraborthy[‡], Stefan Stieglitz[*], Tobias Potthoff[*], Björn Ross[*], and Tim Majchrzakand[‡]
Affiliation:	[†] Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway [‡] Dept. of Information Systems, University of Agder (UiA), N-4604 Kristiansand, Norway [*] Dept. of Professional Communication in Electronic Media and Social Media, University of Duisburg-Essen, N-47057 Duisburg, Germany
Conference:	*The* 27th *International Conference on Information Systems Development*, Lund, Sweden, August 2018
Copyright ©:	AIS

Social Media Analysis in Crisis Situations: Can Social Media be a Reliable Information Source for Emergency Management Services?

Mehdi Ben Lazreg, Narayan Ranjan Chakraborthy, Stefan Stieglitz, Tobias Potthoff, Björn Ross, and Tim Majchrzakand

Abstract — **Learning and understanding what happened before, during, and after a crisis is extremely important for the improvement of the response process. For this purpose, social media has become an important communication medium used by both the affected persons and the emergency management services (EMSs). However, in different crises, different information may be needed, and the information shared in social media varies in its usefulness: It could be highly critical or completely irrelevant to the rescue operation. Supplying the best possible up-to-date information is crucial to the EMS, whose actions based on that information may save lives and resources. This paper studies a particular use case of extreme weather in Norway and identifies the information needs, the problem faced by EMSs, and how they use social media. It, further, pinpoints what different social media analysis platforms can provide in this type of crisis. The results of the research are criteria that social media analysis should follow to address EMSs' concerns. The output of this work can be used to more precisely describe social media communication for crises and to design more efficient platforms for information retrieval from social media.**

Keywords— **Social media, Crisis management, Emergency management services.**

I. Introduction

Social media has become the de facto medium for public crisis communication [38]. It plays a pivotal role in most crises today, from obtaining life signs from people affected to communicating with EMSs [11]. Three types of information sharing on social media during a crisis can be distinguished. First, from EMS to the public: EMSs relay situation updates, evacuation orders, possible dangers... to the public. Second, from the public to public: the public uses social media to maintain contact with relatives, friends and loved ones, and to show support for the community. Finally, from the public to the EMS: The public uses social media

to report problems, needs, calls for help, and provide information throughout the crisis. The first two information sharing types are now well established with training given to EMSs on how to effectively communicate information to the public [8] [7]. Many social media platforms are also providing ways for people in the affected area to report their safety status. The third type, on the other hand, is still facing many challenges. Although the information shared by the public can enhance the situational awareness of the EMS, many EMSs are still skeptical about using that information.

A priori, the EMSs' skepticism might be due to the nature of the massages posted on social media during a crisis: These messages tend to vary in their usefulness highly: The message can be off-topic, personal, or informative. Much of the retrospective research on Twitter messages posted during events show that those messages can identify the problems appearing during the relief effort. Finding this useful information can accelerate disaster response. However, the task is not easy due to, among other things, information overload. In an analysis performed on tweets related to the 2015 Nepal earthquake, discovered that even though relevant topics are discussed, the information present in the discussions is often irrelevant [24]. As an example, monetary support is one of the most discussed topics, but the majority of the messages are appealing for donations from ordinary people outside the affected areas and not actual financial needs which are more relevant to the EMSs.

In addition, the way rumors and misinformation are spread on social media makes it an untrustworthy source of information [31]. The social media analysis platforms can also be to blame for the EMSs' skepticism: Most of the social media analysis platforms during a crisis follow a data-driven approach that analyzes the data first by finding ways to extract as much information related to the crisis as possible [31] [22]. Such an approach may result in information obtained during the analysis that is not useful to the EMS, as opposed to an approach that inspects the data in ways that make it pertinent to the meaningful questions for the EMS.

Many of previously listed reason are founded. However, do they make social media irrelevant as information sources for EMSs? In this paper, we contribute to bridging the gap that exists in information sharing from the public the EMS in social media. We will focus on an extreme weather use case in Norway. Around this use case, we conducted interviews with governmental EMSs that include the police, firefighters, municipality, and red cross. The aim of the interviews was to learn about the standard operating procedure in an extreme weather crisis, how the EMSs share their information and establish situational awareness, and what

role social media plays in the emergency response process. Further, we present an overview of the information social media can provide and the state-of-the-art social media analysis platforms. By studying the data availability and needs from both the EMSs' side and the social media side, we try to identify the common denominator between the two. This paper contributes to addressing the question of how social media analysis platforms should be designed in order to effectively support EMSs.

This paper is organized as follows. Section II presents our interview methodology and outcome. Section III illustrates the different types of information social media can provide during a crisis. Section IV gives an overview of the state-of-the-art social media analysis platforms in crisis situations. Section V discusses the lessons learned from our study and present the primary outcomes of the paper. Finally, Section VI concludes and provides pointers to future work.

II. Crisis response procedures and information needed

Extreme weather is the most damaging and frequent type of crisis in Norway (Norway experiences on average 3 per year). It is characterized by an unusual and unexpected rainfall, snowfall, heat or cold waves. When it occurs, extreme weather may damage the infrastructure including roads, leaving towns cut off from the rest of the country, the electricity network (for example, during the storm Hilda in 2013, 35000 home were out of electricity), and, most importantly, it may lead to human injuries and death. The EMSs that are highly involved in this type of crisis are the police, firefighters, municipalities and the Norwegian red cross. To find out which information these EMSs need, we conducted a two-hour semistructured personal interview with representatives from four prominent local authorities on May 23, 2017: the chief of staff at Agder police district, the crisis preparedness leader at Kristiansand municipality, head of a unit at the Grimstad fire brigade, and a volunteer in Grimstad Red Cross. By using the extreme weather use case, we managed to put the interviewees in a familiar situation in which they have a lot of experience to get the most out of the interviews. The interview was oriented to discover the current crisis response procedure, the information needed during this process, the practice in information gathering, and the gaps of such practice. We brought up social media during the interview to discover how the EMSs currently use it their opinions about its potential.

II.1 Methodology

Since we are concerned with the public-to-EMS information sharing through social media, the interview aims first to understand the current crisis response and information gathering procedure and identify the problems facing it. Further, we

try to learn the extent to which social media is currently used and what they think about its potential use in crisis situations. Therefore, the questions we asked were oriented to achieve those previously cited goals. The questions were:

- How do you proceed during a crisis?
- Which kind of information do you try to collect?
- How do you obtain this information?
- What are the biggest problems you face in this information collection process?
- Do you use social media as an information source?
- Would you be interested in analyzing social media?

For each of the above question, we asked a series of follow-up questions depending on the participant's response.

The analysis of the interviews was done qualitatively following the Mayring approach [15]. When analyzing the interview data, we looked at the opinion held by the EMSs on two critical question for this paper. The first question is: How do they assess the current information gathering process during a crisis? For this question, we distinguish three categories of opinions: high, middle, or low, confidence in the process. Then, we classified the answer of each participant to one of the previously listed categories. The second question is: What do they think about the use of social media in crisis situations? Here, we separate between enthusiastic, halfhearted and skeptical views. We further classified the interview data into one of these categories based on the opinion each participant holds.

Outcome of the interviews

In Norway, during a crisis, police forces are in charge of the situation and responsible for the response. The police operation center should have the overall picture of the crisis and act accordingly. In the case of extreme weather, the most important information EMSs need to know are:

- Which are the affected areas and people?
- What are the areas in danger that need evacuation?
- How fast is the water rising?
- Who has already been evacuated and who still needs to be evacuated?

- Do the evacuated people have what they need?
- Do individuals in a threatening situation know they need to evacuate?

DESCRIPTION OF THE CURRENT INFORMATION GATHERING PROCESS

The information needed during an extreme weather crisis is diverse and sometimes very hard to get promptly. To get that information, EMSs receive phone calls updated from all the emergency call center, regional authorities, and other EMSs for updates about the situation: "In current practice, we get our information through reports from emergency call centers. Besides, we gather information by visiting some news, weather and media websites. But apart from that, we do not collect information from other sources. The rest is information from the staff we send to the event location," the police representative said. The police officer agrees that there are other techniques to gather information they can use to improve their assessment process.

"During search and rescue operations, the more useful information we have, the better the operations. It is very crucial to gather as much information as possible as quickly as possible," asserted the Red Cross volunteer. The amount of information available early in the process that helps direct the crews to the right spot, such as, for example, which places have been searched for the last 2 hours, is crucial to the rescue operation. As it is now, the red cross representative thinks that gathering the necessary information is a process that takes time: there is a lot of information available in sources such as social media and news reports that can be more efficiently collected using simple internet search that is not collected because the EMSs lack the technological tools.

The lack of information during a crisis can lead to unexpected incidents for the EMSs. As an example, the Kristiansand municipality representative shared with us his insight on a flood that happened in Kvinsdal, Norway in 2014. The region experienced 130mm of rainfall in the mountains over one night. To put this in perspective, 44.2mm is a normal rainfall in Norway. The water took 12 to 18 hours to reach the town. When it did, the city bridge was damaged, boathouses destroyed, and the cultural center submerged. The damages could have been avoided with more coordination between the municipality, the firefighters and the police. Due to this lack of coordination, no one had an overall picture of the situation and could assess the extent of the crisis. This lack of a full picture and understanding of the situation caused the authorities in place to allocate fewer resources than necessary. The civil protection, for example, understood the extent of the water flood only when a civilian called to get help pumping water out of his basement. The firefighters

went from house to house trying to help the inhabitants, but they had no idea about the overall extent of the flood. "Having the right picture of the situation allows the persons sitting at the top to make the correct decision based on the right reasons," the municipality representative asserted. He thinks that the main reason behind the shortcoming in getting the overall picture is the lack of coordination and proactive information sharing between different EMSs.

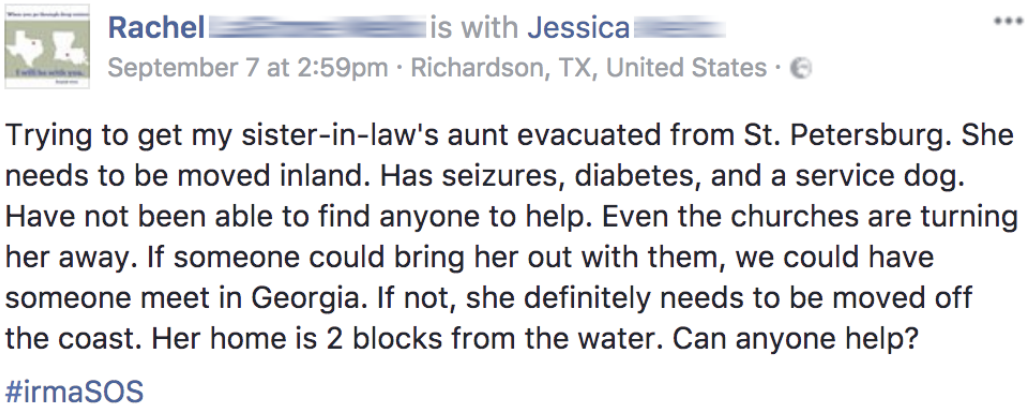
POTENTIAL USE OF SOCIAL MEDIA IN CRISIS SITUATIONS

When we asked the interviewees about their experience with social media so far, they all agreed that they only use social media to communicate updates to the public and monitor social activities of other EMSs. However, when asked if it can solve some of the information needs problems, their sentiments were mixed: The police representative was enthusiastic about the idea. For him, it is important to include technology in crisis management work, especially information from different data sources including social media. This information should enter the control room, and the decision-makers should have access to those data. The Red Cross representative was less enthusiastic, stating: "If we can use social media or other information sources to identify where our presence is needed most instead of sending persons on the ground, that would help save us a lot of resources. But I can not see how." This quote illustrates one of the gaps that exist between EMSs and social media analysis: Despite the abundance of tool we describe in Section IV, the EMS officers we interviewed are either unaware of their existence or do no see how these tool can be useful for them. Finally, the municipality representative was more skeptical of the idea of using social media for information gathering. He did not see how social media can help get a better overall picture of the crisis. Despite the downfall that the current procedure might experience, he does not think that social media is the answer. He stated that "we only use information from different EMSs to get an overall picture. We do not rely on 'Mr. Somebody' for the information." This statement also reflects the trust issues toward the information shared on social media.

III. INFORMATION IN SOCIAL MEDIA

Data produced and shared in social media like Facebook, and Twitter has proven to be valuable in many different contexts.The previous section reviewed the current information sharing procedure for the EMSs we interviewed and the problems and gaps they present especially in getting a comprehensive picture of the situation. In this section, we examine the different kinds of information that are available and accessible on public social media platforms in specific cases and argue that this information can help solve some of these problems.

Figure A.1: Facebook post



III.1 Textual information

Though social media nowadays seem to give more weight to other types of information, purely textual information is still at the heart of most platforms. Textual information is externalized, explicit or codified and accessible with automated techniques [32]. Depending on the language, textual information follows specific grammar rules which allow one to access even free and unstructured text. Dictionary-based analyses allow the selection of relevant texts according to a defined set of keywords or to assess sentiments in social media regarding a particular topic.

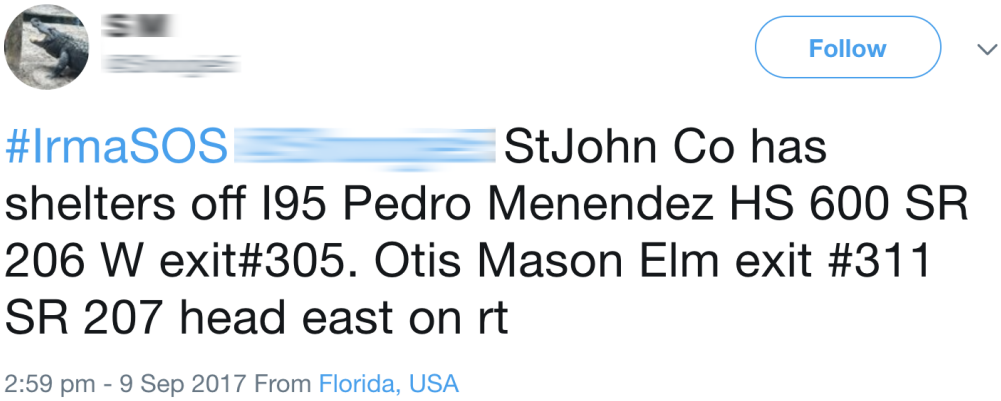
During the recent hurricanes Harvey and Irma in the United States and the Caribbean, a lot of textual information was shared in social media. For example, the following Figure A.1 depicts information from the social networking site (SNS) Facebook about a person that needs to be evacuated for special treatment due to her diabetes.

In Twitter, people also shared information on where to find shelters (Figure A.2). This behavior has been observed in other crises, too [35]. This information can be useful for emergency authorities as well, as they might not be aware of all shelters.

Often, the textual information is used to cluster the high number of posts into relevant and irrelevant posts or further categories [37]. Textual information is often used in aggregation [16]. Still, the individual text may contain information that can be relevant for the response management, e.g., about the inundation height in a particular area. The challenge for textual information is to define and improve filters and other relevance criteria that limit the number of posts to a manageable number.

III.2 Photos and videos

Figure A.2: Twitter post



Most social media platforms are capable of sharing not only text but also pictures and videos. Although the complexity of processing information in pictures or videos is much higher than for texts, they can also inform emergency service agencies better. For example, Fohringer et al. [5] use photos taken by eyewitnesses to derive quantitative data about water depth. Photos and videos depict the real situation on-site. In contrast to textual information, multimedia data is more objective and does not require laymen to interpret what s/he observes locally.

Figure A.3 shows a photo taken from a small flood in Oslo, Norway, and published on Twitter in August 2016. The picture illustrates to the emergency service agencies: The depth of water at the particular place, a car stuck in the flood, the road in need to be closed, and that people do not seem to be injured.

(Live) videos shared on social media platforms usually contain even more information than photos [20]. Videos can better convey, e.g., weather conditions or crisis dynamics. In the case of floods, videos could be used to measure the flow rate. New services like Periscope, which offers an easy-to-use live broadcast, are expected to become "game changers" (p. 8) in the disaster response management because intermediaries can be skipped and hence, transfer time reduced [4].

III.3 Spatial information

Most social media platforms provide location data for shared information, especially if mobile devices with built-in location sensors (e.g., GPS) are used. Spatial information can be used to filter for posts sent from a particular area [37] or to visualize shared information, e.g., with a map [6]. Often, geospatial information is not enabled by default, making the precise location of information impossible. In such cases, textual information mentioning the city or street have to serve as a proxy.

III.4 Response information

Figure A.3: Photo depicting a flood in Oslo from Twitter

A

Figure A.4: Added details in Twitter reaction

Comments, answers or commented retweets can contain information that complements the original information. In FigureA.4, the person first added @mentions to attract the attention of authorities. Later, she wrote the full address.

Eventually, the reply feature was also used to report the successful rescue. Hence, response information is an important information source as it contains information about the actuality or allows a better assessment of the severity of the reported issue (see also next section).

III.5 Quantitative information

Although some of the information mentioned above is also quantitative, we specifically want to point to other numbers available. Reactions to original posts, e.g., expressed as a Like on Facebook, or a Retweet on Twitter, can contain information about the urgency of the textual or other information. Hence, quantitative information could help to prioritize open issues reported through social media [11].

The pure number of comments or answers further indicates relevance. Lastly, before a crisis happens, emergency service agencies can use available quantitative data to assess and improve their potential reach that is also based on followers and like numbers of official accounts.

IV. State of the art in social media use in crisis situations

Social media can provide near-real-time information for the emergency responders to make effective decisions throughout various stages of the disaster management process [10] [33]. It also facilitates connectedness during the emergency and provides relevant and timely information from both official and non-official sources [29]. This information act as an artifact in the online environment [36]. After examining the different kind of information available in social media in some specific cases, this section take a more global view on the platforms available for social media analysis and how it is used around the world.

IV.1 Use of social media in crisis situations

Social media was used as a participatory media during Hurricane Katrina (2005); one of the first natural disasters where social media use was noted [23]. People used online platforms to help affected people by donating clothes, toys, etc., and social media helped emergency responders to coordinate [30]. Palen and Liu [23] noted that social media also played a significant role in finding the missing people in Hurricane Katrina. These cases marked the start of social media uses in crisis situations by EMSs; after that, social media data, or more precisely, Twitter data, has been heavily utilized in the management of almost every kind of disaster [3], for example the 2009 Red River flood in the US [34], the 2010 Haiti earthquake [27], the 2011 Japan tsunami [1], the 2015 Nepal earthquake [24] etc.

During the Haiti earthquake in 2010, digital volunteers used Twitter data to map the affected areas using Ushahidi. This crisis map became an invaluable resource for relief workers in the field [27]. During the 2015 Nepal earthquake, the Nepal police used social media as one of the main communication channels [28]. Kathmandu Living Lab prepared a Nepal quake map based on social media data [13]. The open nature of social media data helps the responders to operate from the ground. Social media are now gaining attention among those dealing with extreme weather disasters [19].

IV.2 Social media analysis platforms

It is always challenging to analyze social media data because of the diverse sources and unstructured nature of the data. Facebook, Twitter, and other social media platforms do not provide similar kinds of data, thus the need for social media analytics tools to analyze this data. There are few platforms available used by the

volunteers to analyze this unstructured data of social media during any crisis. The following list mentions a few standard tools.

- Artificial Intelligence for Disaster Response (AIDR)
- Tweak-the-Tweet
- Ushahidi
- TweetTracker
- TweetCred

The mentioned tools work for Twitter data sources or a combination of other social media platforms. Artificial Intelligence for Disaster Response (AIDR) uses machine learning to identify crisis-specific Twitter data automatically. By using a small of labeled tweets, AIDR allows one to detect the categories of tweets [12]. Through the identification of the category of tweets, it helps the responder to react to particular issues rather than the crisis as a whole.

Researchers at the University of Colorado developed a crowdsourcing platform called Tweak-the-Tweet. It focuses on specific hashtags to make the data structured [14]. After that, a parsing algorithm can be used to extract the information. One of the benefits of this system is that it can work with the existing social media infrastructures [27].

Ushahidi is a crowdsourcing platform first developed to map the reports of Kenyan post-election violence in 2008. Later it was widely used during the Haiti earthquake and Hurricane Sandy. This system is not only used to collect Twitter data. It can also be used to gather data from RSS feeds, email and SMS [26]. This variation makes this system more attractive.

To collect data from Facebook, Twitter, YouTube, etc. with a combination of keywords, location, and user information, TweetTracker was developed. It can easily map the geotagged post. There is a particular module in TweetTracker to facilitate disaster relief [18].

It is essential to get credible information during a crisis to assess the situation accurately. Due to the dynamic nature of social media, fake news can spread quickly and create a mess on the ground [21]. TweetCred was developed to find credible information shared by Twitter users in real time. It provides a credibility rating of a post, and a supervised automated ranking algorithm determined the credibility of that post [9].

V. Discussion and lessons learned

Social media is a platform with a significant potential to be used for public-to-EMS information sharing. We showed in the previous section that the information available in social media during a crisis can address EMSs' needs and help them establish a better situational awareness, and complements the current information gathering procedures to get a complete picture of the crisis. However, social media is still not considered a source of information for governmental EMSs in Norway as well as in many other countries around the world. It is understandable that EMSs question the value of social media for crisis response because of the gap still present in social media analysis and research. In this section, we will pinpoint these gaps based on the outcome of previous sections.

One of the main reasons behind the skepticism behind using social media as an information source by EMSs is the lack of confidence in the information present in the platform: As mentioned by our interviewees, they do not trust Mr. Somebody to deliver accurate situation update and needs. This concern is founded on the amount of rumor and disinformation spread in social media during a crisis [31], which leads us to the first question a social media analysis platform needs to answer: can it establish a trusted network of people to get the information from? A lot of research has been carried on developing an automated trust model for social media networks based on user behavior and interaction with other [25] [17]. However, none of these methods are integrated into the crisis-related social media platforms described in section IV. Social media content varies in quality from excellent to spam and abuse. Once a trusted network is established, we need to ask how we can ensure that only high-quality messages shared by the network are treated by the platform? By quality of the message, we mean one that is clear, readable, and concise. Information quality assurance models to identify high-quality information are another evolving topic of research in social media [2]. Nevertheless, their integration in social media analysis platform for crisis response is still unsatisfactory. Many data scientists are relatively new to the field of social media in crisis research. They are knowledgeable about the management and analysis of large-volume data but lack the understanding of the EMSs' needs. Data scientists tend to think that large volumes of social media data alone will reveal patterns of behavior during a crisis. Moreover, the growth of artificial intelligence and machine learning during the last few years has led to the emergence of many artificial intelligence-based analysis tools [12][37][11]. These tools analyze the data first by extracting as much information about as many topics related to the crisis as possible. When the focus is on the data, and its volume, rigor in data collection becomes an afterthought. In contrast, social media data must be analyzed in ways that provide relevant answers to the question asked by the EMS,

Table A.1: Compliance of social media analysis platforms to the criteria deduced from this research

Social media analysis platform	Trusted network	Quality of information insurance	Important EMSs question answering
AIDR	×	×	√
Tweak-the-Tweet	×	√	×
Ushahidi	√	√	×
TweetTracker	×	×	×
TweetCred	√	×	×

which usually triggers new data collection steps and questions. For a social media analysis platform to be efficient in a crisis situation, it needs to focus on answering EMSs' questions and needs. To summarize, the list of question that social media analysis platform need to answer to be an efficient tool for EMS during a crisis situation are:

To summarize, the list of question that social media analysis platform need to answer to be an efficient tool for EMS during a crisis situation are:

- Can social media analysis platforms establish a trusted network of people to get the information from?
- Can social media analysis platforms ensure that only high-quality messages shared by the network are treated by the platform?
- Can social media analysis platforms analyse social media data in ways that provide relevant answers to the question asked by the EMS?

Table A.1 shows how the current social media analysis platform discussed in section IV.2 comply to the criteria discussed in this section. Many of these approaches are just data driven classification of social media message. The table shows the gap that still exists between what the social media analysis platform ca provide and what the EMSs require.

VI. Conclusion

The information available on social media during a crisis, from textual information to information in images and videos, can improve the current information gathering procedures of the EMSs, help them get a better overview of the situation, assess the crisis response effort as well as discover what the people in the affect areas need. Despite this information, EMSs in Norway are still reluctant to use it as

an information source. In this paper, we investigated the reasons behind this reluctance via interviews with four major EMSs in Norway. As a result of this research, we were able to assess what the social media analysis platforms currently lack and should provide to address the concerns of the EMSs. The criteria are a trusted network, information quality assurance, and answering the meaningful question for the EMSs.

Acknowledgements

We would like to thank our interviewees (the Agder police district chief of staff, the crisis preparedness leader at Kristiansand municipality, the head of a unit at the Grimstad fire brigade, and the Grimstad Red Cross volunteer) for their collaboration and patience in answering all our questions.

Paper A: References

[1] A. Acar and Y. Muraki. Twitter for crisis communication: lessons learned from japan's tsunami disaster. *International Journal of Web Based Communities*, 7(3):392–402, 2011.

[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *Proceedings of the International Conference on Web Search and Web Data Mining - WSDM '08*, page 183, New York, New York, USA, 2008. ACM Press.

[3] C. A. Cassa, R. Chunara, K. Mandl, and J. S. Brownstein. Twitter as a sentinel in emergency situations: lessons from the boston marathon explosions. *PLoS currents*, 5, 2013.

[4] E. Fichet, J. Robinson, and K. Starbird. Eyes on the Ground: Emerging Practices in Periscope Use during Crisis Events. *Proceedings of the 13th International ISCRAM Conference*, (Hardy 2013), 2015.

[5] J. Fohringer, D. Dransch, H. Kreibich, and K. Schröter. Social media as an information source for rapid flood inundation mapping. *Natural Hazards and Earth System Sciences*, 15(12):2725–2738, dec 2015.

[6] H. Gao, G. Barbier, and R. Goolsby. Harnessing the Crowdsourcing Power of Social Media for Disaster Relief. *IEEE Intelligent Systems*, 26(3):10–14, may 2011.

[7] V. S. M. W. Group et al. Next steps: Social media for emergency response. *Homeland security science and technology*, 2012.

[8] V. S. M. W. Group et al. Lessons learned: Social media and hurricane sandy. *Homeland security science and technology*, pages 1–39, 2013.

[9] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. Tweetcred: A real-time web-based system for assessing credibility of content on twitter. In *Proc. 6th International Conference on Social Informatics (SocInfo). Barcelona, Spain*, 2014.

[10] Q. Huang and G. Cervone. Usage of social media and cloud computing during natural hazards. *Cloud Computing in Ocean and Atmospheric Sciences*, pages 297–324, 2016.

[11] M. Imran, C. Castillo, F. Diaz, and S. Vieweg. Processing social media messages in mass emergency: a survey. *ACM Computing Surveys (CSUR)*, 47(4), 2015.

[12] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg. Aidr: Artificial intelligence for disaster response. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, 2014.

[13] A. Lambert. Disaster data assemblages: Five perspectives on social media and communities in response and recovery. In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pages 2237–2245. IEEE, 2016.

[14] Y. Liu and R. Raveendranath. Social media analysis for crisis management: A brief survey.

[15] P. Mayring. Qualitative content analysis/philipp mayring. In *Qualitative content analysis. Forum: Qualitative Social Research*, volume 1, 2000.

[16] M. Mendoza, B. Poblete, and C. Castillo. Twitter under crisis: Can we trust what we RT? In *Proceedings of the First Workshop on Social Media Analytics - SOMA '10*, SOMA '10, pages 71–79, New York, New York, USA, 2010. ACM Press.

[17] A. Mohaisen, H. Tran, N. Hopper, and Y. Kim. Understanding social networks properties for trustworthy computing. In *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pages 154–159. IEEE, 2011.

[18] T. H. Nazer, G. Xue, Y. Ji, and H. Liu. Intelligent disaster response via social media analysis a survey. *ACM SIGKDD Explorations Newsletter*, 19(1):46–59, 2017.

[19] B. Nerlich and R. Jaspal. Images of extreme weather: Symbolising human responses to climate change. *Science as Culture*, 23(2):253–276, 2014.

[20] T. Norri-Sederholm, M. Joensuu, and A.-M. Huhtinen. Ensuring Information Flow and the Situation Picture in Public Safety Organisations Situation Organisations Situation Centres. In *Proceedings of the 16th European Conference on Cyber Warfare and Security*, pages 267–273, 2017.

[21] O. Oh, M. Agrawal, and H. R. Rao. Information control and terrorism: Tracking the mumbai terrorist attack through twitter. *Information Systems Frontiers*, 13(1):33–43, 2011.

[22] L. Palen and K. M. Anderson. Crisis informaticsnew data for extraordinary times. *Science*, 353(6296):224–225, 2016.

[23] L. Palen and S. B. Liu. Citizen communications in crisis: anticipating a future of ict-supported public participation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 727–736. ACM, 2007.

[24] J. Radianti, S. R. Hiltz, and L. Labaka. An overview of public concerns during the recovery period after a major earthquake: Nepal twitter analysis. In *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pages 136–145. IEEE, 2016.

[25] M. K. Rahman and M. A. Adnan. Dynamic weight on static trust for trustworthy social media networks. In *Privacy, Security and Trust (PST), 2016 14th Annual Conference on*, pages 62–69. IEEE, 2016.

[26] A. Skuse and T. Brimacombe. Social networking, social media and complex emergencies: Issues paper, 2014.

[27] K. Starbird. Digital volunteerism during disaster: Crowdsourcing information processing. In *Conference on human factors in computing systems*, pages 7–12, 2011.

[28] R. Subba and T. Bui. Online convergence behavior, social media communications and crisis response: An empirical study of the 2015 nepal earthquake police twitter project. In *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[29] M. Taylor, G. Wells, G. Howell, B. Raphael, et al. The role of social media as psychological first aid as a support to community resilience building. *Australian Journal of Emergency Management, The*, 27(1):20, 2012.

[30] C. Torrey, M. Burke, M. Lee, A. Dey, S. Fussell, and S. Kiesler. Connected giving: Ordinary people coordinating disaster relief on the internet. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, pages 179a–179a. IEEE, 2007.

[31] R. M. Tripathy, A. Bagchi, and S. Mehta. A study of rumor control strategies on social networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1817–1820. ACM, 2010.

[32] I. Tuomi. Data is more than knowledge: Implications of the reversed knowledge hierarchy for knowledge management and organizational memory. *Journal of Management Information Systems*, 16(3):103–117, 1999.

[33] D. Velev and P. Zlateva. Use of social media in natural disaster management. *Intl. Proc. of Economic Development and Research*, 39:41–45, 2012.

[34] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen. Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1079–1088. ACM, 2010.

[35] M. Yadav and Z. Rahman. The social role of social media: the case of Chennai rains-2015. *Social Network Analysis and Mining*, 6(1):1–12, 2016.

[36] D. Yates and S. Paquette. Emergency knowledge management and social media technologies: A case study of the 2010 haitian earthquake. *International journal of information management*, 31(1):6–13, 2011.

[37] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power. Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems*, 27(6), 2012.

[38] A. Zielinski, S. E. Middleton, L. N. Tokarchuk, and X. Wang. Social media text mining and network analysis for decision support in natural crisis management. *Proc. ISCRAM. Baden-Baden, Germany*, 2013.

Paper B

B

Title:	Combining a Context Aware Neural Network with a Denoising Autoencoder for Measuring String Similarities
Authors:	Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo
Affiliation:	Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
Journal:	*Computer Speech and Language*, vol. 20, no. 101028, Marsh 2020.
Copyright ©:	Elsevier

Combining a Context Aware Neural Network with a Denoising Autoencoder for Measuring String Similarities

Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo

***Abstract* — Measuring similarities between strings is central for many established and fast-growing research areas, including information retrieval, biology, and natural-language processing. The traditional approach to string similarity measurements is to define a metric with respect to a word space that quantifies and sums up the differences between characters in two strings; surprisingly, these metrics have not evolved a great deal over the past few decades. Indeed, the majority of them are still based on making a simple comparison between character and character distributions without considering the words context. This paper proposes a string metric that encompasses similarities between strings based on (1) the character similarities between the words, including non-standard and standard spellings of the same words, and (2) the context of these words. We propose a neural network composed of a denoising autoencoder and what we call a context encoder, both specifically designed to find similarities between the words based on their context. Experimental results show that the resulting metrics have succeeded in 85.4% of the cases in finding the correct version of a non-standard spelling among the closest words, compared to 63.2% using the established Normalised-Levenshtein distance. We also show that by employing our approach, the words used in similar context are calculated to be more similar than words with different contexts, which is a desirable property lacking in established string metrics.**

Keywords— String Metric, Neural Network, Autoencoder, Context Encoder.

I. Introduction

Measuring similarities between strings is an essential component of a large number of language and string processing tasks, including information retrieval, natural biology, and natural language processing. A string metric is a function that quantifies the distance between two strings. The most widely known string metric is the edit distance one, also known as the Levenshtein distance, which represents the number of substitution insertions, or deletions operations needed to transform one string into another [1]. The fewer operations needed to move from one string to another, the more similar the two strings are.

Interestingly, string metrics are key to comprehending the approximate string matching problem present in many fields. As an example, natural language processing needs automatic spelling correction, and in bioinformatics, finding similarities between DNA sequences is a crucial task; However, both tasks are string approximation problems. Common to all string similarity metrics is the fact that they are used to find matching patterns for a string that has undergone a certain amount of distortion, including, but not limited to, misspellings, abbreviations, slang or irregularities in DNA sequences. The focus in this paper is on a novel distance metric that has certain advantageous properties not present in other string metrics, including misspellings, nonstandard usage of words, and the string context itself.

Of course, the evolution and distortion of languages is nothing new. However, one consequence of the global social media era is the non-standardization of languages, which means that the same phrase – and even the same word – can be communicated in a variety of ways within the same language. This evolution presents a challenge to natural language processing as any data handling, such as translation classification, has become less formal. Natural language processing itself would be much easier if everyone wrote the same way which is unrealistic. One mitigation of this situation is to normalize non-standard words to a more standard format that is easier to handle.

To normalize misspellings, non-standard words and phrases, abbreviations, dialects, sociolects and other text variations (referred to here as non-standard spelling and non-standard words), three approaches other than string metrics are currently available in the literature.

The first method is to view normalization of any non-standard spelling as a translation problem [2]. One example of this is based on statistical tools that map the non-standard words with their English counterparts based on a probability distribution [3]. Certainly, the translation method is a promising strategy, however, using a method designed to capture the complex relationship between two different

languages in the context of word normalization is an "overreach" given the strong relationship between the English words and their non-standard forms.

A second approach for solving the non-standard spelling challenge is to consider it initially as plain spell checking. The method tries to correct the misspelled words based a probability model [2]. Yet, the challenge with the latter is that the difference between non-standard and standard spelling is substantial. Hence, words far from all normalizations may be viewed as corrections.

Third, normalizing non-standard spelling may be viewed as a speech recognition problem [2] [4]. In this approach, the texts are regarded as a phonetic approximation of the correctly spelled message. The factor inspiring this view is that many non-standard spellings are written based on their phonetic rather than normative spelling. However, this view is also an oversimplification of the texts nature, as it contains non-standard spellings that are not merely phonetic spellings of the correct word. For example, texts containing abbreviation (lol for laugh out loud), truncated words (n for and), and leetspeak (4ever for forever) cannot be handled by this approach.

This paper proposes a new method that maps each word in a vocabulary into a real vectors space. As a result, the distance between two words will be the distance between the vector representation of those words in the real vector space. The mapping between the word vocabulary and the real vector space must satisfy two premises.

The first premise is that the distance between the vector representations of a non-standard word and its corrected form should be shorter than the distance between the non-standard word and any other unrelated known word. To achieve this constraint, the vector representation needs to spur positive correlations between a corrected spelling and *every* possible non-standard spelling of that word, while simultaneously minimizing the correlation with all other words and *their* non-standard spellings. It important to note here that by utilizing this premise, it will not straightforward to generalize the proposed string metric to non-NLP applications like DNA sequences.

The second premise is that the vector representation of a word should also be such that words with similar meaning have similar representations. We assume that two words have a similar meaning when they are often used in the same context. The context of a word is determined by the words surrounding it. To obtain this representation, we combined a predictive word, embedding methods with a denoising autoencoder [5]. A denoising autoencoder is an artificial neural network that takes a data set as input, adds some noise to the data, then tries to reconstruct the initial data from the noisy version. By performing this reconstruction, the denoising

autoencoder discerns the feature present in the initial data in its hidden layer. In our approach, we consider the non-standard spellings to be the noisy versions of the corrected word forms. In a predictive word embedding method, each word is represented by a real-valued vector based on the usage of words and their context. A neural network learns this real-valued vector representation in a way that minimizes the loss of predicting a word based only on its context. This representation is in contrast with the representation in a bag of words model, where, unless explicitly managed, different words have different representations, regardless of their use [6].

II. Background

A string metric or string distance function, defines a distance between every element of a set of strings A. Any distance function d on A must satisfy the following conditions for all x, y, and $z \in A$ [7]:

$$d(x, y) \geq 0 \quad \text{non-negativity} \tag{B.1}$$

$$d(x, y) = 0 \iff x = y \quad \text{identity of indiscernibles} \tag{B.2}$$

$$d(x, y) = d(y, x) \quad \text{symmetry} \tag{B.3}$$

$$d(x, z) \leq d(x, y) + d(y, z) \quad \text{triangle inequality} \tag{B.4}$$

Hence, the comparison between any two strings is larger or equal to 0 (Equation B.1), identical strings have distance 0 (Equation B.2), the distance between two strings is independent of whether the first is compared to the second or vice versa (Equation B.3), and the distance between two strings is always equal to or smaller the inclusion of a third string in the measurements (Equation B.4).

Over the years, several attempts at defining an all-encompassing string metric have been carried out. The most well-known of these is the edit distance (Levenshtein distance), which has been one of the most widely used string comparison functions since its introduction in 1965 [1]. It counts the minimum number of operations (deletion, insertion and substitution of a character) required to transform one string into another. It also assigns a cost to each operation. For example, if the weight assigned to the operation is one, the distance between the words “vector” and “doctor” is two, since only two substitutions are required for a transformation. The edit distance satisfies all the requirements as a distance function (equations B.1,B.2,B.3 and B.4).

The edit distance is called a simple edit distance when all operations have the same cost and a general edit distance when operations have different costs. Other

than that, the edit distance has four notable variants. First, a longest common subsequence (LCS) is when only insertions and deletions are allowed with cost one [8]. A second simplification is a variant that only allows substitution. In this case, the distance is called the Hamming distance [9]. Third, the Damerau-Levenshtein Distances adds the transposition of two adjacent characters to the operations allowed by the edit distance [10]. Finally, the episode distance allows only insertions that cost 1. The episode distance is not symmetrical and does not satisfy Equation B.3. Since insertions do not allow to transforming a string x to y if x is longer than y, $d(x, y)$ is either $|y| - |x|$ or ∞.

In 1992, Ukkonen [11] introduced the q-gram distance. This method is based on counting the number of occurrences of common q-grams (strings of length $q \in \mathbb{N}$) in each string, the strings having a closer distance the more q-grams they have in common. The q-gram distance is not a metric function, since it does not obey the identity of indiscernible requirement (Equation B.2).

Later, Kondrak [12] developed the notion of N-gram distance in which he extended the edit and LCS distance to consider the deletions, insertions, and substitutions of N-grams. The use of N-grams enabled a certain number of new statistical methods for string metrics originating from the field of samples and sets. The use of N-gram introduces the notion of statistical string metrics, which are metrics that measure the statistical properties of the compared strings. For example, the Sorensen-Dice coefficient was used as a metric to measure the similarity between two strings [13] [14], initially a method used to compare the similarity between two samples. In the case of strings the coefficient is computed as follows:

$$d(x, y) = \frac{2n_t}{n_x + n_y} \tag{B.5}$$

where n_t is the number of character N-grams found in both strings, n_x is the number of N-grams in string x and n_y is the number of N-grams in string y.

The Jaccard Index is another statistical method used to compare the similarity between two sample sets, including strings. It is calculated as one minus the quotient of shared N-grams by all observed N-grams in both strings.

Certain vector similarity functions have been extended to include string similarity as well, of which the most notable is the string cosine similarity measuring the cosine similarity between vector representations of the two strings being compared. As regards English words, the vectors have a size 26, one element for each character, and the number of occurrences of each character in each string.

The use of machine learning techniques for vector representations of words has

been around since 1986 thanks to the work of Rumelhart, Hinton, and Williams [15]. The string similarity measurements are used as features in supervised, natural language processing tasks to increase the performance of the classifier. More recently, a method called local linear embedding has been introduced. This method computes low-dimensional, neighborhood-preserving embedding of high dimensional input. The method is applied to generate a two-dimensional embedding of words that preserves their semantics [16].

Subsequently, feedforward neural networks have been used to generate a distributed vector representation of words [17]. By predicting the next word giving the previous words in the context, the neural network learns a vector representation of the words in its hidden layer. The method is extended to take into consideration the surrounding words not only the previous words [18].

In the same context, the feedforward neural network is replaced by a restricted Boltzmann machine to produce the vector representations [19]. A word vector representation variant learns for each word a low dimensional linear projection of the one-hot encoding of a word by incorporating the projection in the energy function of a restricted Boltzmann machine [20] [21].

Finally, GloVe is one of the most successful attempts at producing vector representations of words for string comparisons [22]. GloVe learns a log-bi-linear model that combines the advantages of global matrix factorization and local context window to produce a vector representation of word based on the word count. A vector similarity measure, for instance the Euclidean distance, cosine similarity, or L_1 measure, may then be used to measure the similarity between two strings.

III. Word coding approach

The objective of this research is to find a function F that maps words into real vector space in such a way that the distance between two similar words (i.e., non-standard spellings of the same word or words used in the same context) will be the shortest distance between the corresponding mapping in the real vector space. To achieve this goal, F needs to obey two constraints. The first constraint is that the distance in real vector space between the mapping of a word and its non-standard versions must be shorter than the distance between that word and non-standard versions of other words. The second constraint is that the distance in real vector space between the mapping of words with similar meanings must be shorter than the distance between words with dissimilar meanings. We use context as a determining factor of a words meaning: we assume that words used in the same context have a similar meaning. To model the first constraint, we have used a denoising autoencoder, and to model the second constraint, we have introduced a context encoder.

The denoising autoencoder and the context encoder are explained in sections III.1 and III.2 respectively. The overall method is explained in section III.3. A summary of all the notation, parameters, and functions used in this section is summarized in Appendix A.

III.1 Denoising autoencoder

An autoencoder is an unsupervised learning algorithm based on artificial neural networks in which the target value is equal to the input [5]. An autoencoder can in its simplest form be represented by a network composed of the following:

- An input layer representing the feature vector of the input
- A hidden layer that applies a non-linear transformation of the input
- An output layer representing either the target value or label

Suppose we have a training example x, the autoencoder tries to learn a function id such that $id(x) \simeq x$, an approximation of the identity function. The identity function seems to be a trivial function to learn; however, if we put some constraints on the autoencoder, it can learn a function that captures features and structures in the data. For example, limiting the number of hidden units in the network to be less than the input units forces the network to learn a compressed representation of the input. Instead of copying the value of the input in the hidden layer, the network must discern which parts of the input are more important and will lead to a better reconstruction. Adding noise to the input is another constraint that forces the autoencoder to discern the datas most relevant features. By reconstructing the data based on its noisy version, the autoencoder undoes the effect of the noise. This undoing of the noisy effects can only be performed when the autoencoder has discerned the statistical dependencies between the different types of input. Regarding the latter example, the autoencoder is called a denoising autoencoder.

In our approach, we insert the non-standard spelling into a denoising autoencoder and try to reconstruct the original word. Any non-standard spelling of a word may be seen as a noisy version of the original word. The aim is that the network should learn two essential features: (1) the relationship between non-standard and standard word spellings, and (2) what separates the standard words. Both features should lie in the hidden layer, which is used to reconstruct the standard word from the non-standard spellings.

The denoising autoencoder in our approach includes a vocabulary $A = \{a_1, a_2, ..., a_r\}$ of r words, which can be standard English[1] words and non-standard variants of

[1]The approach is not limited to English. However, all our examples are from the English language.

those words. A consists of the following subsets: The standard words $C = \{c_1, c_2, ..., c_m\} \subset A$, and the non-standard spelling of every word $c_i \in C$ as $M_{c_i} \subset A$. We define an initialization function v that transforms a word in A into a vector of real numbers in $\mathbb{R}^n$. v can be a function that performs a one-hot encoding of the words in A, or it can map each character in a word to a unique number, or assign a random vector to each word. v can be presented by a $\mid A \mid \times n$ matrix of free parameters.

The input of the denoising autoencoder is the non-standard word spelling $m_j \in M_{c_i}$ corresponding to word c_i, and the output of the hidden layer is $h(v(m_j))$. The reconstructed word produced by the autoencoder, $\tilde{\boldsymbol{c_i}}$ should ideally be equal to the c_i. The details are presented in Equation B.6 where W is a matrix of weights, o is an activation function, and $\boldsymbol{b}$ is a bias term. Each element w_{pq} in W is associated with the connection between the pth element of $v(m_j)$ and the qth hidden unit of the autoencoder.

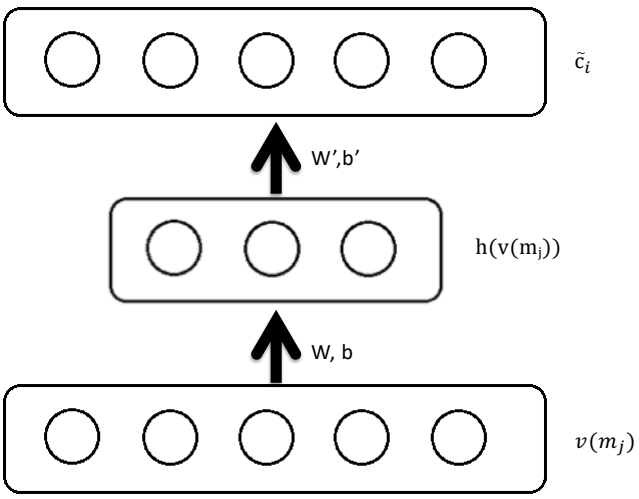
$$h(v(m_j)) = o(Wv(m_j) + \boldsymbol{b}) \tag{B.6}$$

The reconstruction $\tilde{\boldsymbol{c_i}}$ of the original word $\boldsymbol{c_i}$ by the output layer of the autoencoder is given by Equation B.7 where W' is a matrix of weights. Each element w'_{pq} in W' is associated with the connection between the pth hidden unit of the autoencoder and the qth element of the reconstruction $\tilde{\boldsymbol{c_i}}$. $\boldsymbol{b'}$ is a bias term.

$$\tilde{\boldsymbol{c_i}} = o(W'h(v(m_j)) + \boldsymbol{b'}) \tag{B.7}$$

The overall architecture of the denoising autoencoder is presented in Figure B.1.

Figure B.1: Overall architecture of the denoising autoencoder



Regarding a distance function d in real vector space, the autoencoder learns the parameters W, $\boldsymbol{b}$, W', and $\boldsymbol{b'}$ that minimize the loss function L, which is given by the distance between the initialization $v(m_j)$ of the non-standard version of c_i and its reconstruction $\boldsymbol{\tilde{c_i}}$ (Equation B.8).

$$L = d(\tilde{c}_i, v(c_i)) \tag{B.8}$$

The output of the encoders hidden layer may be fed as input into another autoencoder, which then tries to reconstruct it. In this case, the second autoencoder discerns features about the features learned by the first autoencoder: It discerns a second-degree feature abstraction of the input data. This process of stacking autoencoders may be repeated indefinitely. The obtained network is called a deep belief network [5]. With respect to each layer, all elements in W, $\boldsymbol{b}$, W', or $\boldsymbol{b'}$ are updated using back-propagation and stochastic gradient the descent [23].

III.2 Context based coding

To increase the relevance of a denoising autoencoder, we connect each with their context, which means the text close to the word used in a setting. We define the context as a sequence $a_1, a_2, ..., a_T$ of words $a_t \in A$. The objective of the context based encoding is to learn a model g representing the probability of a word given its context so that $g(a_t, a_{t-1}, ..., a_{t-s-1}) = P(a_t|a_{t-1}, ..., a_{t-s-1})$. g presents the likelihood of the word a_t appearing after the sequence $a_{t-1}, ..., a_{t-s-1}$. This method was first introduced by Bengio et al. [17]. We have divided the function g into two parts:

1. A mapping u from an element $a_i \in A$ to a vector $u(a_i)$, which represents the vector associated with each word in the vocabulary.

2. A probability function f over vector representation of words assigned by u. f maps an input sequence of vectors representation of words in a context, $(u(a_t), ..., u(a_{t-s-1}))$, to a conditional probability distribution over words in A for the next word a_t. Thus, $g(a_t, a_{t-1}, ..., a_{t-s-1}) = f(u(a_t), ..., u(a_{t-s-1}))$

Hence, the function g is a composite of the two functions, u and f. A certain number of parameters are associated with each of these two parts. The parameters of u are the elements of the matrix U presenting the words vector representations. The function f may be implemented by a neural network with parameters ω. Training is achieved by looking for $\theta = (\omega, U)$ which maximizes the training corpus log-likelihood:

$$L = \frac{1}{T} \sum_t log(g(a_t, a_{t-1}, ..., a_{t-n})) \tag{B.9}$$

The neural network presenting f has a softmax output layer, which guarantees positive probabilities adding up to 1:

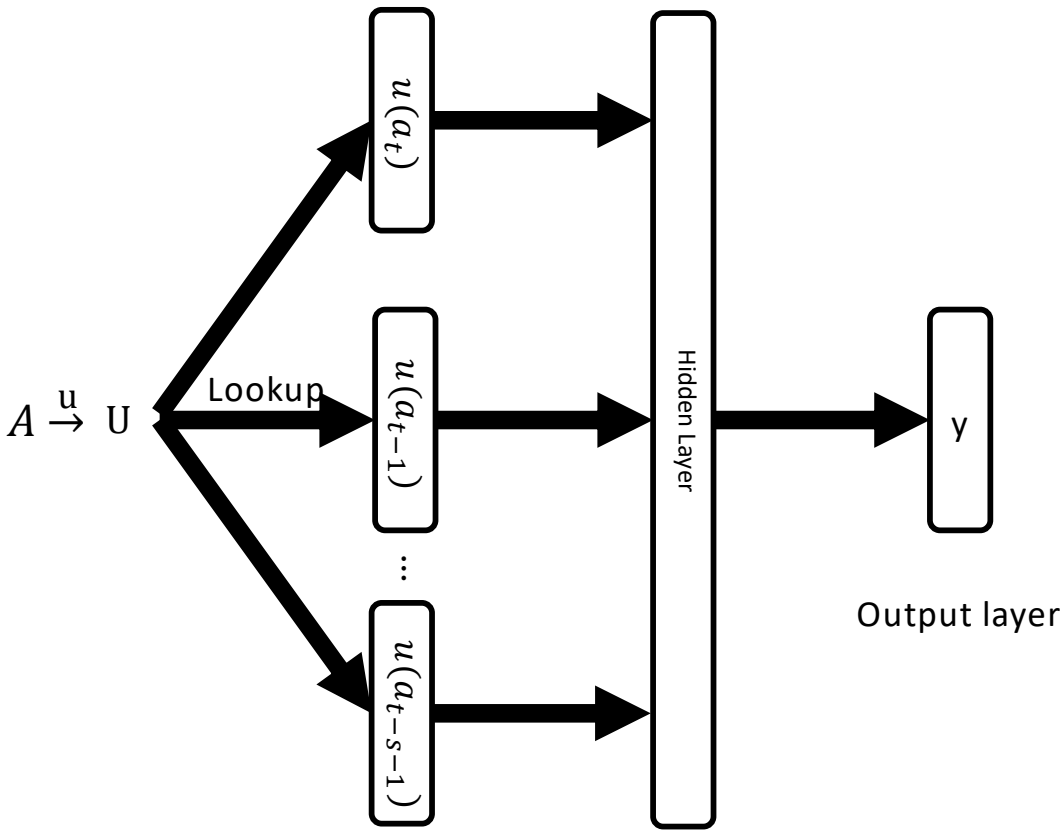
$$P(a_t|a_1, a_2, ..., a_{t-1}) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{B.10}$$

y is the output of the neural networks hidden layer.

$$y = \tilde{\boldsymbol{b}'} + \tilde{W'}o(\tilde{\boldsymbol{b}} + \tilde{W}x) \tag{B.11}$$

where o is the activation function of the hidden layer, $\tilde{W'}$, $\tilde{W}$, and U are the matrix of weights, $\tilde{\boldsymbol{b}'}$ and $\tilde{\boldsymbol{b}}$ are the biases, and x is feature vector of word vector representation from the matrix H: $x = (u(a_{t-1}), u(a_{t-2}), ..., u(a_{t-n}))$. The parameters of the model are $\theta = \tilde{\boldsymbol{b}'}, \tilde{\boldsymbol{b}}, \tilde{W'}, \tilde{W}, U$, and the overall architecture of the context encoder is presented in Figure B.2.

Figure B.2: Overall architecture of the context encoder



III.3 Distance over word space

If h and v are bijections, the denoising autoencoder transformation $h(v(a_i))$ of the initialization $v(a_i)$ of a word $a_i \in C$ is a bijection from the word space C to

$h(v(C)) \subset \mathbb{R}^n$. This observation provides the function D_a, producing the distance between the autoencoding representations for the words a_i and a_j (Equation B.12), with the property of a metric in C. First, the non-negativity, triangular inequality, and symmetry of D are derived from the same properties of the d. Secondly, the identity of the indiscernible is automatically deduced from the observation that h and v are bijective functions. The advantage of this distance is that the function h contains in its weight matrix W and bias $\boldsymbol{b}$, an encoding that captures the stochastic structure of misspelling patterns representing the words observed the autoencoders learning phase. It is important to note that D_a is only a pseudometric on A, since the purpose of the autoencoder is to minimize the distance between vector representation of a correct word and its non-standard version. Therefore, h is no longer a bijection form A to $h(v(A))$, and the identity of indiscernibles cannot be guaranteed.

$$D_a(a_i, a_j) = d(h(v(a_i)), h(v(a_j))) \tag{B.12}$$

In order to arrive at the mapping F, the Matrix U in section III.2 may be updated by combining the autoencoder in section III.1 and the context encoder; in this case, both methods will work in a parallel manner to update the words vector representation.

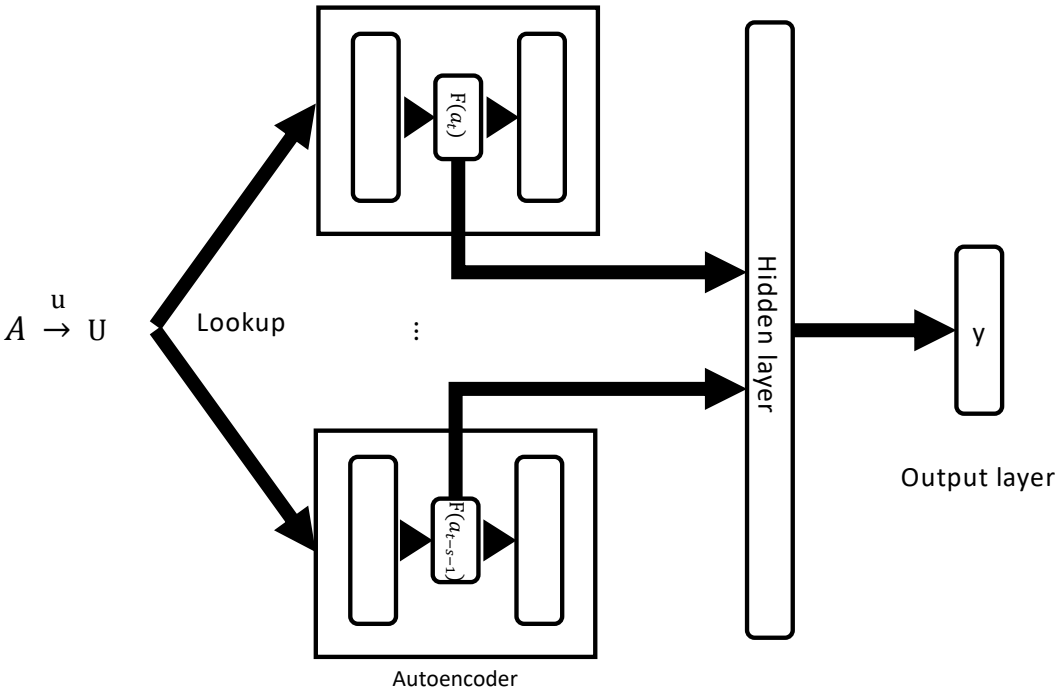
Thus, the vector representations of the words in the matrix U are learned using the context method, and the vector representations of non-standard words in U are also calculated based the autoencoder (see Figure B.3). The denoising autoencoder we have used, in this case, is a seven-layer deep autoencoder. The initialization function v is a one-hot encoding of the words in A, resulting in an input layer of the denoising autoencoder of $|A|$ nodes (not shown in Figure B.3 for the sake of presentability). The combination of the autoencoder and context coding to produce the mapping F is used to define the function D_c on A in Equation B.13. D_c may be seen as an extension of D_a; however, it includes the context of the words. D_c is a function that finds the distance between the words a_i and a_j using the function F. D_c is not a metric in A for the same reason D_a is not, nor is it a metric in C, since the context encoding produces similar vector presentation to words used in the same context; therefore, the identity of indiscernibles is not guaranteed here, either.

$$D_c(a_i, a_j) = d(F(a_i), F(a_j)) \tag{B.13}$$

IV. Results and discussion

To test our approach, we used a data set composed of the 1051 most frequently

Figure B.3: Overall architecture of the autoencoder in combination with the context encoder to find the similarity between the words



used words from Twitter paired with their various misspellings, drawing from a data set initially used in an IBM data normalization challenge in 2015 [2]. To train the context-based encoder, we used a data set containing 97191 different sentences with vocabulary words and their non-standard form. It is important to note that the data is imbalanced: Some words have only one non-standard form, while other words have multiple non-standard forms. This imbalance may potentially introduce certain challenges, since the autoencoder might not discern accurate features of words having a few non-standard versions. Since we used softmax units, the number of nodes in the last hidden layer of the autoencoder is set to $11 \simeq log_2(1051)$, which also represents the length of the obtained encoding vector. The autoencoder is trained using minbatch gradient descent containing batches of 100 examples each and a learning rate of 0.01. The closest standard word has been picked as the most likely standard version of the non-standard spelling. By using the term "closest, we mean the word that has the shortest distance D_a or D_c to the non-standard spelling as defined in subsection[3] III.3.

[2]The data is available here: `https://noisy-text.github.io/norm-shared-task.html`

[3]The code and data part of these experiments are available here `https://github.com/mehdi-mbl/WordCoding`

Table B.1: Performance comparison including our two approaches: D_a Denoising autoencoder and D_c the combination of autoencoder and context encoder

Distance	**Closest word**	**5th closest word**
Cosine Similarity	46.33%	60.22%
Q-Gram	47.57%	62.41%
Srensen-Dice coefficient	47.85%	60.03%
Edit distance	55.75%	68.22%
Weighted-Levenshtein	55.85%	67.93%
Damerau-Levenshtein distance	56.51%	68.03%
N-Gram	58.23%	76.49%
Metric-Longest Common Subsequence	60.89%	75.73%
Longest Common Subsequence	61.37%	74.31%
Normalised-Levenshtein	63.17%	78.30%
D_a with Cosine similarity	83.82%	89.53%
D_c with L_1 distance	76.37%	81.53%
D_c with Euclidean distance	82.71%	87.35%
D_c with Cosine similarity	**85.37%**	**89.61%**

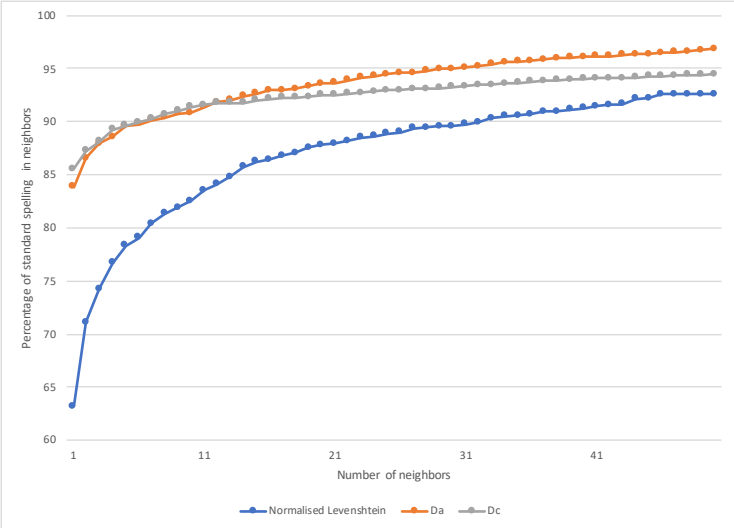
Table B.1 compares the results produced by our distance metrics D_a, D_c, with the existing string metrics presented in Section II in finding the correct version of a non-standard spelling. The table shows a huge increase in accuracy, ranging from 63.17% for the best metric available in the literature (Normalised-Levenshtein) to 85.37% when D_c is used. The reason is that unlike the state-of-the-art metric, D_c captures stochastic word patterns shared between the nonstandard word and its correct form. Figure B.4 shows the percentage of finding the correct spelling of a non-standard word in its first neighbors x using the Normalised-Levenshtein D_a, and D_c. The x axis represents the number of neighbors. Figure B.4 shows that after ten neighbors, D_a starts to outperform D_c because D_a has been modeled by an autoencoder whose the main purpose is to model such non-standard words. Using, D_c, as we move further away from a word, the nearest word will contain words used in a similar context which are not necessarily standard versions of the word (see table B.3).

Our approach has not been limited to one vector distance. In fact, neural representation parts of the inner autoencoder may be measured with any vector similarity.

Table B.1 also compares the performance of D_c with different vector distances. The cosine similarity produces the best performance in this task, producing a result of 85.37%.

Table B.2 shows the closest correct word to a sample of non-standard spellings

Figure B.4: Performance of different distances in finding the standard form of a non-standard word



utilizing the autoencoder D_a without having any context encoder. Table B.2 shows the results that the closest words share a certain number of patterns with their non-standard counterparts. For example, the four closest word to "starin" all end with "ing", and three of these start with "s" (staring, praying sucking, slipping). Notice also the similarity in character between the two closest words "staring" and "praying". The same may be said about the closest word to "ddnt". In the case of "omg" and "justunfollow", all the closest words are combinations of more than one word, which suggests that the autoencoder has determined that they are abbreviations or combinations of words. The next examples in Table B.2 presents a non-standard spelling for which the approach with denoising autoencoder fails to recognize the correct version in the five closest words: while the correct version of "wada" is "water", our algorithm chooses "wanna" as the closest correct version. So even though it is not the correct assumption, the resemblance between "wada" and "wanna" justifies this guess; moreover, a human being could arguably have made the same mistake. The same may also said about "bea" and "the". As regards "dats", the algorithm chooses the correct word as the fourth closest word. However, the first choice ("what's") may also be justified, since it shares three characters with the non-standard spelling.

Table B.3 shows the closest words to a sample of other words in terms of distance D_c. In addition to the non-standard spelling being close to the standard word, Table B.3 shows that words similar in meaning are also introduced to the closest words. For example, the third closest word to "dogg" is "cat" both being domestic animals. Notice also that "boy" and "kid" come next because in many of the train-

Table B.2: Example of words and their closest standard form using denoising autoencoder

Non-standard spellings	Closest word	2nd closest word	3rd closest word	4th closest word	5th closest word	Correct word
thng	**thing**	there	wanna	right	where	thing
starin	**staring**	praying	sucking	slipping	weekend	staring
omg	**oh my god**	at least	in front	in spite	what's up	oh my god
ddnt	**didn't**	that's	aren't	what's	better	didn't
justunfollow	**just unfollow**	what about you	ultra violence	direct message	what are you doing	just unfollow
wada	wanna	sugar	sense	never	speed	water
dats	what's	wasn't	aren't	**that's**	give a	that's
bea	the	why	kid	old	yes	tea

Table B.3: Example of words and their closest standard form using combination of denoising autoencoder and context coding

Non-standard spellings	Closest word	2nd closest word	3rd closest word	4th closest word	5th closest word
dogg	dog	doog	cat	boy	kid
txt	text	texting	txted	texted	work
teering	wearing	meeting	tearing	shaking	picking
tomorrow	tmrw	tmr	today	yesterday	judgment
video	vid	vids	videos	sleep	remix
birthday	bday	biryhday	birthdayyy	drinking	dropping
thng	ting	thing	think	right	stuff
starin	staring	looking	glaring	slipping	praying
omg	omgg	omfg	ohmygod	ohmygad	oh my god
ddnt	didn	didnt	didn't	havn't	aren't
wada	wanna	sugar	sense	never	speed
dats	dat	that	thats	thts	that's
bea	the	tea	yes	old	coffee

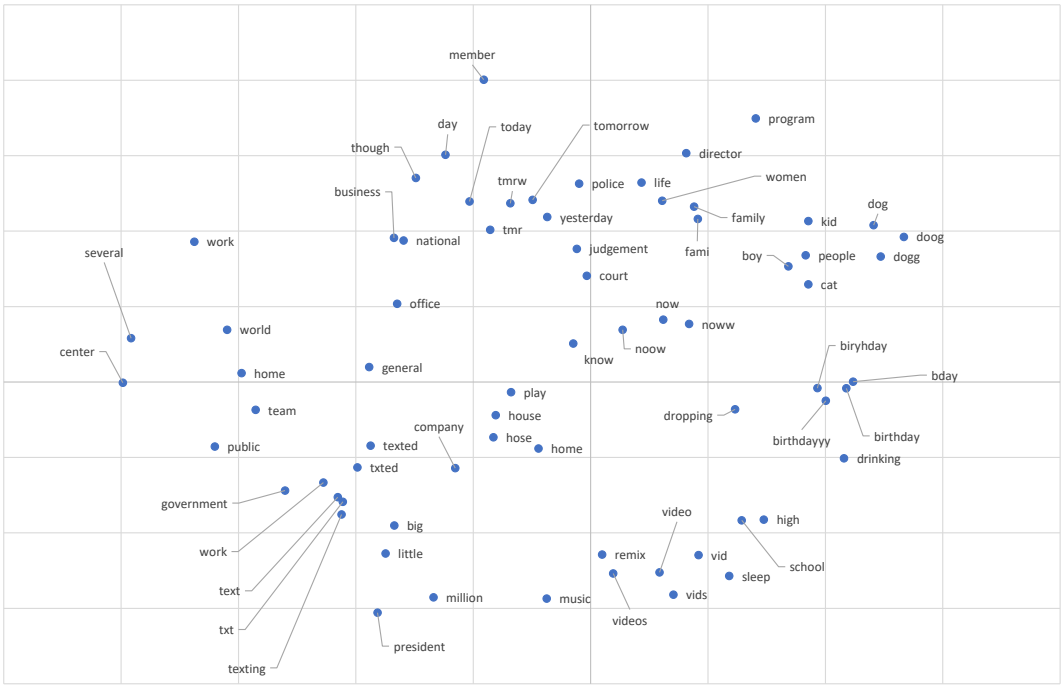
ing sentences a boy a kid is mentioned in the same context as a dog or cat. The same may be said about the closest word to "tomorrow". In the case of "txt" and "birthday", most of the closest words are their standard/ non-standard versions. The next examples in Table B.3 presents a non-standard spelling, "teering", for which the approach using the distance D_c fails to recognize a word with a close meaning to it in the five closest word. The correct version of "teering" is "tearing" which is the third closest word in table B.3. Even though the closest word to "teering" is not related to it, the resemblance between "weering" and "teering" can justify it being the closest word. Table B.3 shows that D_c finds the standard version of "bea" as the thirds closest word which is an improvement in this case over D_a.

Figure B.5 shows a scatter plot of the vector representation of the words presented in table B.3 after a dimension reduction. T-SNE [24] has been used for the dimension reduction.

V. Conclusion

In this paper, we have proposed combining a denoising autoencoder along with a context encoder to determine a mapping between vocabulary and real vector space. This mapping allows us in turn to define a metric in word space that encompasses non-standard spelling variations of words as well as words used in similar contexts.

Figure B.5: 2D plot of some example of vector representation of words using T-SNE



This work is a first attempt at defining a fully determined metric in word space using neural networks. Granted, the resulting metric does not satisfy all the theoretical properties of a metric. However, our experimental results have shown that the resulting metric succeeds in 85.4% of the cases in finding the correct version of a non-standard spelling a considerable increase in accuracy from the established Normalised Levenshtein rate of 63.2%. Moreover,, we have shown that words used in similar contexts have a shorter distance between them than words used in different contexts.

Appendix A: Model's parameters and functions

Parameters/functions	**Description**
A	Vocabulary
C	Set of standard words in A
M_{c_i}	Set of non-standard version of the word $c_i \in C$
v	Initialization function $A \rightarrow \mathbb{R}^n$
h	Output of the hidden layer of the denoising autoencoder
W, W'	Weights of the denoising autoencoder
$\boldsymbol{b}$, $\boldsymbol{b'}$	Biases of the denoising autoencoder
$\tilde{c}_i$	Output of the denoising autoencoder
L	Loss function
d	Metric in real vector space
o	Activation function
g	Probability distribution over A
u	Mapping function $A \rightarrow \mathbb{R}^n$
f	Probability distribution over mappings of words produced by u
y	Output of the context encoder
$\tilde{W}$, $\tilde{W}'$	Weights of the context encoder
$\tilde{\boldsymbol{b}}$, $\tilde{\boldsymbol{b'}}$	Biases of the context encoder
F	Mapping function $A \rightarrow \mathbb{R}^n$ learned by the combination of the denoising autoencoder and context encoder
D_a	Metric over A resulting from the denosing entoencoder mapping
D_c	Metric over A resulting from F

PAPER B: REFERENCES

[1] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, in: Soviet physics doklady, Vol. 10, 1966, p. 707.

[2] A. Aw, M. Zhang, J. Xiao, J. Su, A phrase-based statistical model for sms text normalization, in: Proceedings of the COLING/ACL on Main conference poster sessions, Association for Computational Linguistics, 2006, pp. 33–40.

[3] C. Kobus, F. Yvon, G. Damnati, Normalizing sms: are two metaphors better than one?, in: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 2008, pp. 441–448.

[4] D. Pennell, Y. Liu, Toward text message normalization: Modeling abbreviation generation, in: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 5364–5367.

[5] Y. Bengio, Learning deep architectures for ai, Foundations and trends® in Machine Learning 2 (1) (2009) 1–127.

[6] M. Dillon, Introduction to modern information retrieval: G. salton and m. mcgill. mcgraw-hill, new york (1983). xv+ 448 pp., 32.95 isbn 0-07-054484-0.

[7] M. Hazewinkel, Encyclopaedia of Mathematics, Springer Science & Business Media, 2013.

[8] D. Bakkelund, An lcs-based string metric, Olso, Norway: University of Oslo.

[9] R. W. Hamming, Error detecting and error correcting codes, Bell System technical journal 29 (2) (1950) 147–160.

[10] F. J. Damerau, A technique for computer detection and correction of spelling errors, Communications of the ACM 7 (3) (1964) 171–176.

[11] E. Ukkonen, Approximate string-matching with q-grams and maximal matches, Theoretical computer science 92 (1) (1992) 191–211.

[12] G. Kondrak, N-gram similarity and distance, in: International symposium on string processing and information retrieval, Springer, 2005, pp. 115–126.

[13] T. Sørensen, A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons, Biol. Skr. 5 (1948) 1–34.

[14] L. R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302.

[15] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Cognitive modeling 5 (3) (1988) 1.

[16] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[17] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, journal of machine learning research 3 (Feb) (2003) 1137–1155.

[18] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[19] A. Mnih, G. Hinton, Three new graphical models for statistical language modelling, in: Proceedings of the 24th international conference on Machine learning, ACM, 2007, pp. 641–648.

[20] G. E. Dahl, R. P. Adams, H. Larochelle, Training restricted boltzmann machines on word observations, arXiv preprint arXiv:1202.5695.

[21] G. E. Hinton, R. R. Salakhutdinov, Replicated softmax: an undirected topic model, in: Advances in neural information processing systems, 2009, pp. 1607–1614.

[22] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation., in: EMNLP, Vol. 14, 2014, pp. 1532–43.

[23] Y. A. LeCun, L. Bottou, G. B. Orr, K.-R. Müller, Efficient backprop, in: Neural networks: Tricks of the trade, Springer, 2012, pp. 9–48.

[24] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.

Paper C

Title:	A Neural Turing Machine for Conditional Transition Graph Modeling
Authors:	Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo
Affiliation:	Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
Journal:	*Submitted to IEEE Transactions on Neural Networks and Learning Systems*.
Copyright ©:	IEEE

C

A Neural Turing Machine for Conditional Transition Graph Modeling

Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo

***Abstract* — Graphs are an essential part of many machine learning problems such as analysis of parse trees, social networks, knowledge graphs, transportation systems, and molecular structures. Applying machine learning in these areas typically involves learning the graph structure and the relationship between the nodes of the graph. However, learning the graph structure is often complicated, mainly when the graph is cyclic, and the transitions from one node to another are conditioned. To solve this problem, we propose to extend the memory-based Neural Turing Machine (NTM) with two novel additions. We allow for transitions between nodes to be influenced by information received from external environments, and we let the NTM learn the context of those transitions. We refer to this extension as the Conditional Neural Turing Machine (CNTM).**

We show that the CNTM can infer conditional transitions in graphs by empirically verifying the model on two data sets: a broad set of randomly generated graphs, and a case study graph modeling the information retrieval process during specific crises. The results show that the CNTM can reproduce the paths inside the graph with accuracy ranging from 82.12% for ten nodes graphs to 65.25% for hundred nodes graphs..

Keywords—Memory based neural network , Graph modeling, Link prediction, Crisis management.

I. Introduction

Many important machine learning tasks involve data modeled as graphs such as classification and analysis of parse trees, social networks, knowledge graphs, transportation systems, and molecular structures. Those tasks typically involve learning the graph structure, including the relationship between the nodes, often based on partial graph observations. An example of partial observation is a family tree showing, for example, that Davis is John's father and Alice John's sister. The learning algorithm need then to infer that David is Alice's father. Leaning such relations is a challenging task, mainly when the graph is cyclic, transitions from one node to another are conditioned, and the observable data does not contain all the edges of the graph.

Over the years, several machine learning approaches have been introduced to model graph data ranging from the simple Bayesian networks [1] to recurrent neural networks (RNN) [2], and their more recent memory augmented versions: the Neural Turing Machine(NTM) [3] and Deferential Neural Computers(DNC) [4]. RNNs have been used to learn functions over sequences for more than three decades. The recent development of RNNs including the sequence-to-sequence paradigm [5], GTP-2 [6], content-based attention mechanism [7], and pointer networks [8], have gone a long way into solving significant challenges in sequence learning. Further, the NTM introduced an interaction between the network and an external memory which made it possible for RNN to be applied in new domains such as learning functions over trees, or graphs.

Although impressive results shown by the RNN applied to learn family trees, sparse trees for natural language processing, and transportation systems, its application on network and graph data is still limited to simple cases. In this paper, we are interested in conditional graphs. In such graphs, external input conditions the transition from a node to another.

A real-world analogy to better understand conditional graphs is a model of the thought process of a person. Let us assume that the person is hungry. In our simple example, many states can follow, but we narrow it down two possibilities. The first possibility is that he sits down for lunch. The other possibility is that he instead only has a small snack. The possible states are then either “eating lunch” or “eating a snack”. Whether he goes to any of those states is conditioned. It depends on many aspects, much of which he does not have control over, such as the time of day and his hunger level. In this case, the person undergoes a conditional transition from hungry to both “eating lunch” and “eating a snack”.

Another example which we take as a case study in this paper is the information gathering process during a crisis. A crisis is a complex event in which many variables change over time. The information needed by crisis responders largely varies from a crisis context to another. Furthermore, a typical situation is that any new information provided makes the responders require even more information, e.g., receiving information that a fire has broken out leads to the needed information of where the fire is located. Such information gathering process can be modeled as a graph which directly influences the decisions and interventions to take depending on the status of the crisis.

Hence, the information gathering process depends on the status of the crisis, and the information gathered so far. One might argue that such a graph can be represented using a simple finite state machine (FSM) in which each state represent the

needed information, and the inputs are the statue of the crisis. However, the number of crisis types ranging from natural, human-made to technological and this number is continuously growing, the dynamic and evolving nature of each crisis all are factors that make the FSM designed to model the information graph infinitely big and exhausting to maintain and update. Nevertheless, if we assume that we have an FSM that represents the information graph of specific crises, the question becomes: can that generalize to other crises and other situation in a specific crisis? In this case, the problem becomes that of link prediction in the sense of inferring missing links from an observed FSM graph. As example, if we know that in a state A for crisis C we require information I (I in the next state in the FSM), then, in the same state A of a similar crisis $C^{'}$, it is highly likely that we require the same information I. To be more concrete, if there is a fire (crises C) and we are in a situation where we do not know the location (we are in state A), we require information about the location (information I). If, on the other hand, there is a shooting (crises $C^{'}$), and do not know the location (state A), we also need the location (information I).

In this paper, we propose an extension of the memory-based neural Turing machine to model conditional transition graphs; we call it the Conditional Neural Turing Machine (CNTM). The aim is to allow the CNTM to change state, infer missing links in a conditional transition graph, and transit from a node to another based on input received from an external environment. First, to prove the concept, we test our model on a set of randomly generated conditional transition graphs. Then, to practically test our approach, we consider the use case of a humanitarian crisis. We show how the iterative information gathering process during a crisis can be modeled in a conditional graph, and we use that graph to test our proposed model.

II. Background

II.1 State of the art

During the first years of artificial intelligence (AI), neural networks were considered an unpromising research direction. From the 1950s to the late 1980s, AI was dominated by symbolic approaches that attempted to explain how the human brain might function in terms of symbols, structures, and rules that could manipulate said symbols and structures [9]. It was considered by many that the brain function could be implemented using a Turing machine. It was not until 1986 thanks to the work of Hinton that neural networks or the more commonly used term connectionism regained traction by exhibiting the ability for the distributed representation of concepts [10].

Despite this new capability, two significant criticisms were made against neural networks as tools capable of implementing intelligence. First, neural networks with

fixed-size inputs were seemingly unable to solve problems with variable-size inputs like words and sentences. Second, neural networks seemed unable to do a symbol level representation i.e., to represent a state that has a combination of syntactic and semantic structure such as language.

The first challenge was answered with the creation of advancement in RNNs, in particular, LSTM and GRU [11] [12]. RNNs can now process variable-size inputs without needing to be constrained by a fixed frame rate. This advancement brought breakthrough and state-of-the-art results in core problems such as translation, parsing, and video captioning.

The second criticism (i.e., missing symbol level representation) is still a pending issue. However, attempts to solve that problem started from the early 1990s. In 1990, Touretzky designed BlotzCONS [13], a neural network model capable of creating and manipulating composite symbols structures (implemented using a linked list). BlotzCONS shows that a neural network can exhibit compositionality, and reference a complex structure via abbreviated tags -two properties that distinguish symbol processing from a low-level cognitive function such as pattern recognition. Later, Smolensky continued by defining a general neural network method capable of value/variable bindings [14]. The methods permit a fully distributed representation of bindings and symbolic structures. At the same time, Pollak [15] designed a neural network architecture capable of automatically develop a distributed representation of compositional recursive data structure such lists and trees. In 1997, Hochreiter et al. [12] developed the Long Short-Term Memory network (LSTM) mainly to solve the exploding/degeneration gradient problem, but the network also exhibits memory like features such as copy and forget. In the early 2000s, Plate [16] worked on the same problem of distributed representation of compositional structures by using convolutions to associate items of these structures represented by vectors. Graves et al. [3] developed the neural Turing machine by giving a neural network an external memory and the capacity to learn how to access it, read from it and writes to it. The NTM reconciles the connectionist approach and the symbolic approach with the idea that brain functions can be implemented using a Turing machine. Several extension of the NTM was developed over the past few years most notability the sparse NTM [17] and the DNC [4].

In this paper, we extend the NTM to learn partially observed graphs. The link prediction problem is related to inferring missing links form an observed network or graph. It is based on constructing a network of observable data and try to infer additional links that, while not present in the observed data, are likely to exist. For a graph $G = (V, E)$ where V is the set of nodes, and E is the set of edges, the prob-

ability of choosing correctly at random an edge in a sparse graph (which is the case in most applications domain) is $O(1/V^2)$: As the graph grows bigger, the problem link prediction problem becomes more difficult . The link prediction problem is a common problem in social networks where the objective is to predict if two people are likely to connect (the friend suggestion feature on Facebook, for example) [18]. Beyond social networks, link prediction have applications in bioinformatics [19], e-commerce [20], and security [21]. Different approaches have been used for that purpose [22]: First, the non-Bayesian approach which trains a binary classification model on a set of extracted features. Second, the probabilistic approach which models the joint-probability among the entities in a network using Bayesian models. Finally, the linear algebraic approach computes the similarity between the nodes in a network using similarity matrices.

All of the previously cited link prediction applications do not consider the case in which the edges are conditioned by an external input: so-called conditional graphs. A typical example of a conditional graph is the graph represented by an FSM, An FSM has a structure that exhibits a syntactic and semantic meaning, which often is cyclic, and with transitions between nodes dependent on external input. On the other hand, if we only have an FSM that only represents a part of the system, and we want to complete this FSM by inferring new links making it fully descriptive of the system, then the problem becomes challenging to model using traditional link prediction solution because it introduces a new variable which is the external input. A typical example is a graph where some links are missing or not known which such as in crisis information retrieval problems introduced earlier. However, an FSM can be represented by a Turing machine. We use this feature to design a neural Turing machine that can infer the kind of link present in an FSM.

II.2 Neural Turing Machine

An NTM is composed of a neural network, called the controller, and a two-dimensional matrix often referred to as the memory (Figure C.1). The controller is a feed-forward or recurrent neural network that can read from and write to selected memory locations using read and write heads. Graves et al. [3] draw inspiration from the traditional Turing machine and use the term head to describe the vector the controller uses to access the selected memory location. The read head $w^r(t)$ and the write head $w^w(t)$ have the property described in equation C.1.

$$\sum_i w_i^r(t) = \sum_i w_i^w(t) = 1. \tag{C.1}$$

Let $M(t)$ be the $n \times m$ memory matrix at time t. To read values from M, we need an addressing mechanism that dictates from where the head should read. A

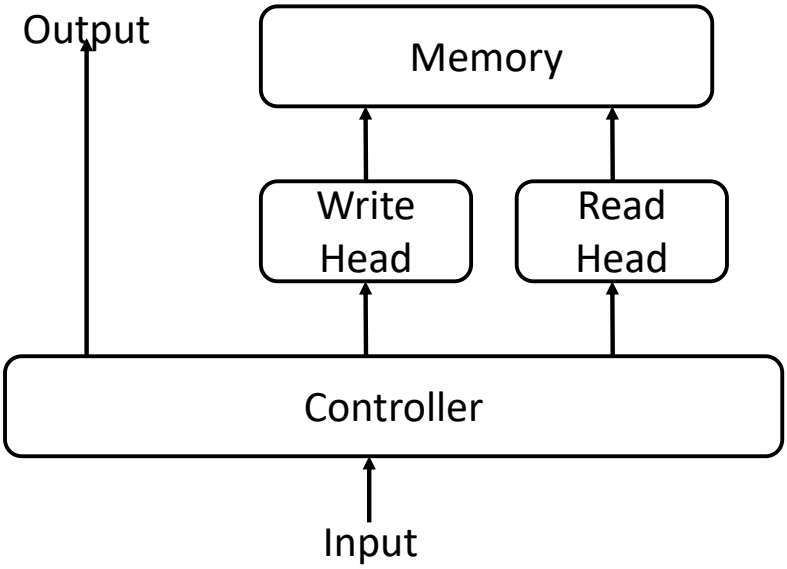


Figure C.1: An NTM block

read operation is defined as the weighted sum over the memory rows $M_i(t)$:

$$r(t) = M(t)^T w^r(t). \tag{C.2}$$

The writing operation is composed of an erase operation, and an add operation. The erase operation deletes certain elements from the memory $M(t-1)$ using an erase vector $e(t) \in [0,1]^m$. The add operation replaces the deleted values with elements from an add vector $a(t)$. Thus, the writing operation can be expressed by the following equation where $\circ$ is the element-wise multiplication:

$$M(t) = M(t-1) \circ [1 - w^w(t)e(t)^T] + w^w(t)a(t)^T. \tag{C.3}$$

The calculations of the vectors $w^r(t)$ and $w^w(t)$ is done independently but using the same approach. Thus, in the remaining of this section $w(t)$ will denote $w^r(t)$ or $w^w(t)$ interchangeably.

There are two types of addressing methods used to create the vector $w(t)$: content-based and location-based addressing. First, the content-based addressing selects the weights based on the similarity between a row in the memory matrix and a given query $k(t)$ generated by the controller:

$$w(t) = w^c(t) = \frac{f(\beta(t)d(k(t), M_i(t)))}{\sum_j f(\beta(t)d(k(t), M_j(t)))} \tag{C.4}$$

where d is a similarity measure (typically cosine similarity), f a differentiable

monotonic transformation (typically a softmax), and $\beta(t) > 0$ a key strength that amplifies or attenuate the precision of the focus.

Second, location-based addressing goes through three different phases:

1. An interpolation between the previous weights $w(t-1)$ an the wights produced by the content based addressing using a gate $g(t) \in [0,1]$ (equation C.5). This method is used when we want to have a combination of content based and location based addressing. It yields the weight $w^g(t)$

$$w^g(t) = g(t)w^c(t) + (1 - g(t))w(t-1) \tag{C.5}$$

2. A shift operation that rotates the elements of the weights using a shift vector $s(t) \in [0,1]^n$ (equation C.6). The shift produces the weights $w^s(t)$.

$$w_i^s(t) = \sum_j w_j^g(t) s_{i-j}(t) \tag{C.6}$$

3. A sharpening that combats any leakage or dispersion of weights over time if the element of the shift vector s(t) are not sharp i.e. neither close to 1 or 0.

$$w_i(t) = \frac{w_i^s(t)^{\gamma(t)}}{\sum_j w_j^s(t)^{\gamma(t)}} \tag{C.7}$$

All the parameters $\beta(t), k(t), g(t), s(t), and \gamma(t)$ used to compute $w(t)$ are calculated using neural layers that takes as input the output of the controller $h(t)$ at time t . Given the constraint applied to some, we use different activation functions to compute these parameters: Rectifier linear for $\beta(t)$, Sigmoid for $g(t)$, Softmax for $s(t)$, and Oneplus for $\gamma(t)$.

III. Theoretical approach

III.1 Problem definition

In a conditional transition graph, external knowledge conditions the transitions from one node to another. Figure C.2 shows a simple example of such a graph: The transition from node A to D is performed when the proposition C is true. A conditional transition graph is composed of:

- A finite set Q of node or states
- A finite set C of input. C can be a set of logical proposition p_i that can be true or false as presented in Figure C.2, or a vector of logical propositions. In the context of this paper, C is a set of variables $c_i \in [0,1]^n$.

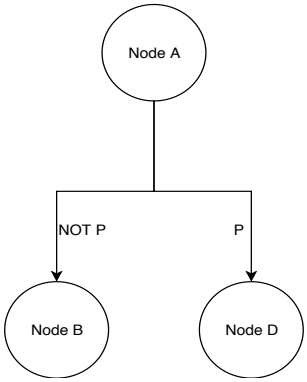


Figure C.2: Example of a simple conditional graph

- A transition $\delta : Q \times C \longrightarrow Q$ from a node to the next.
- A final node of state F

In this paper, we model two essential parts of a conditional transition graph: The first part is the input C that triggers that transition. The second part is the transition δ.

To produce C, we introduce what we call an environment. The environment's role is to produce an input given the current node in the graph. As an example, a node in the graph can represent a database query. The environment, in this case, is the database that, given the query, returns a set of data (the condition C). That data is then used to select the next node or query in this case. It is worth noting that the environment can be any simple or complex system, such a deterministic, or a real-world system (e.g., database). In this paper, we consider the environment be random: Consider that from node N_0, we can transition to nodes N_1, N_2, ..., or N_n. Each transition is conditioned with c_1 c_2,..., and c_n respectively. A random environment E gives a probability distribution over all the possible value of C: $P(c_i|N_0); i = 1, .., n$.

The problem of learning the transition δ can be expressed as learning a conditional probability distribution over the set of sates Q knowing the current state and the input form the environment:

$$\delta(y, c) = \arg\max_{y_i \in Q} P(y_i|y, c); y \in Q, c \in C. \tag{C.8}$$

In the next section, we detail how the CNTM learns such a probability.

III.2 Neural Turing Machine for conditional graphs

This section extends the existing NTM to be able to predict links in a conditional graph. We call this extension the conditional neural Turing machine (CNTM),

which is the contributions of the paper.

In Section III.1, we introduced the environment which randomly produces an input $c \in C$. The input produced by the environment can be extended to include the current node in the graph $x(t)$. This extension produces what we call a context vector $v = [x(t), c] \in Q \times C$ which is the input of the transition δ.

The first step of the CNTM is to produce a coding U given the current context v and the sequence of previous contexts. The idea is to use the NTM attention mechanisms (content-based and location-based addressing) to retrieve a representation of the context. Using these attention mechanisms, the vector $r(t)$ produced form the memory presents the closest vector to v.

The coding U is, then, implemented using a neural layer that takes as input the output of the controller $h(t)$ and the read vector $r(t)$ and calculates a linear combination between them. Thus the activation function for that output layer is a linear activation:

$$U = W_1 * h(t) + W_2 r(t) + b. \tag{C.9}$$

In the second step, the transition δ form a node to the other is implemented using an output layer. The output layer's role is to produce the next node in the graph $x(t+1)$ given the previous set of codings of the context U produced by the NTM block. At each time step t, the output layer takes as input U. Its output at time t is a probability distribution over the nodes of the graph $P(y|U, \beta)$, where β is the parameters of the output layer. It is implemented using an LSTM with a Softmax output layer.

The training phase of the CNTM is divided into two phases: A description phase, and an answer phase. During the description phase, the input (v) is presented to the CNTM in random order. During the answer phase, the CNTM is to produce the target node. For a sequence of contexts v and a sequence of targets y both of length T, the parameters of the model are trained to maximize the cross-entropy loss function:

$$L(x, y) = -\sum_{t=1}^{T} A(t) log(P(y_t|v_t)) \tag{C.10}$$

Where $A(t)$ is an indicator function whose value is 1 during answer phases and 0 otherwise. The overall model is presented in Figure C.3. The CNTM is differentiable from end to end, and its parameters can be optimized using stochastic gradient descent, or other standard neural network optimizers.

IV. Experimental results

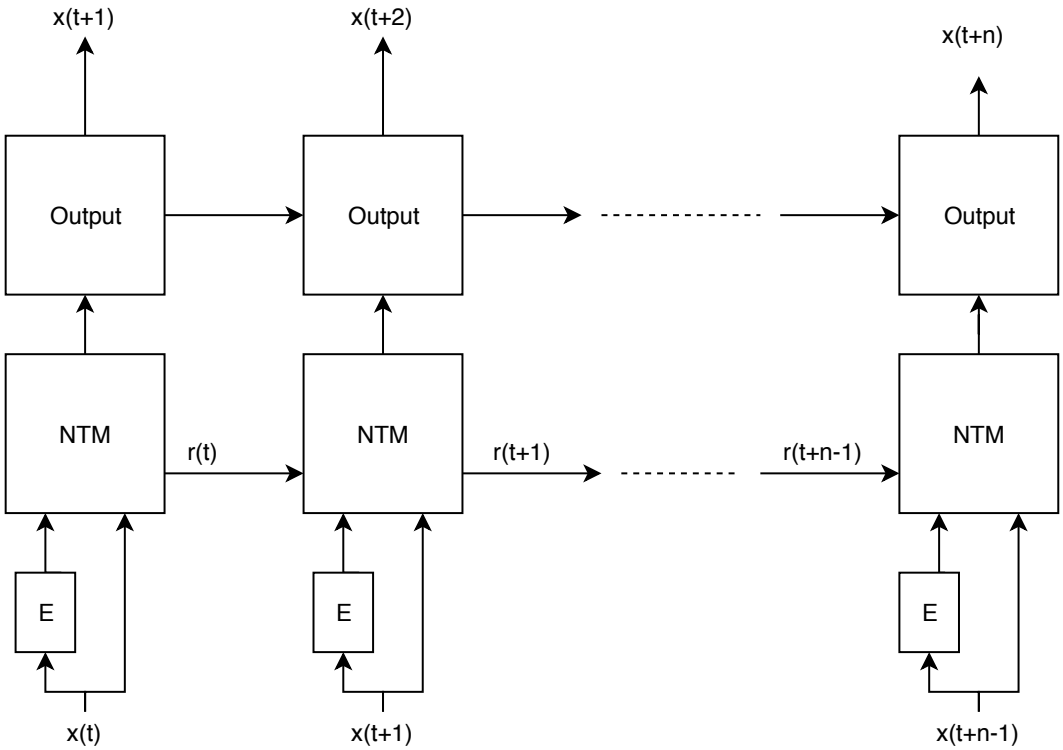


Figure C.3: Neural network for conditional graph modeling (CNTM)

In the conditional graph inference task, the input of the CNTM consists of a tuple encoding the current state, the input from the environment, and the target state. Each element is coded using a binary vector with a vector of all zeros reserved for undefined elements. We set the length of the vector to 30, so the input of the CNTM is a 60 elements vector.

The experiment CNTM works in two phases, training and validation. The training is performed, as explained in the previous Section. For evaluation, the first input to the network is the initial node with the input from the environment. In the rest of the time steps, the input tuple contains only the input from the environment. To succeed, the network had to infer the destination of each tuple, and remember it as the implicit current state for the next time step. We assume that the output of the CNTM is correct if the produced path is a sub-path of the complete graph.

For the output, we used an LSTM with 256 hidden units, a feed-forward network for the NTM controller of 128 units, a memory of 128×128. All the weights and the memory were initialized using a Xavier initialization. The CNTM is trained with RMSprop stochastic gradient descent with a learning rate of 0.001 a batch size of 128.[4]

IV.1 Random graphs

Table C.1: Accuracy on the randomly generated graphs and the crisis data

Data	CNTM	LSTM	Graph distance
Randomly generated graphs with 10 nodes	82.12%	79.51%	19.45%
Randomly generated graphs with 20 nodes	78.54%	70.23%	18.03%
Randomly generated graphs with 40 nodes	72.62%	62.46%	15.78%
Randomly generated graphs with 60 nodes	70.78%	57.93%	13.59%
Randomly generated graphs with 80 nodes	67.61%	50.89%	10.30%
Randomly generated graphs with 100 nodes	65.25%	42.47%	6.34%
Crisis data: 50 nodes	78,59%	67.29%	16.46%

We train and test out model with two datasets. The first dataset is used to prove the concept and is composed of randomly generated sparse conditional graphs. Here we compiled 6 different datasets. Each dataset contains 1000 different conditional graphs of 10, 20, 40, 60, 80, 100 nodes each.

It is important to note here that during the training phase, we only train the algorithm on graphs containing 70% of the links in the randomly generated graphs. Table C.1 shows the accuracy of the CNTM compared to the vanilla Graph distance [23], and the LSTM [12] in inferring the correct links for randomly generated conditional transition graphs. The table shows a clear advantage of using the CNTM over the other approaches. As can be expected, the bigger the graph (in the number of nodes), the less accurate the predictions become. For a graph with 100 nodes, the accuracy is 65.25%. However, as the number of nodes grows, the gap in performance between the CNTM and the other approaches grows exponentially: The gap between the CNTM and the LSTM starts with 2.6% for 10 nodes graph, and it grows to reach approximately 23% for 100 nodes graphs. Note if we randomly pick a 10 nodes-long path from the same graph the change of getting a correct pick is approximately 10^{-18}.

Figures C.4 to C.6 show box-plots comparing the result produced by the CNTM with three other approaches on all the randomly generated context graphs: a random predictor, graph distance, and LSTM respectively. Figure C.4 compare the CNTM, LSTM, and graph distance with a random predictor as a baseline. It shows that all these approaches perform at least 10% better on average than the random predictor.

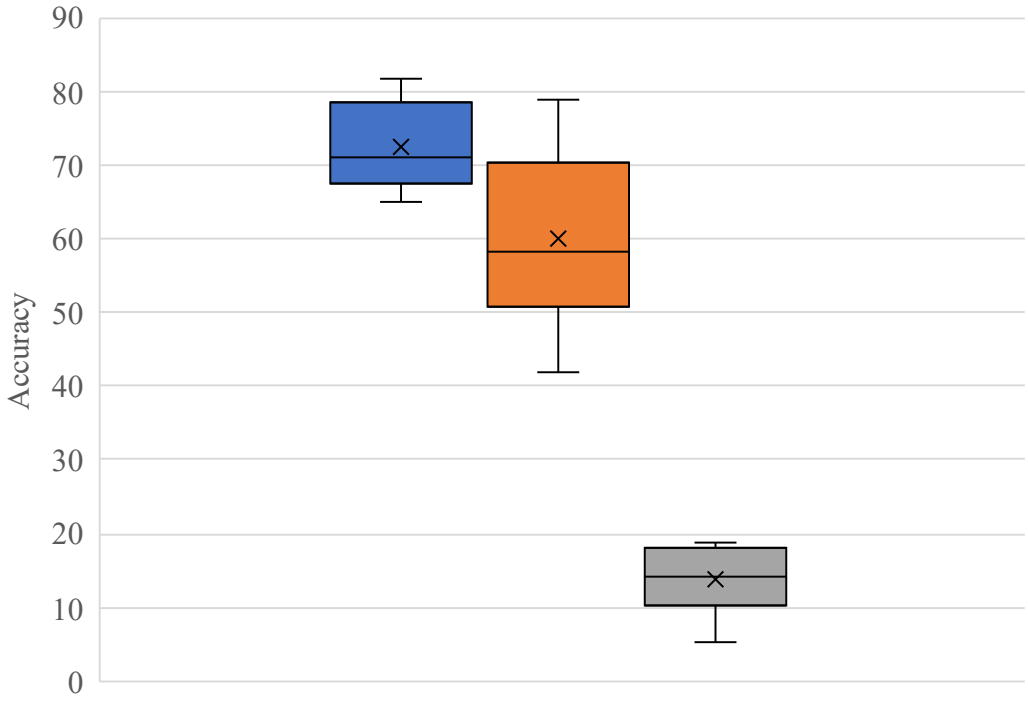


Figure C.4: Comparison of different link predictor with the random predictor as the baseline.

The CNTM is on average, approximately 70% better than a random predictor. Figure C.5 compares the CNTM and the LSTM with the graph distance as a baseline. It illustrates that both these approaches perform on average 42% better than the graph distance. Finally, Figure C.6 uses the LSTM as a baseline. It shows that the CNTM performs 10% better than the LSTM on average. It is important to point here that the variance in performance of the CNTM is much lower than the other approaches.

IV.2 Case study

The second data set is a much more use case to model the information needs by emergency management services during a crisis (Figure C.7 presents a portion of that graph). Emergency management is chosen to prove the practical applicability of the CNTM since this is a particularly challenging scenario. Emergency personnel relies on correct information in dynamic and chaotic situations. Further, the graph is highly conditioned as much of the emergency response relies upon previous information such as type of crisis and location. For example, emergency personnel needs to respond differently if a crisis is a public disturbance or a fire outbreak. The type of response is conditioned on the type of crises. Besides, emergency management services are well documented in the literature.

The environment graph is compiled using the information available in the lit-

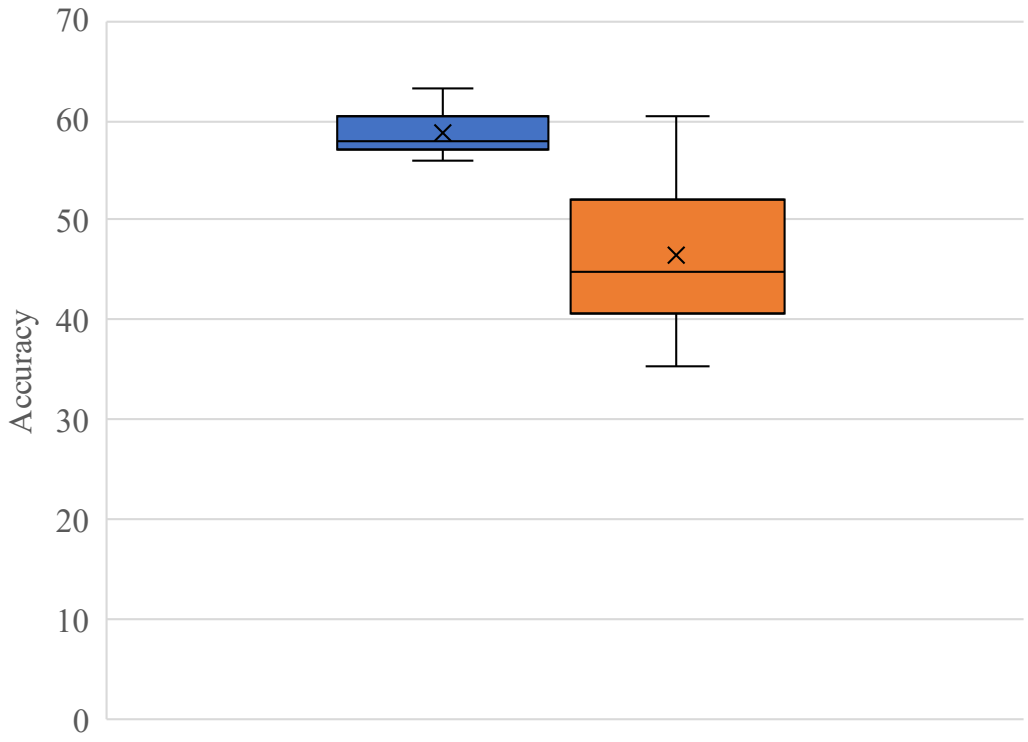


Figure C.5: Comparison of different link predictor with the graph distance predictor as the baseline.

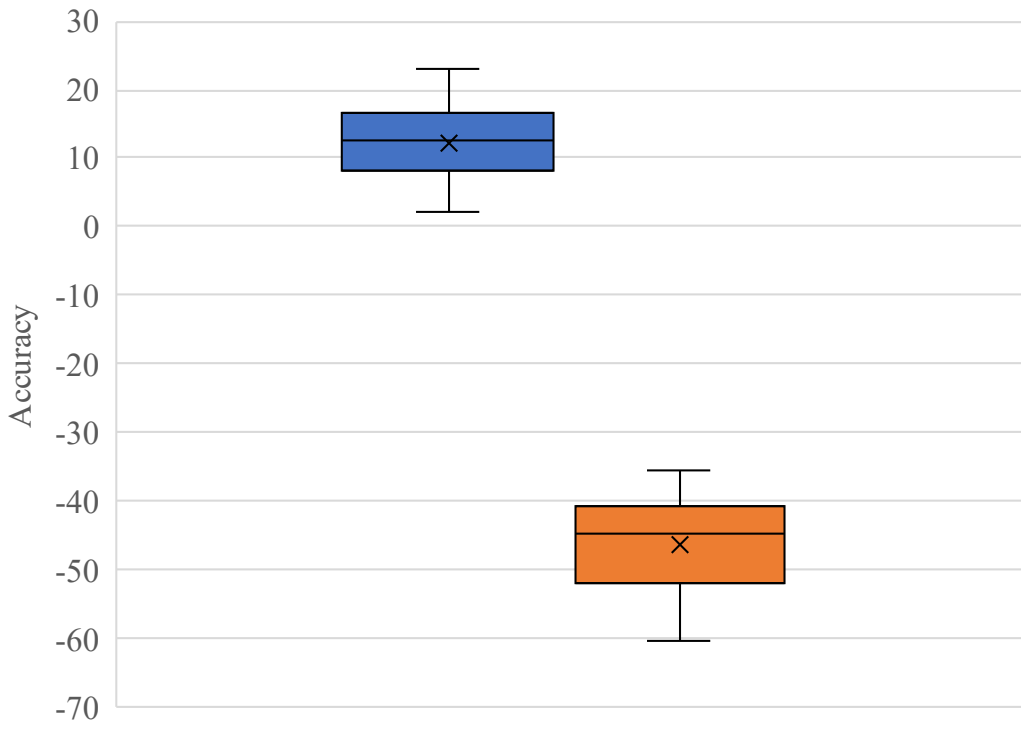


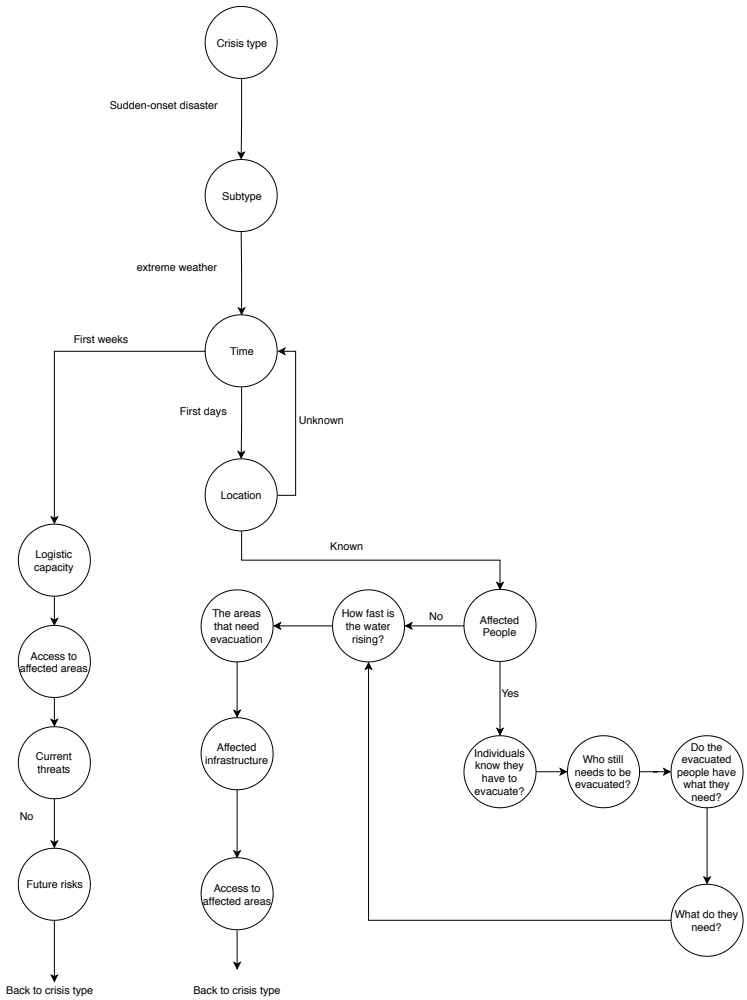
Figure C.6: Comparison of different link predictor with the LSTM predictor as the baseline.

erature, particularly in three areas: Fire, extreme weather, and public disturbance. Emergency personnel acts differently in these three scenarios. For a fire emergency, the sub-graph is extracted from the work of Nunvath et al. [24] who did an extensive interview of firefighters about the type of information they need during an indoor fire crisis. For extreme weather, the sub-graph was extracted from the work of Ben Lazreg et al. [25] who collected personnel form police and municipality to gather the type and flow of information they need during an extreme weather crisis. Finally, The public disturbance sub-graph, as well as the rest of the graph, was vetted by two policemen from Oslo police station who are expert in riots, demonstrations and public disturbance control. The nodes in the graph present the type of information needed by emergency manger during a crisis. The transition from node 1 to node 2 is conditioned on whether the information that node 1 requires is answered or not. The environment provides the answers.

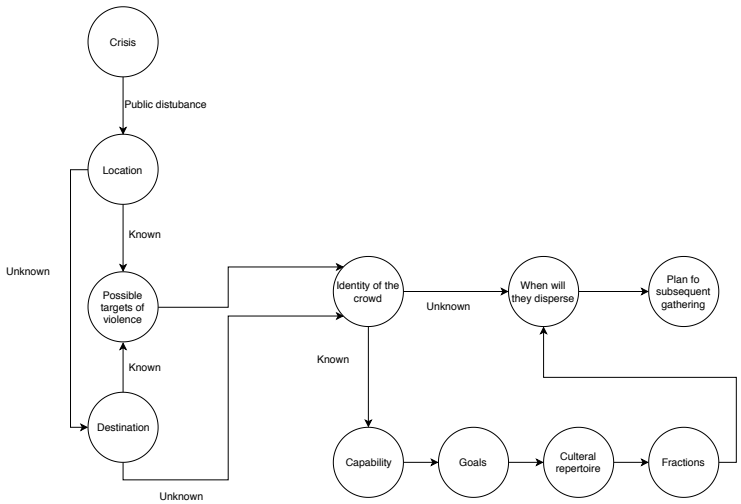
Similarly to the randomly generated graph, we only train the algorithm on graphs containing 70% of the links in the crisis graphs. The accuracy of the network in inferring the correct links for the crisis graphs is 78,59%. It is in the same range of the accuracy obtained using a randomly generated graph of 20 nodes. This increase in accuracy might be because, in randomly generated graphs, we average the results over 100 different graphs. Some of the graphs might perform worst or better than the average depending on randomly generated edges. The crisis graph, on the other hand, is a well-defined graph presenting logical edges and connections.

Figure C.8 shows an example of paths in the graph proposed by the network from the crisis information graph. In this test, the environment is given externally by what is the correct and wrong transitions in the emergency graph. Note that the third path in the figure contains a link not available in the full graph, therefore classified as a wrong. It proposes a transition from location to affected people. The link from the location node in the context of indoor fire is not available in the training data. However, a transition from location to affected peoples is present in the training data in the context of extreme weather. Since both extreme weather and indoor fire are sudden-onset disasters, the CNTM was able to predict a link between the location and the affected people nodes in the context of indoor fire.

We tried to investigate further the links predicted by the CNTM. We designed six crisis scenarios, and we run the CNTM to provide the required information in those contexts. For each scenario, we gave a graded of 4 to the information predicted by the CNTM and a grade of 1 the other information. Let us note these grades by E_{CNTM}. We proposed the same scenarios to four emergency management services (EMSs) experts (two Red Cross (RC) personnel, a police officer, and a



(a) Example of an extreme weather information graph



(b) Example of a public disturbance information graph

Figure C.7: Conditional graph for for information needed by crisis emergency management

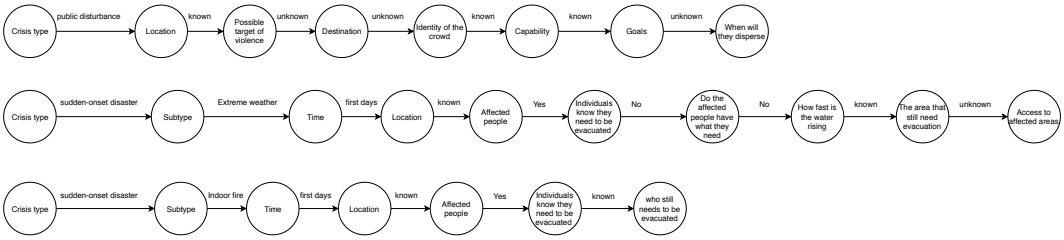


Figure C.8: Example of results provided by the model

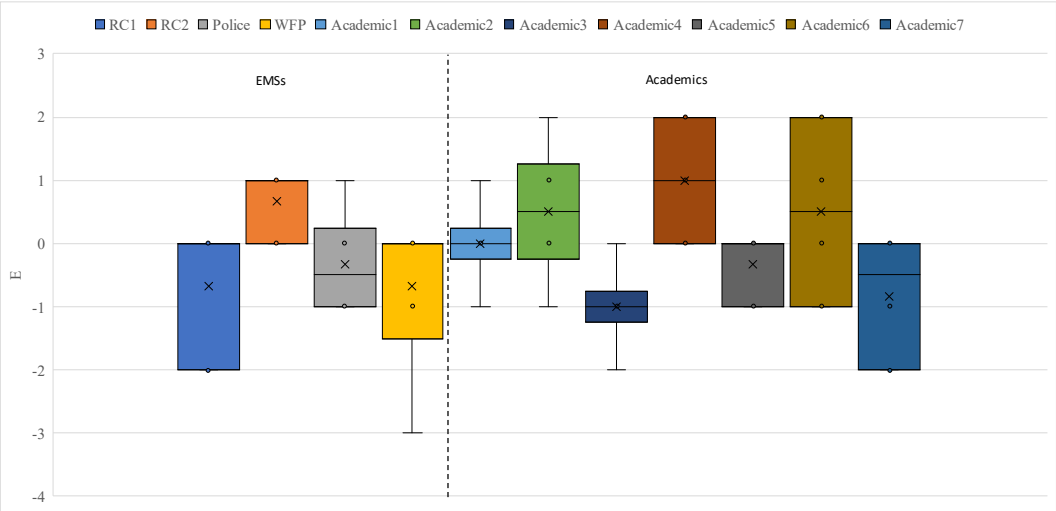


Figure C.9: Comparison of expert opinion with the CNTM predictor.

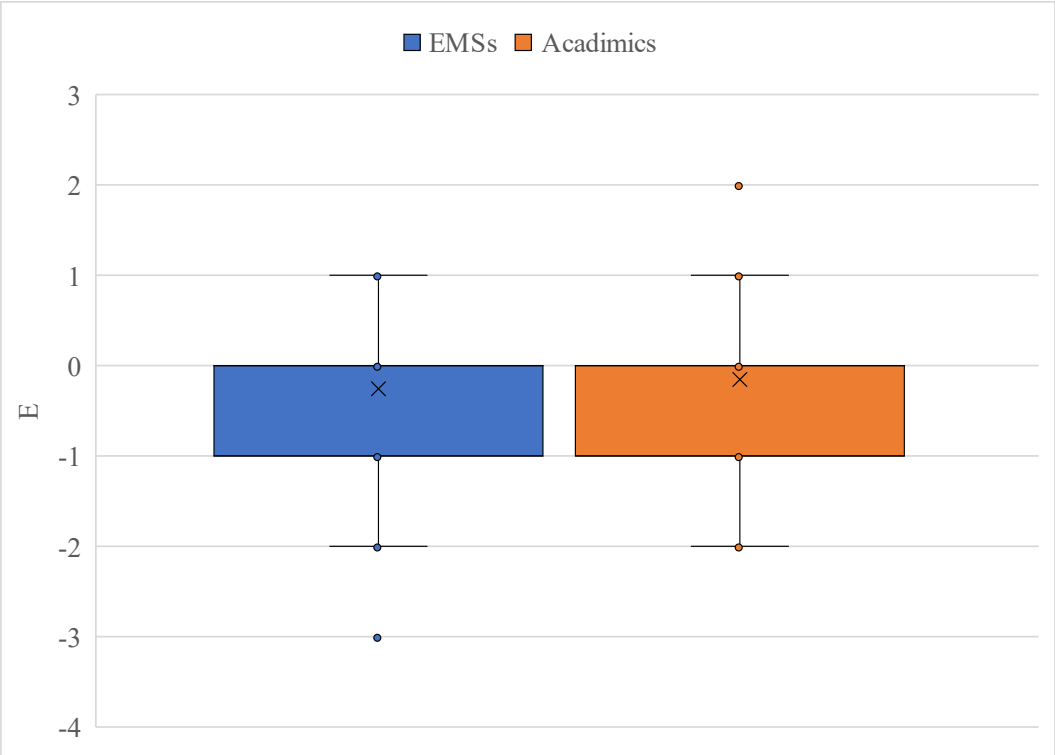


Figure C.10: Comparison of EMSs and academics expert opinions with the CNTM predictor.

World Food Program (WFP) field worker) and seven academic experts in emergency management. We asked them to rank on a scale from 1 to 4 (1 being not relevant, and 4 highly relevant) four required information, including the information proposed by the CNTM and three other random information. Let us note the grade assigned by the expert by E_{expert}. Then, we study the correlation between the expert evaluation and the information predicted by the CNTM by examining the distribution of E:

$$E = E_{CNTM} - E_{expert}. \tag{C.11}$$

Figure C.9 shows a box plot of the distribution of E for each expert. The figure shows in 83.03% of the expert evaluation is within the $[-1, 1]$ interval, which means that the expert assigns a grade of 3 or 4 to the information paths predicted by the CNTM, and a grade of 1 or 2 to the other proposed information. Figure C.10 compares the EMSs evaluation with the academics expert evaluation. It shows the generally, the evaluations do not differ. However, E tends to be on the negative side, which means that in some scenarios, the experts picked information different from what the CNTM suggested as relevant. Positive values for E would mean that experts either tend to judge all the proposed information as irrelevant, or the predicted

information by the CNTM not highly relevant and the other three as irrelevant.

V. Conclusion

This paper presents a neural network able to predict links in conditional graphs. The proposed network, the Conditional Neural Turing Computer (CNTM), extends the Neural Turing Computer with the capability to manage external conditions.

A conditional graph is a graph in which an external input conditions the transition from a node to the other. We showed that such graphs could be divided into two parts: an environment and a transition. In our experiments, the environment is modeled as a random generator of inputs. To present the transition, we used the CNTM carried out empirical tests on two data sets. The first was a broad set of randomly generated conditional graphs followed by a graph modeling the information retrieval process during specific crises. The results showed that the CNTM could reproduce the paths inside the graph with accuracy ranging from 82.12% for 10 nodes graphs to 65.25% for 100 nodes graphs.

Paper C: References

[1] R. Gupta and V. C. Pedro, "Knowledge representation and bayesian inference for response to situations," in *AAAI 2005 Workshop on Link Analysis*, 2005.

[2] S. Ahn, H. Choi, T. Pärnamaa, and Y. Bengio, "A neural knowledge language model," *arXiv preprint arXiv:1608.00318*, 2016.

[3] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[4] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S. G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, p. 471, 2016.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners."

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[8] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2692–2700.

[9] J. A. Fodor and Z. W. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis," *Cognition*, vol. 28, no. 1-2, pp. 3–71, 1988.

[10] G. E. Hinton *et al.*, "Learning distributed representations of concepts," in *Proceedings of the eighth annual conference of the cognitive science society*, vol. 1. Amherst, MA, 1986, p. 12.

[11] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[13] D. S. Touretzky, "Boltzcons: Dynamic symbol structures in a connectionist network," *Artificial Intelligence*, vol. 46, no. 1-2, pp. 5–46, 1990.

[14] P. Smolensky, "Tensor product variable binding and the representation of symbolic structures in connectionist systems," *Artificial intelligence*, vol. 46, no. 1-2, pp. 159–216, 1990.

[15] J. B. Pollack, "Recursive distributed representations," *Artificial Intelligence*, vol. 46, no. 1-2, pp. 77–105, 1990.

[16] T. A. Plate, "Holographic reduced representations," *IEEE Transactions on Neural networks*, vol. 6, no. 3, pp. 623–641, 1995.

[17] J. Rae, J. J. Hunt, I. Danihelka, T. Harley, A. W. Senior, G. Wayne, A. Graves, and T. Lillicrap, "Scaling memory-augmented neural networks with sparse reads and writes," in *Advances in Neural Information Processing Systems*, 2016, pp. 3621–3629.

[18] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[19] E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, and T. Jaakkola, "Mixed membership stochastic block models for relational data with application to

protein-protein interactions," in *Proceedings of the international biometrics society annual meeting*, vol. 15, 2006.

[20] S.-L. Huang, "Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods," *Electronic Commerce Research and Applications*, vol. 10, no. 4, pp. 398–407, 2011.

[21] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *SDM06: workshop on link analysis, counter-terrorism and security*, 2006.

[22] M. Al Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social network data analytics*. Springer, 2011, pp. 243–275.

[23] H. H. Song, T. W. Cho, V. Dave, Y. Zhang, and L. Qiu, "Scalable proximity estimation and link prediction in online social networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement*. ACM, 2009, pp. 322–335.

[24] V. Nunavath, A. Prinz, and T. Comes, "Identifying first responders information needs: supporting search and rescue operations for fire emergency response," *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, vol. 8, no. 1, pp. 25–46, 2016.

[25] M. Ben Lazreg, N. R. Chakraborty, S. Stieglitz, T. Potthoff, B. Ross, and T. A. Majchrzak, "Social media analysis in crisis situations: Can social media be a reliable information source for emergency management services?" 2018.

Paper D

Title:	An Iterative Information Retrieval Approach from Social Media in Crisis Situations Situations
Authors:	Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo
Affiliation:	Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
Conference:	*2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)* , Paris, France, December 2019.
Copyright ©:	IEEE

D

An Iterative Information Retrieval Approach from Social Media in Crisis Situations

Mehdi Ben Lazreg, Morten Goodwin, and Ole-Christoffer Granmo

***Abstract* — During to past few years, social media have gained a pivotal role in crisis communication. Its usage has ranged from informing the public about the status of a crisis and what precaution need to be taken, to family members checking on the safety of loved ones. Despite the widespread use of social media in crises situations and the clear potential benefit from collecting potential critical information from social media, emergency management services (EMSs) are still reluctant to use social media as a source of information to improve their situational awareness. One of the reasons for the reluctance is that crises management are typically overloaded with information. Adding social media will only increase the information overload, and the EMSs risk being provided with more and possibly irrelevant information from social media sources. Furthermore, most automated social media analysis platforms are designed exclusively to classifying messages into crisis and not crisis-related categories. The platforms do not take into account the degree of relevance of social media information to the EMSs. Such relevance further depends on the crisis' status at a certain point in time, and the information gathered up to that point.**

This paper proposes an intelligent information retrieval framework from social media during crises. The developed framework combines two components: The first component classifies social media messages into separate topics representing the information needed in different situations. The second component decides which information to retrieve based on the information available and the status of the crisis. The framework is evaluated through a survey of EMSs. They agree with the framework in 70.09% of the cases about what information is needed.

Keywords—Information retrieval, Social media, Machine learning, Crisis management.

I. Introduction

Social media has become a pivotal communication medium in most crises today. During a crisis, social media has been used mainly as three different communication channels. First, from EMS to the public, the EMSs relay situation updates,

evacuation orders, and possible dangers. Second, from the public to the public, the public uses social media to maintain contact with relatives, friends, and loved ones, and to support the community. Finally, from the public back to the EMS, the public uses social media to report problems, needs, calls for help, and provide information throughout the crisis.

The first two channels (EMS to public and public to the public) are starting to be well established: training is given to EMSs on how to effectively communicate information to the public [1] [2], and many social media platforms are providing ways for people in the affected area to report their safety status. The third type, on the other hand, is still facing many challenges. Much of the retrospective research on Twitter messages posted during events show that those messages can identify the problems appearing during the relief effort. For example, an analysis performed on tweets related to the 2015 Nepal earthquake, discovered that the most discussed topics are monetary needs, and reports on missing, injured, and dead persons [3]. Although the information shared by the public can enhance the situational awareness of the EMS, many EMSs are still skeptical about using that information.

A study by Ben Lazreg et al. [4] identified three reasons behind the skepticism of EMSs. The first reason is a lack of confidence in the information present in the platform, i.e., EMSs do not trust seemingly random social media users to deliver accurate situation update and needs by the public. The second reason is the quality of the content polished in social media such as clarity, readability, and conciseness of a message. The final reason is the inefficiency of data analysis platforms in collecting useful information. Social media data must be analyzed in ways that provide relevant answers to the question asked by the EMS during a crisis, which usually triggers new data collection steps and questions. For a social media analysis platform to be efficient in a crisis, it needs to focus on answering EMSs' questions and needs.

Finding the before-mentioned useful information from social media can accelerate disaster response. However, the task is not easy much because of information overload and the evolving nature of valuable information, i.e., a piece of useful information in a particular crisis context might be useless in another context. The analysis by Randianti et at. [3] shows that even though relevant topics are discussed, much of the information present in the discussions is often irrelevant. As an example, monetary support is one of the most discussed topics, but the majority of the messages are appealing for donations from ordinary people or organizations outside the affected areas and not actual financial needs, which are more relevant to the EMSs.

The growth of artificial intelligence and machine learning during the last few years has led to the emergence of many artificial intelligence-based analysis tools [5] [6] [7]. These tools analyze the data first by extracting as much information about as many topics related to the crisis as possible. They are used as mitigation to the information overload. However, when the focus is on the data, and how accurately it can be classified, rigor in collecting relevant data for EMSs becomes an afterthought. Moreover, many data scientists are relatively new to the field of social media in crisis research. They are knowledgeable about the management and analysis of large-volume data but lack the understanding of the EMSs' needs [8]. Thus, the need for an information extraction mechanism, that given the status and context of the crisis, provides the most relevant information to the EMSs.

This paper tries to provide a solution for the need for useful information retrieval from social media in a crisis. It proposes a framework based on two main modules. Both modules were part of a previous stage in this research, and we dedicated a paper on each. The first module is a link prediction algorithm that finds connections between nodes in a graph in which the edged are weighted be a specific condition which makes the transition between nodes conditioned by an input from an external environment [9]. We tune this link prediction algorithm in this paper to be a what we call a query generator. It generates a so-called "information need query" (information needed by the EMS) based on the context of the crisis and the information available so far. The second module takes that query and fetches the information related to it for social media [10]. It returns its results to the first module, which in turn generates another query based on the newly available information and so on.

The solution as a whole becomes an iterative information retrieval framework from social media. This paper also concentrates on the integration of both modules and the evaluation of the framework. The article is structured as follows: Section II gives an overview of the state-of-the-art in social media analysis. Section III provides a description of the separate module and the way they are integrated. Section IV evaluates the framework by judging its results in front of an expert opinion from different EMSs.

II. Information retrieval from social media during a crisis situation

There is no doubt that valuable, high throughput data is produced on social media only seconds after a crisis occurs [11]. To cope with the complexity of the social media data, and extract information from crisis-related messages, automated social media analysis platforms have become valuable [7].

Table D.1: Social media analysis platforms in the academic literature

Analysis platform	**Data**	**Dimension of categorization**	**Machine learning approach**	**Reference**
AIDR	Twitter	Information provided	Random forest	[5]
CrisisTracker	Twitter	Information source and Information provided	LSH	[12]
TweetCred	Twitter	Credibility of the information	Ranking SVM	[16]
Twitris	Twitter	Factual, subjective, or emotional content	Lexicon-based classification	[17]
EMERSE	Twitter, SMS	Information provided	SVM	[14]
ESA	Twitter	Information provided	Naïve Bayes and SVM	[15]
Tweedr	Twitter	Information provided	Logistic regression	[13]

CrisisTracker [12], for example, is a crowdsourcing platform used during a crisis by registered users to report needs and risks. CrisisTracker has additional functionality to cluster the collected messages onto different discussion topics using LSH. AIDR [5] uses random forest to classify social media posts into a set of predefined categories. For the same purpose Tweedr [13] uses Logistic regression, EMRSE [14] uses SVM, and ESA [15] uses a combination of Naïve Bayes and SVM. However, in addition to the classification, Tweedr used conditional random fields to extract text token from the social media message that can be related to numerical value, building, and transportation. ESA provides a keyword burst detector that generates alerts when a word starts to be used in high frequency. TweetCred [16] uses Ranking SVM to classify social media posts by credibility. Twitris [17] is more of a sentiment analysis platform that uses lexicon-based classification to classify social media posts into factual, subjective, or emotional. Table D.1 summarizes the most notable social media analysis platforms dedicated to crisis management found in the scientific literature.

All of the research on Twitter data during a crisis described previously follows a data-driven approach that analyzes the data first by finding a way to extract as much

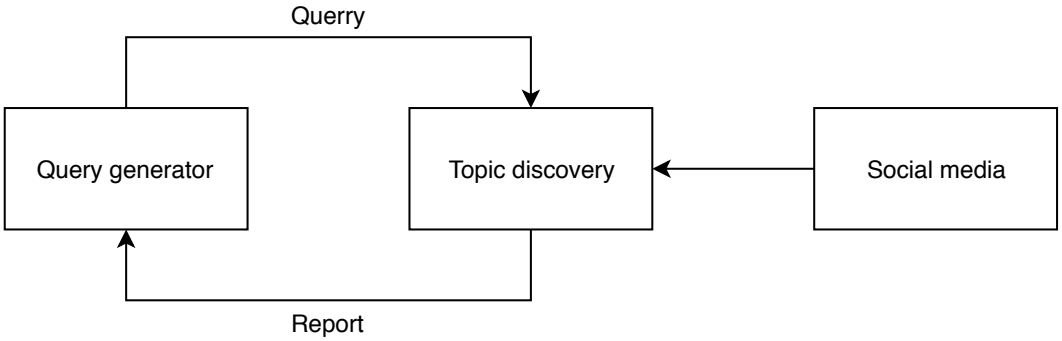


Figure D.1: General illustration of the framework

information related to the crisis as possible. Then, the EMSs' needs are addressed later. Such an approach may result in an excess of irrelevant information extracted during the analysis. In the next session, we propose a framework that tries to remedy this problem.

III. Iterative information retrieval framework

In this section, we introduce the iterative information retrieval framework, which is composed of two major components (Figure D.1). The first component is the query generator (QG). It acts as the "brain" of the framework and decides what type of information is needed based on the current status of the crisis and the information gathered so far. The second component is the topic discovery (TD). It is a classifier that classifies the social media message into the different information needs topics and returns the results to the query generator. In the following section, we will detail each component.

III.1 Topic detection

The topic detection main task is to classify each social media message into the different information need topics that EMSs are interested in during a crisis. To do so, it detects a specific topic related patterns in textual data. The topic detection is composed of three sub-components (see Figure D.2):

- A named entity recognizer (NER) which extracts metadata from the text it receives. The information is the location, number of injured, killed, and missing persons.
- A word embedding that transforms the text into a vector.
- A neural-network-based detection layer that combines the word embedding and the information extracted by the NER to produce the probability distribution over topics.

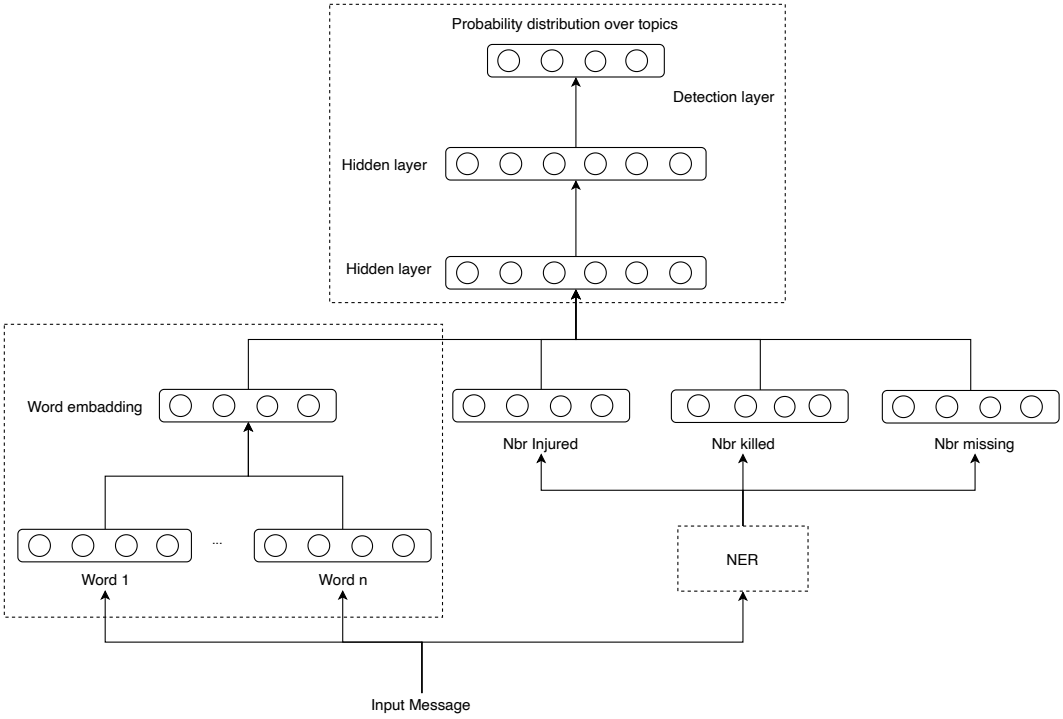


Figure D.2: Neural network model for topic discovery

III.1.1 Named entity recognizer

The NER uses a maximum entropy classifier developed by [18], a statistically-based classifier, to identify named entities by looking at each word in a sentence and deciding whether it is the start of a named entity, the continuation of an already started entity, or not part of any known named entity. By combining these probabilities, the classifier can extract the sequence of words that make up a predefined name.

In short, the model is based on the set of statistics found in the training sample. For example, the model might find that there is statistical evidence in the training samples that the number of injured appears before the word "injured" in the message. This statistical evidence is included in the estimation of the probabilities.

The model works as follows. Let Y be the set of all the words in the given text. The task is to find which set of words $y = \{y_1, .., y_n\} \in Y$ is a named entity given the set of surrounding words $x = \{x_1, ..., x_m\}$ (x is called the context). The goal is then to find the probability that y is a named entity given the context x i.e. $p(y|x)$. The probability distribution p is chosen in a way that maximizes the conditional cross-entropy over the training corpus:

$$p = \arg\max_{p} - \sum_{x,y} p(x)p(y|x)log(p(y|x)). \tag{D.1}$$

If more than one probability distributions maximizes the cross entropy, we select the most uniform distribution (following the principals of insufficient reasoning).

III.1.2 Word embadding

Social media text often uses non-standardized languages, which means that the same phrase – and even the same word – can be communicated in a variety of ways within the same language. The text typically also contains a certain amount of distortion, including, but not limited to, misspellings, abbreviations, and slang. This feature presents a challenge to natural language processing such as classification and information retrieval. One mitigation of this situation is to normalize non-standard words to a more standard format that is easier to handle for the classification algorithms.

This paper uses the word embedding developed by Ben Lazreg et al. [19] in which they take into consideration the specificity of the social media message. The method uses a neural model to learn a function F that maps words into real vector space in such a way that the distance between two similar words (i.e., non-standard spellings of the same word, or words used in the same context) will be the shortest distance between the corresponding mapping in the real vector space.

F obeys two constraints. The first constraint is that the distance in real vector space between the mapping of a word and its non-standard versions must be shorter than the distance between that word and non-standard versions of other words. The second constraint is that the distance in real vector space between the mapping of words with similar meanings must be shorter than the distance from words with different meanings. By meaning here, we intend context since we assume that words used in the same context are more likely to have a similar meaning.

To model the first constraint, they used a denoising autoencoder. In general, the denoising autoencoder is designed to learn the pattern in the input data by adding noise to it and try to restore the original data from the noisy version. If one considers the non-standard spelling of the word to be the noisy version of the input data and try to reconstruct the standard version of the word, the denoising autoencoder will learn patterns of relationship between both.

To model the second constraint, Lazreg et al. used a neural network that predicts a word in the sentence given its surrounding words (context). The word embeddings are considered to be the weight matrix in the first layer of the neural network. At the learning phase, the embeddings are learned to maximize the log-likelihood of predicting the correct words which will assure they contain patterns about the relationship of the word and its context.

III.1.3 Topic discovery

An additional feedforward neural network takes as input the information extracted by the NER and the message embeddings and outputs the probability of the message belonging to a specific topic.

$$o = p(topic|(a, C(m))) = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{D.2}$$

where a is the information extracted by the NER, and $C(m)$ is the embedding of message m, and z_i is the weighted sum of the output of the last hidden layer. Hence, the overall output is the probability of the message being related to a specific topic given a and $C(m)$. The output of the TD is composed of the tuple *<message, number of injured, number of killed, number of missing>* where the *message* is the content of the tweet.

III.2 Query generator

A crisis is a complex event in which many variables change over time. The information needed by EMS varies from a crisis to another, and from a situation to another during the same crisis. The information-gathering process is such that each new piece of original information made available to the EMS team will trigger a need for further information which makes it an iterative process. As an example, if information about a fire event is made available, there is an obvious need to know the location and magnitude. When this is made available, the number of people in danger is important, and so on. This information-gathering process can be modeled as a graph in which the nodes represent the information needed. The availability of that information need conditions the transition for a node to another. Therefore, the edges of the graph need to be weighted by that answer. Figure D.3 gives an example of such graph. It shows that knowing there are affected persons in an extreme weather situation will trigger the need to know if they know they need to be evacuated, how still need to be evacuated, and if the evacuated persons have what they need.

It is nearly impossible to design such a graph for every type of crisis and each situation in a crisis, simply because every crisis differs in fundamental ways in practice making every crisis a one-off event the mitigation used by responders to follow general of known crises types.

Then, the question becomes twofold: Can the process of graph designing be automated? If so, what happens if we only have a graph for a limited set of situations: can that generalize to other crises and other situation in a specific crisis? In this case, the problem becomes that of link prediction in the sense of inferring missing links from an observed graph. This process is done under the assumption that if we

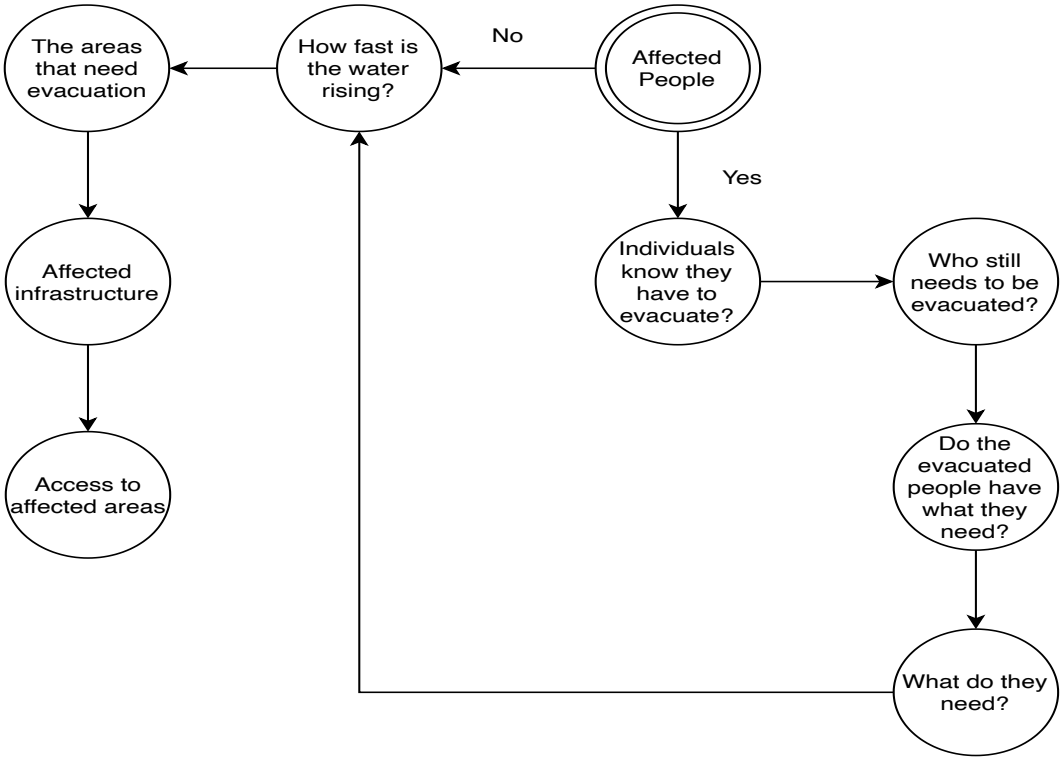


Figure D.3: Example of information needs graph for extreme weather

know that in a state A for crisis C we require information I, then, in the same state A of a similar crisis C', it is highly likely that we require the same information I.

To solve this problem, we used an extension of the memory-based neural Turing machine developed by Ben Lazreg et. al. [9] model conditional transition graphs, they call it the Conditional Neural Turing Machine (CNTM). The aim is to allow the CNTM to change state, infer missing links in a conditional transition graph, and transit from a node to another based on input received from an external environment. The CNTM is a generic link prediction algorithm that we apply in this case to a graph in which the nodes are the information needed by EMSs in a crisis.

The environment's, in this case, is the topic discovery component described in Section III.1. Its role is to produce an input that specifies if the information needed (presented by the current node) is available in social media or not i.e., the topic discovery was able, for example, to find tweets addressing that information need (which in the context of the topic discovery is the topic). As an example, if the current node is affected persons, this request will be sent to the topic discovery. If there are posts on social media about that topic, a yes answer will be sent back to the query generator which will, in turn, generate a request about knowledge of evacuation.

The transition δ form a node to the other is implemented using the CNTM.

The CNTM is an overall recurrent neural network. Its role is to produce the next node in the graph $x(t+1)$ given the previous set of nodes and the feedback from the environment. Let Q be a set of information-needs and C the set of feedbacks produced by the environment, δ is calculated using the following equation:

$$\delta(y, c) = \arg\max_{y_i \in Q} p(y_i|y, c); y \in Q, c \in C. \tag{D.3}$$

III.3 Integrated framework

The communication between different components of the framework is as follows: The QG queries the TD via its interface with the id of the requested topic. The TD send a feed request to the social media engine (SME). The SME's role is to connect to a social media platform and produce a feed of social media messages. This feed is returned to the TD. The TD classifies each message in the feed into different topics. If a message belonging to the topic requested by the QG is detected, the TD will call the QG via its interface and gives <*TRUE*> as input. The input of the call will be <*FALSE*> otherwise. The QG will then generate a new query. All the messages belonging to the query topic along with the metadata extracted by the NER will be sent to the EMS (see Figure D.4).

IV. Experimental results

The framework has been evaluated in two independent ways. First, we evaluate the components of the framework, namely the query generator and topic discovery component. Second, we evaluate the framework with by correlation analysis with the opinion of two Red Cross experts and a police officer. For both evaluations, we used extreme weather as a case study. Extreme weather is the most frequent event in Norway (Norway experiences on average 3 per year). Since all the experts evaluating the framework are from Norway, we choose the extreme weather case to put them in the most familiar situation.

IV.1 Framework evaluation

IV.1.1 Topic discovery

For the topic discovery component, we used data from *CrisisLex* [20] and CrisisNLP [21] both platform for collecting and filtering communications during a crisis. The data contains tweets about the Alberta and Queensland crises, Typhoon Hagupit, and Cyclone PAM. It is a mix of tweets where some are related to the respective crisis, and some are not. The percentage of unrelated tweets is 44% for the Alberta flood and 43% for the Queensland flood. Moreover, in the case of the Alberta flood, only 30% out of the related tweets gave useful concrete information about the crisis. It goes up to a 48% in the Queensland flood data. The data was

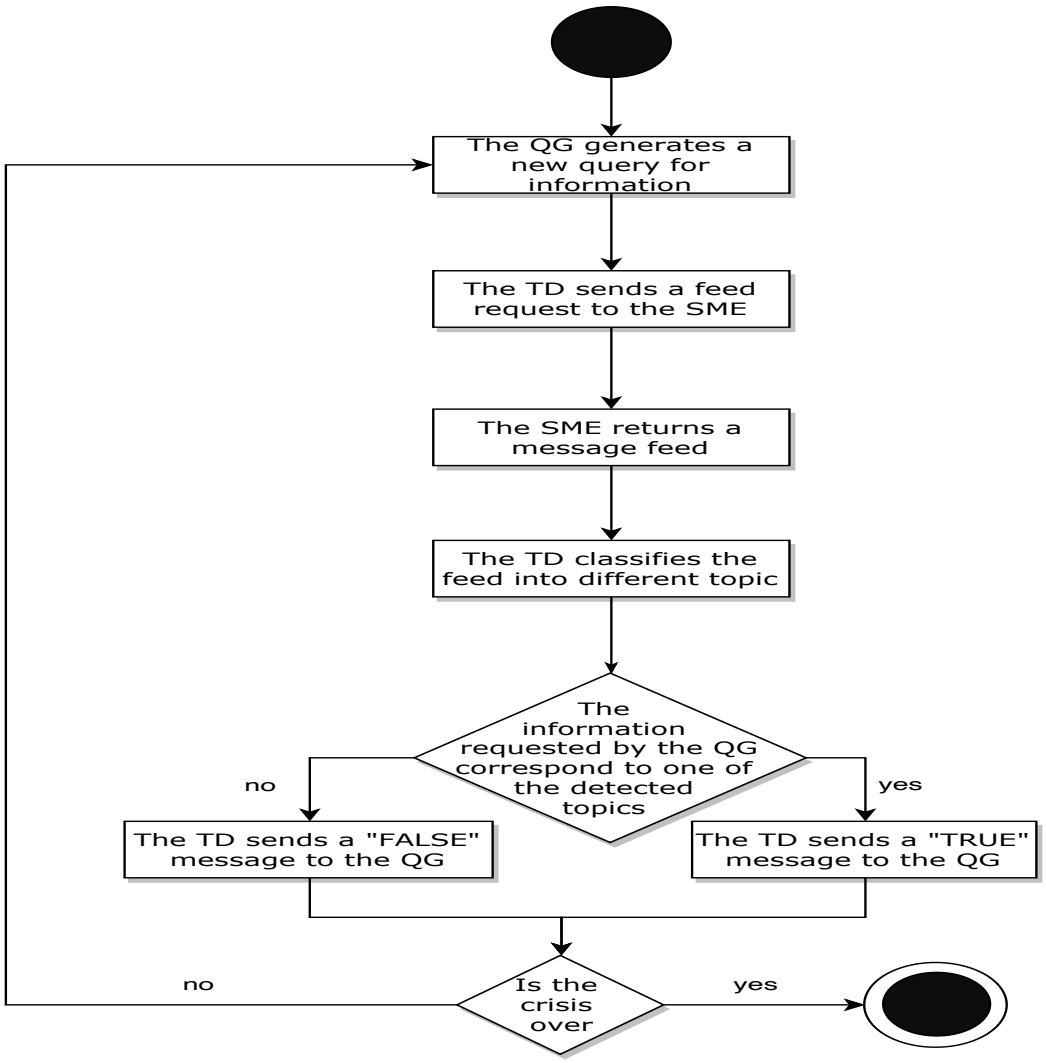


Figure D.4: Flow diagram of the framework

further labeled into 10 topics, including the ones presented in the nodes of Figure D.3. The topics are: Affected people, people needing evacuation, needs of evacuated people, water level, area needing evacuation, affected infrastructure, access to affected areas, other threats, future risks, and unrelated. 70% of the data was used for training and 30% for testing. To train and test the NER, every occurrence of the number of victims (killed, injured, and missing) in the data set was annotated in both the number and written form.

Table D.2 compares the results of our classifier with other baseline classifiers. Generally, our neural network classifier performed better than all other tested classifiers. The improvements are significant in the case of the Queensland flood, Typhoon Hagupit, and Cyclone PAM. In those case, our classifier outperformed SVM, for example by an average margin of 0.076 in F-measure. In the Alberta flood case, all classifiers achieved very comparable performance. Even though the objective of this work is not to design a state-of-the-art classifier, the off the shelf classifier we used can compete with basic and classical classifiers delivering acceptable results. However, more advanced classifiers are already developed which show more robust results. For example, Nguyen et al. developed a convolutional neural network based classifier and tested on the same data set [22]. They reported on average a 6.48% improvement over our classifier. In addition, the main problem that faces our classifier is the lower recall, which indicates that it fails to identify a bigger proportion of a topic relates tweets.

Table D.3 compares the results of the neural network classification presented in Section III.1 with a changing word embedding approach. The table shows that using our word embedding approach; the classifier performs on average 0.012 better in F-measure compared to other word embeddings. This improvement can be expected since our word embadding approach is more oriented towards social media posts with all the non-standard spellings encountered there.

Tables D.4 summarizes the results of the NER. The NER shows an acceptable performance in detecting the number of injured and killed, but do not perform as well in detecting the number of missing. Looking at the low recall indicates that the NER is not able to detect the number of missing that is labeled as a number of messing in the data set. The higher precision on the other hand indicates that it is better at eliminating numbers that are not numbers of missing. Upon further investigation of the data we noticed that a variety of words are used words to reference a missing person (“two people were kidnapped”, “3 missing persons”, “one person is lost”...). In contrast the killed and injured are more consistently referenced (“3 persons were killed”, “one injured”). This variety can be one of the reasons behind

Table D.2: Performance of the topic classification

	Alberta flood	**Queensland flood**	**Typhoon Hagupit**	**Cyclone PAM**
		SVM		
Precision	0.85	0.78	0.78	0.90
Recall	0.78	0.60	0.71	0.80
F-measure	0.81	0.68	0.74	0.85
		Logistic regression		
Precision	0.85	0.79	0.82	0.90
Recall	0.78	0.60	0.71	0.79
F-measure	0.81	0.68	0.76	0.84
		Random forest		
Precision	0.82	0.75	0.82	0.90
Recall	0.74	0.66	0.73	0.80
F-measure	0.78	0.70	0.77	0.85
		Our approach		
Precision	0.85	0.84	0.87	0.93
Recall	0.78	0.71	0.82	0.85
F-measure	0.81	0.77	0.84	0.89

the low performance of the NER in detecting the number of missing.

IV.1.2 Query generator

To test the query generator individually, we compiled different graphs representing the information need flow in different scenarios of three different crises: Fire, extreme weather, and public disturbance. The nodes in the graph present the type of information needed by emergency manager during a crisis. For a fire emergency, the graphs are extracted from the work of Nunvath et. al. [23] who did an extensive interview of firefighters about the type of information they need during different indoor fire crises. For extreme weather, the sub-graph was extracted from the work of Ben Lazreg et. al. [4] who collected personnel from police and municipality to gather the type and flow of information they need during extreme weather crises. Finally, the public disturbance graphs, as well as the rest of the graphs, were vetted by two policemen from Oslo police station who are expert in riots, demonstrations and public disturbance control. During the training phase, we only train the algorithm on graphs containing 70% of the links in the full graph. We evaluate the query on how well it can predict the missing links. The accuracy of the network in inferring the correct links for the crisis graphs is 78.59%. For more detailed, the results and

Table D.3: Results of the neural network classification with a changing word embedding approach

	Alberta flood	**Queensland flood**	**Typhoon Hagupit**	**Cyclone PAM**
		Word2Vec		
Precision	0.82	0.83	0.86	0.90
Recall	0.77	0.70	0.81	0.82
F-measure	0.79	0.76	0.83	0.86
		Glove		
Precision	0.80	0.85	0.83	0.91
Recall	0.76	0.70	0.82	0.83
F-measure	0.78	0.77	0.82	0.87
		Our approach		
Precision	0.85	0.84	0.87	0.93
Recall	0.78	0.71	0.82	0.85
F-measure	0.81	0.77	0.84	0.89

Table D.4: Performance of the NER specifically (part of topic detection)

	Number of injured	**Number of killed**	**Number of missing**
Precision	0.76	0.80	0.74
Recall	0.72	0.71	0.61
F-measure	0.74	0.75	0.67

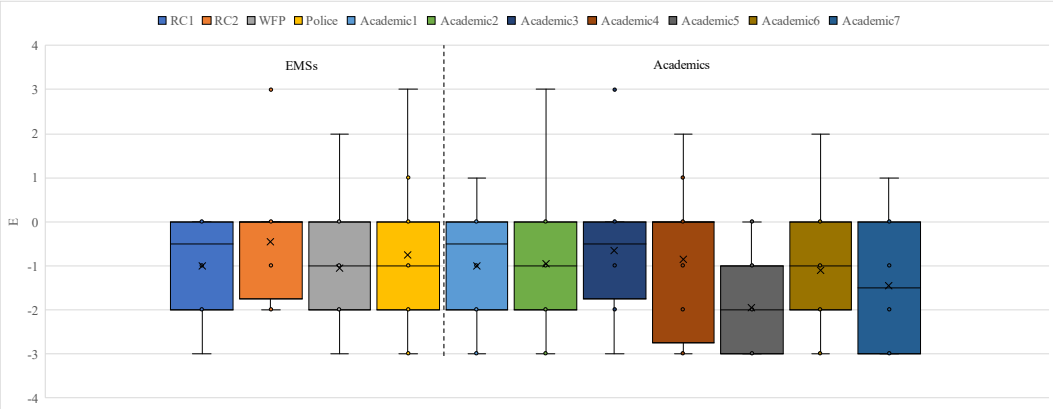


Figure D.5: Comparison of the experts opinions with the framework assessment

description of the learning process the reader can check our dedicated paper to this topic [9].

IV.2 Expert evaluation

In addition to the single component evaluation, we wanted to verify the usefulness of the whole information retrieval framework to experts in emergency management. To do so, we designed a survey as follows. We simulated a run of the framework following the same scenario given to the experts, and we retrieved the tweets judged by the framework to be most relevant to the situation. Those tweets were given a grade of 4, and the rest of the tweets a grade of 1. Let us call this grade $E_{framework}$ A total of 20 tweets are provided to each expert containing tweets picked randomly from the data set along with the tweets extracted by the framework. The experts are presented with a crisis scenario (status of the crisis and information gathered so far) and are asked to evaluates the usefulness of the tweet from a scale of 1 to 4 (4 being very useful and 1 being completely useless). Let call this grade E_{expert}. Then, we study the correlation between what the expert evaluation judges as useful and with what the framework retrieves for the same situation by examining the distribution of the difference E between $E_{framework}$ and E_{expert} (Equation D.4).

$$E = E_{framework} - E_{expert}. \tag{D.4}$$

The survey was answered by two Red Cross (RC) workers, a policeman, a Word Food Program (WFP) field worker, and seven academical experts. The results show that both the experts and the framework "agree" on the usefulness of 70.09% of the tweets. By "agree" we mean that the experts give a grade of 3 or 4 for a tweet tweets retrieved by the framework, or a grade of 1 or 2 to the information disregarded by the framework in the same situation. Figure D.5 shows a box-plot comparing the expert opinion with the information retrieved by the framework (E). It reveals that,

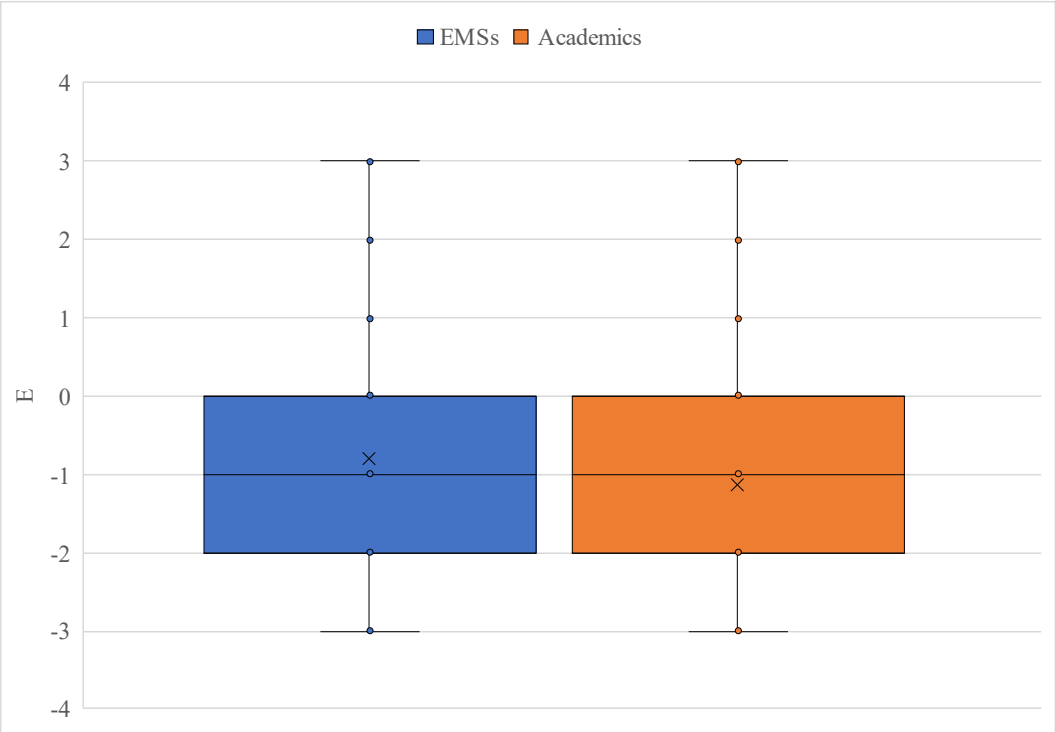


Figure D.6: Comparison of the EMSs and academic experts opinions with the framework assessment

on average, most of the expert opinion on the tweets varies within 1 point (in the survey scale) of the framework judgment. Figure D.6 shows a box-plot comparing the EMSs opinion (RC workers, policeman, WFP field worker) with the academics' opinions. The figure shows that the distribution of E does not vary much between the two types of experts. However, E tends to have negative values, which means that there is a trend of the expert judging tweets as useful while the framework did not extract those tweets. These results are justified by what we discussed in Section IV.1.1, where we mentioned that our classifier suffers from low recall, indicating that it fails to identify a proportion of the topic relates tweets.

V. Conclusion

This paper proposes an intelligent information retrieval framework from social media in crises situations. Most of the social media analysis platforms available do not put the information needed by EMS during a crisis as the central objective. The developed framework tries to fill this gap by combining two components. The first component classifies social media messages into distinct and separate topics representing a piece of information or a question asked by EMSs during a specific situation. The second component is a link prediction component that decides which

information (topic) to retrieve based on the information available so far and the status of the crisis. Since crises situations typically result in highly complex scenarios, information overload and erroneous information is a significant problem. The initial evaluation carried out by an expert in the field confirms that the framework yields the exact information needed in 70.09% of the cases.

PAPER D: REFERENCES

[1] V. S. M. W. Group *et al.*, "Lessons learned: Social media and hurricane sandy," *Homeland security science and technology*, pp. 1–39, 2013.

[2] ——, "Next steps: Social media for emergency response," *Homeland security science and technology*, 2012.

[3] J. Radianti, S. R. Hiltz, and L. Labaka, "An overview of public concerns during the recovery period after a major earthquake: Nepal twitter analysis," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, pp. 136–145.

[4] M. Ben Lazreg, N. R. Chakraborty, S. Stieglitz, T. Potthoff, B. Ross, and T. A. Majchrzak, "Social media analysis in crisis situations: Can social media be a reliable information source for emergency management services?" 2018.

[5] M. Imran, C. Castillo, J. Lucas, P. Meier, and S. Vieweg, "Aidr: Artificial intelligence for disaster response," in *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, 2014.

[6] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power, "Using social media to enhance emergency situation awareness," *IEEE Intelligent Systems*, vol. 27, no. 6, 2012.

[7] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: a survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, 2015.

[8] L. Palen and K. M. Anderson, "Crisis informatics new data for extraordinary times," *Science*, vol. 353, no. 6296, pp. 224–225, 2016.

[9] M. B. Lazreg, M. Goodwin, and O.-c. Granamo, "A neural turing machine for conditional transition graph modeling," *arXiv preprint arXiv:1807.2767782*, 2019.

[10] M. Ben Lazreg, N. Noori, T. Comes, and M. Goodwin, "Not a target. a deep learning approach for a warning and decision support system to improve safety and security of humanitarian aid workers," in *IEEE/WIC/ACM International Conference on Web Intelligence*, ser. WI '19. New York, NY, USA: ACM, 2019, pp. 378–382. [Online]. Available: http://doi.acm.org/10.1145/3350546.3352551

[11] A. Zielinski, S. E. Middleton, L. N. Tokarchuk, and X. Wang, "Social media text mining and network analysis for decision support in natural crisis management," *Proc. ISCRAM. Baden-Baden, Germany*, 2013.

[12] J. Rogstadius, M. Vukovic, C. Teixeira, V. Kostakos, E. Karapanos, and J. A. Laredo, "Crisistracker: Crowdsourced social media curation for disaster awareness," *IBM Journal of Research and Development*, vol. 57, no. 5, 2013.

[13] Z. Ashktorab, C. Brown, M. Nandi, and A. Culotta, "Tweedr: Mining twitter to inform disaster response," *Proc. of ISCRAM*, 2014.

[14] C. Caragea, N. McNeese, A. Jaiswal, G. Traylor, H.-W. Kim, P. Mitra, D. Wu, A. H. Tapia, L. Giles, B. J. Jansen *et al.*, "Classifying text messages for the haiti earthquake," in *Proceedings of the 8th international conference on information systems for crisis response and management (ISCRAM2011)*. Citeseer, 2011.

[15] R. Power, B. Robinson, J. Colton, and M. Cameron, "Emergency situation awareness: Twitter case studies," in *International conference on information systems for crisis response and management in mediterranean countries*. Springer, 2014, pp. 218–231.

[16] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier, "Tweetcred: Real-time credibility assessment of content on twitter," in *International Conference on Social Informatics*. Springer, 2014, pp. 228–243.

[17] A. S. Jadhav, H. Purohit, P. Kapanipathi, P. Anantharam, A. H. Ranabahu, V. Nguyen, P. N. Mendes, A. G. Smith, M. Cooney, and A. P. Sheth, "Twitris 2.0: Semantically empowered system for understanding perceptions from social data," 2010.

[18] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Computational linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[19] M. B. Lazreg, M. Goodwin, and O.-C. Granmo, "Combining a context aware neural network with a denoising autoencoder for measuring string similarities," *Computer Speech & Language*, p. 101028, 2019.

[20] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg, "Crisislex: A lexicon for collecting and filtering microblogged communications in crises." in *ICWSM*, 2014.

[21] M. Imran, P. Mitra, and C. Castillo, "Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages," *arXiv preprint arXiv:1605.05894*, 2016.

[22] D. T. Nguyen, K. A. Al Mannai, S. Joty, H. Sajjad, M. Imran, and P. Mitra, "Robust classification of crisis-related data on social networks using convolutional neural networks," in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[23] V. Nunavath, A. Prinz, and T. Comes, "Identifying first responders information needs: supporting search and rescue operations for fire emergency response," *International Journal of Information Systems for Crisis Response and Management (IJISCRAM)*, vol. 8, no. 1, pp. 25–46, 2016.

Paper E

Title:	Not a Target. A Deep Learning Approach for a Warning and Decision Support System to Improve Safety and Security of Humanitarian Aid Workers
Authors:	Mehdi Ben Lazreg[†], Nadia Noori[‡], Tina Comes[†], and Morten Goodwin[†]
Affiliation:	[†] Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway [‡] Norwegian Research Center - NORCE, N-4898 Grimstad, Norway
Conference:	*IEEE/WIC/ACM International Conference on Web Intelligence*, Thessaloniki, Greece, October 2019.
Copyright ©:	ACM

E

Not a Target. A Deep Learning Approach for a Warning and Decision Support System to Improve Safety and Security of Humanitarian Aid Workers

Mehdi Ben Lazreg, Nadia Noori, Tina Comes, and Morten Goodwin

Abstract — **Humanitarian aid workers who try to provide aid to the most vulnerable populations in the Middle East or Africa are risking their own lives and safety to help others. The current lack of a collaborative real-time information system to predict threats prevents responders and local partners from developing a shared understanding of potentially threatening situations, causing increased response times and leading to inadequate protection.**

To solve this problem, this paper presents a threat detection and decision support system that combines knowledge and information from a network of responders with automated and modular threat detection. The system consists of three parts. It first collects textual information, ranging from social media, and online news reports to reports and text messages from a decentralized network of humanitarian staff. Second, the system uses deep neural network techniques to automatically detects a threat or incident and provide information including location, threat category, and casualties. The neural network is composed of three parts: A named entity recognizer (NER) that extracts the location, casualties, target, and attacker for the text, a word embedding that transforms the text into a vector, and a feedforward neural network that detects the type of threat based on the word embedding and the information extracted by the NER. Third, given the type of threat and the information extracted by the NER, a feedforward network proposes a mitigation plan based on humanitarian standard operating procedures. The classified information is rapidly redistributed to potentially affected humanitarian workers at any level. The system testing results show a high precision of 0.91 and 0.98 as well as an F-measure of 0.87 and 0.88 in detecting the threats and decision support respectively. We thus combine the collaborative intelligence of a decentralized network of aid workers with the power of deep neural networks.

Keywords— Humanitarian disasters, conflict, threat detection, decentralized decision support, deep learning, neural network.

I. Introduction

The enduring humanitarian crises in the Middle East, the unrelenting high levels of violence in Afghanistan, and the new outbursts of violence in Sub-Saharan Africa

have turned the provision of aid into these countries a continuous risk for humanitarian aid workers. Despite the availability of tracking and monitoring technologies [15], and the increasing efforts of humanitarian organizations to protect their staff, the number of humanitarian workers that fall victim to violence remains at a high level.

To recognize and respond to threats, aid workers need timely information that allows them to understand their situation, and react to a very volatile situation. Yet, data collection, processing, and decision-making in humanitarian organizations is often focused on strategic decisions at headquarters level, leading to delays, bottlenecks and putting the safety of operational aid workers at risk. What is missing is thus a decentralized system that enables data collection from various sources; rapid information processing and sharing of threat information.

In this paper, we make headway in bridging this gap between operational field workers and the strategic level by making threat information accessible to decision-makers in the field. We introduce a deep neural network solution designed for threat detection and decision support in conflict zones. The fact that deep neural networks can be trained one layer at a time enables the modular building of threat detectors and a decision support platform. This feature allows us to propose a threat detection and decision support system that feeds on the different types of textual information that typically will need to be triangulated (in term of comparing different information for the purpose of fact checking) and processed manually, including social media, news reports, and most importantly reports from humanitarian aid workers in the field.

Our system is composed of four layers. The first layer is a named entity recognition layer. It is trained to extract metadata (including the location, target, attacker, casualties) from the textual information it receives. The second layer is a work embedding layer. It codes the text into a vector (the text presenting syntactical similarities will have closer vectors in term of cosine similarity). The third layer combines the embedding and the extracted metadata to deduce a probability distribution over types of threats. Finally, the fourth layer proposes mitigation and protective measures on the bases of type of incident (i.e. threat type and extracted metadata) and set of applied Standard Operational Procedures (SOPs) adopted by humanitarian organizations.

I. Background: Humanitarian aid in conflict zones

II.1 Aid in danger: Information needs and system requirements

The persisting trend is that violence against aid workers is increasing globally. To

mitigate this problem, humanitarian organizations often transfer risks to local partners and operate remotely from neighboring countries or regions. However, this means that information sharing and analysis is directed "upwards" to enable planning and coordination at a regional level. In highly volatile and risky environments, such as those driven by warfare and continuously shifting lines with limited access and resources, feedback to operational responders is provided too late, and too little [7].

Knowing who did what, where, and when, (4W) has been one of the most important tools for coordination. Overlaying 4W maps, often manually, with information on (armed) actors, internally displaced persons, and refugees is currently the most important coordination mechanism for the identification of risks and threats toward humanitarian missions [1]. Analyses are often provided too late, and in fragmented ways and are hampered by a lack of trusted data sources, or unreliable and insecure information sharing and communication channels, thus placing humanitarian responders at risk and preventing efficient response [6].

In response to delays within official reporting channels, alternative and social networks have been established to share important information via tools that are known and easily usable, such as WhatsApp. This implies that information is lost to actors who do not have access to the network and that there is no standard on how to process information, thus leaving room for rumors, misreporting and judgemental biases [5].

In such an environment characterized by time pressure, mis-trust, complexity and uncertainty, the information acquired by team members must be easy to understand, reliable, secure, fast to process and must include clear cues that trigger action [17]. Moreover, the information should be efficiently shared within the team such that the team can collectively make decisions on the basis of better situation awareness. Furthermore, the process of obtaining and sharing information in aid networks always entails a delicate balance between information richness and information centralization [16].

In this research work, we focus on "information processing mechanisms", "Long term memory stores," and "automaticity" components to improve the situational awareness of humanitarians on the ground. By using artificial intelligence tools and existing data, we provide them with near-real time information about potential threats and concrete decision support to make timely, informed decisions.

Our work is embedded in the European Research Project iTRACK, which aims to create an integrated monitoring and tracking system that improves the safety of humanitarian aid workers in high-risk and fragile areas. iTRACK is designed to

the support responders to acquire and categorize valued information to collectively make informed decisions in real-time. The system comprises different components related to tracking, threat detection, navigation, logistics, and coordination in humanitarian disasters. To use data and information in protective measures, it is necessary to capture the "small" events that are neglected in regular security protocols and to understand the events that may constitute or trigger an actual threat. The project focuses on one of todays largest disasters the conflict zones in the Middle East. All information about the system architecture, the individual components and the tests and validations are available on `www.itrack-project.eu`.

II.2 State of the art: artificial intelligence for threat detection

To extract threat-related messages from information channels such as social media, news and threat reports, machine learning techniques are increasingly suggested [8]. Supervised and unsupervised learning are the primary investigated approaches.

In supervised approaches, the goal is to classify the information available, such as social media messages, on the basis of the crisis event it describes. Within a crisis, supervised learning techniques are also used to classify social media messages into subevents that occurred. To achieve this classification, the algorithm learns a predictive function that classifies any new unknown message as part of one of the categories of crises. A number of approaches have been investigated including Naïve Bayes, support vector machine (SVM) [18], random forests [9], and logistic regression [2]. Some researchers focus on only analyzing tweets containing certain keywords [8] to replace manual labeling for training. As an example, SVM was used to classify tweets related to earthquakes and landslides [12, 14]. In a supervised approach, labels are necessary for training classifiers, but they might be highly difficult to obtain, particularly in the case of multilanguage messages or context knowledge [8].

Unsupervised methods are used to identify patterns in unlabeled data. They are most useful when it is not exactly known what information to look for. An example is grouping tweets into stories (clusters of tweets) after a keyword filter [13]. This method reduces the number of social media messages to be handled by humans because it groups similar messages. Another application using unsupervised learning identifies events related to public and safety by using a spatio-temporal clustering approach [4]. In addition to strictly clustering elements into groups, soft clusters allow items to belong simultaneously to several clusters with variant degrees. In this approach, the words contained in the tweets and the length of the tweets are

used as the basis for computing the similarity [10]. The method was applied during the earthquake in Indonesia (2009) to detected different aspects related to the crisis (relief, deaths, missing persons, etc.).

Despite the aforementioned advances in social media classification, there still is a gap to fill to address specific requirement of humanitarian aid workers operating in conflict zones. First, detecting the type of event is insufficient. In certain event, more information need to be presented to the aid worker in a precise and brief manner. This information includes the location, target, attacker and casualties in case of a terrorist attack for example. Second, the classification framework need to be scalable enough to be applied for different data sources. In addition to social media, these data sources might include news feeds, and internal data sources to the humanitarian organization. Finally, beyond the detection and classification of an event, a mitigation plan needs to be quickly available to support the humanitarian aid worker in their decision making process at the presence of a threat.

III. Threat detection and decision support system: Design

To meet the requirements for humanitarian conflicts, we designed a threat detection and decision support system, see Figure E.1.

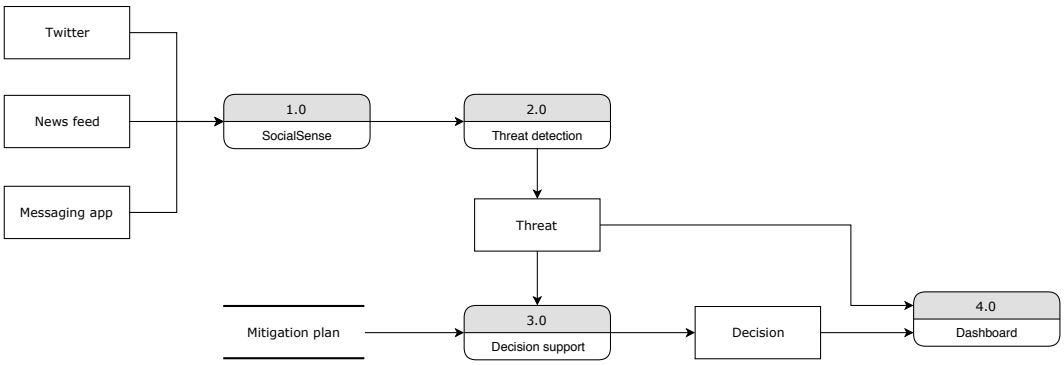


Figure E.1: Flow diagram for threat detection and decision support

This system consists of three main parts: SocialSense, threat detection & decision support, and a dashboard. SocialSense collects information from social media (twitter), RSS news feed, and messages shared between personnel through the secure iTRACK messaging app. It delivers this information in a unified payload to the threat detection and decision support system. The threat detection and decision support detects, which of the received pieces of information indicate a threat, and recommends mitigation plans. Furthermore, it communicates discovered threats to humanitarian staff at coordination and operational levels through a monitoring dashboard (see Figure E.2). This design enables decentralized information sharing for

rapid response and avoids the information clutters or standstills described in Section II.1.

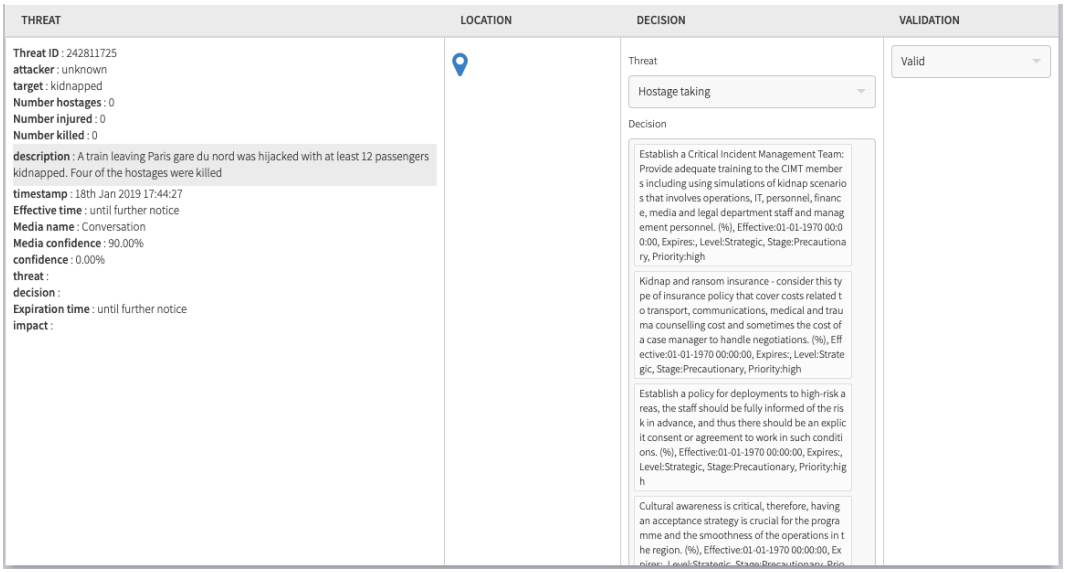


Figure E.2: Illustration of the dashboard

Detected threats are presented together with a confidence level (ranging from zero to one), which is an indication of how self-confident the component is about its decision. The confidence level attribute is crucial information for humanitarian workers because knowledge about information quality can play a critical role in decision-making.

Once a threat has been detected, the decision support module comes into play. Its first functionality is to find a suitable action based on the predicted threat. It does so by taking a threat report as input and uses a neural network based system to generate possible action plan for the incident based on the attributes of the incident that will mitigate the threat.

E

IV. Threat detection model

The threat detection module detects threat patterns in textual data. A threat pattern is based on features found within the data that correlate with a threat. Examples of these features include specific words and combination of words that typically describe a threat.

IV.1 Framework

To extract threat patterns, the development framework is composed of three essential functions illustrated in Figure E.3:

- A named entity recognizer (NER) which extracts metadata from the text it receives. This information includes location, target, attacker, and number of victims. For the remainder of this paper, we will call the information extracted by NER as threat attributes.
- A word embedding that transforms the text into a vector. Reports belonging to the same threat will have a short distance between their vectors embedding in terms of cosine similarity.
- A neural-network-based detection layer that combines the coded text and the threat attributes to produce the probability of a threat. The decision layer produces a probability distribution over type of threats.

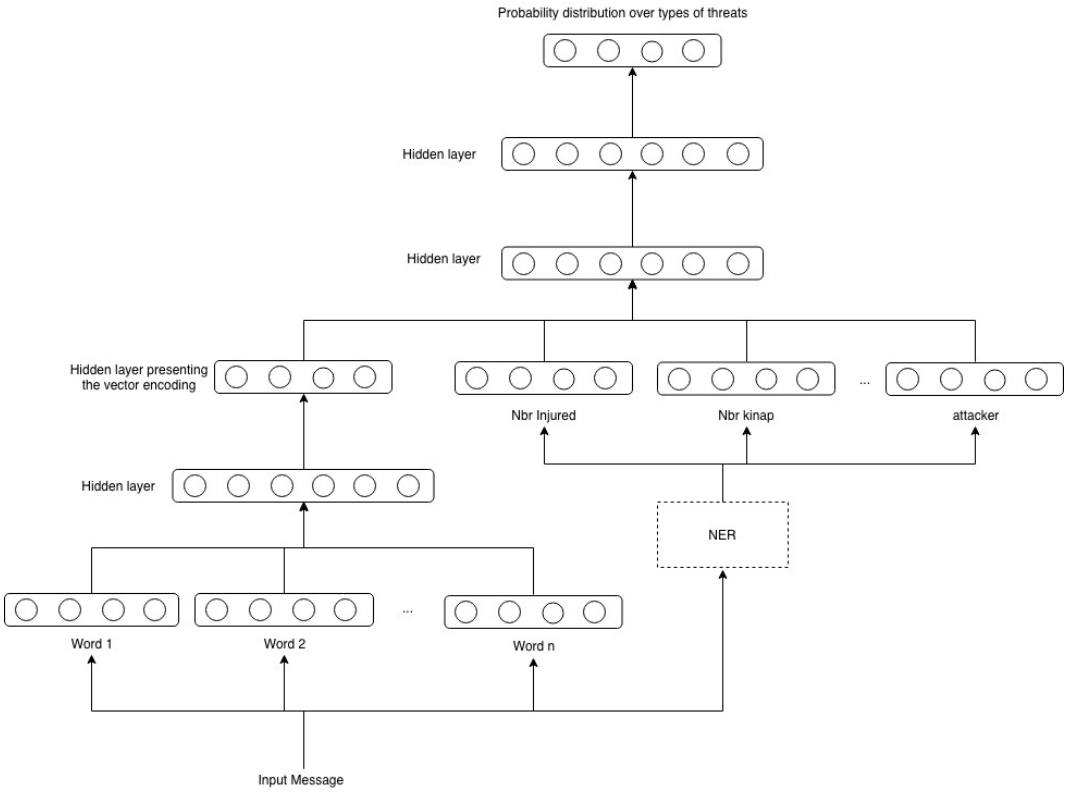


Figure E.3: Threat detection model

IV.2 Named entity recognizer

The NER uses a maximum entropy classifier [3], which is a statistical classifier, to identify named entities by looking at each word in a sentence and deciding whether it is the start of a named entity, the continuation of an already started entity, or not part of any known name. By combining these predictions, the classifier is used to identify a sequence of words that make up a name. Given that the NER uses

a probabilistic model, it is possible to determine the probability associated with an identified name: names to which the model assigns low probabilities are less likely to be accurate. In the context of a threat to humanitarian aid workers, we want to identify the received (potentially threat related) text contains the target, the attacker, the location, and the number of killed, injured, and kidnapped people.

The algorithm can be explained by the following example. Suppose that this component receives a text that contains details about an attack on an aid convoy, and we want to extract the name of the attacker from the message. Let Y be the set of all the possible words in the message. The algorithm is a process that finds which word $y \in Y$ is the attacker influenced by a contextual information x surrounding the words. Our approach is bases on maximum entropy that constructs a stochastic model that accurately represents the behavior of the random process, i.e., it estimates the conditional probability of y given a context x. The process will output $p(x|y)$ where p is an element of the set P of all the possible probability distributions.

To do so, the model studies the behavior of the random process on a collection of training samples. The building blocks of the model are a set of statistics found in the training sample. For example, the model might find that there is statistical evidence in the training samples that the name of the attacker appears after the word "attacked" in the message. This statistical evidence is included in the estimation of a context given a word $p(x|y)$ thus limiting our search. To mitigate an infinite number of probability distributions, we select the most uniform distribution (following the principals of insufficient reasoning).

The threat attributes are also used to compare threats: If a the same threat is detected from a different source or the same source and its threat attributes match a previously detected threat than it will not be forwarded to dashboard. The mechanism will ensure that the same threat is not reported twice. However, the reader have to notice that in some cases the NER might fails to extract all the threat attribute from a certain text. For example, suppose we have two different texts describing the same bombing. For the first text, the NER is able to extract the location and the number of killed. For the second text, it extracts the number of injured. In this case both text will be considered as describing different incidents. The mechanism is, therefore, not perfect, but it ensures at least the elimination of redundant threat with identical input text.

IV.3 Word embedding

The embedding sub-part of threat detection transforms a message to a vector that captures the semantics of the message. The coding method is based on the

by work done by [11] on word embedding. The objective is to find a function F that maps words into real vector space in such as a way the distance between two similar words (i.e., non-standard spellings of the same word or words used in the same context) will be the smallest distance between the corresponding mapping in the real vector space.

To achieve this goal, F needs to obey two constraints. The first constraint is that the distance in real vector space between the mapping of a word and its non-standard versions must be shorter than the distance between that word and non-standard versions of other words. The second constraint is that the distance in real vector space between the mapping of words with similar meanings must be shorter than the distance between words with dissimilar meanings. We define meaning by similar context: we assume that words used in the same context have a similar meaning. To model the first constraint, the method uses a denoising autoencoder, and to model the second constraint, it introduce a context encoder which tries to predict a word in the message on the basis of the surrounding words following the continuous bag of word method.

IV.4 Detection layer

An additional feedforward neural network takes the output of the threat attributes and the message coding and outputs the probability of the message as a specific threat or not a threat.

$$o = p(threat|(a, F(m))) \tag{E.1}$$

where a is the threat attributes, and $F(m)$ is the vector representation of the message. Hence, the overall output is the probability of the message being related to a specific threat given the threat attributes and the message encoding.

V. Decision support model

The decision support component proposes mitigation and protective measures on the basis of threats (and their attributes) as described in Section IV. When a new threat is detected, the decision support component receives this threat from the threat detection module (see Figure E.1) and then uses a neural network to suggest an action plan (decision) to mitigate the incident. Similar to threat detection, the suggested decisions come with a confidence level. The proposed decision also take into consideration the specificity of each situation i.e. type of threat and threat attributes.

The decision support component is composed of a neural network that takes as input the type of the detected threat represented by o (calculated in Equation

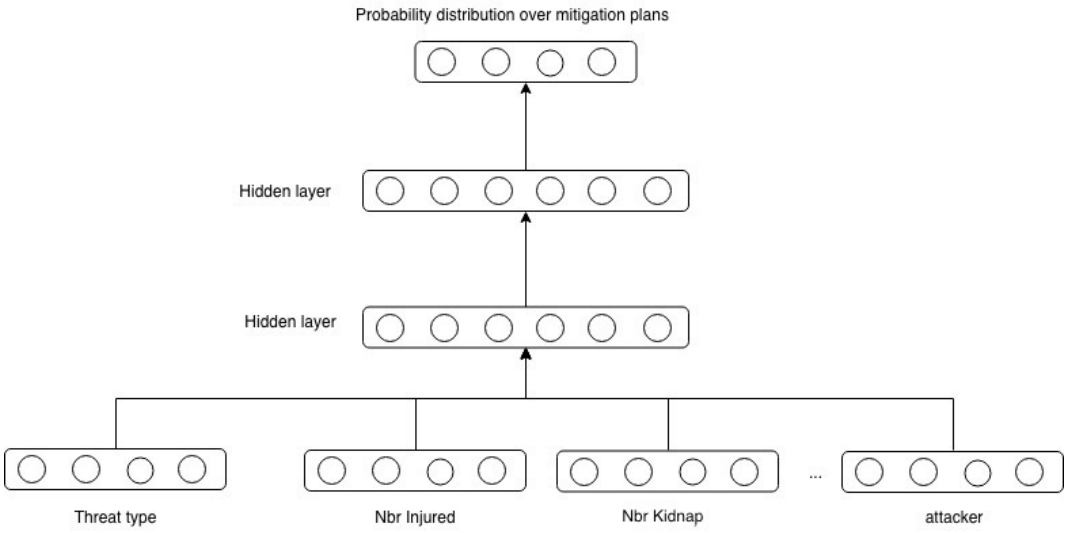


Figure E.4: Decision support model

E.1), and the threat attributes a extracted by the NER and produces a probability distribution over mitigation plans $m = (mitigation_1, ..., mitigation_q)$ (see Figure E.4):

$$m = \sigma(b_d + U_d tanh(d_d + H_d(o, a))). \tag{E.2}$$

In this context, the neural network contains h_d hidden units, and b_d is the output layer biases (with q elements), U_d is the output to hidden layer weights (a $q \times h_d$ matrix), d_d is the input layer biases (with $|(o, a)|$ elements) and H_d is hidden to input layer weights (a $h_d \times |(o, a)|$ matrix). The neural network is trained using stochastic gradient decent to minimize the cross-entropy loss function over the training data:

$$L = \frac{1}{q} \sum_i p(mitigation_i) log(m_i). \tag{E.3}$$

The mitigation plan with the highest probability giving the input of the network will be recommended as the plan to follow.

The whole model (threat detection and decision support) forms a deep neural network in which each layer is trained to perform a specific task and calculate features to feed to the next layer.

VI. Case Study Set-Up

We illustrate and model our approach by using the example of threat prediction in the Middle East. In the following, we describe the underlying data and sources.

VI.1 Threat Detection

The Global Terrorism Database is an open-access database that includes information on terrorist events around the world. The database contains entries on domestic, transnational, and international terrorist incidents that have occurred from 1970 through 2016, exceeding 170,000 cases. Each incident has information on the target, the number of people affected (killed, or kidnapped), and the group or individual responsible when identifiable.

The information in the Global Terrorism Database is used to train and test the NER to identify a threat and classify the nature of the threat. To this end, we annotated every occurrence of a location, target, attacker, and number of killed, injured, and kidnapped in every article in the dataset as follows: for the location, we consider the names of countries and cities. For the target and attacker, we annotated the name of the group or organization that made the attack or was the victim of the attack and their abbreviations. For the number of killed, injured, and kidnapped, we annotated them in both the number and written form. We trained the NER to detect the annotated attributes.

The Global Terrorism Database contains over 4,000,000 news articles and 25,000 news sources were reviewed to collect incident reports from 1998 to 2016. These reports are used to train and test the threat detection component. However, we did not use all 170,000 incidents as our dataset; we filtered the incident connected to humanitarian workers in the MENA region, which amounts to 12,000 incidents. We also added to the data another 4,000 articles not related to the threats and we grouped them in a category named "others." We divided the data into training and test sets (70% for training, 30% for validation).

For the massages shared between personnel, we have collected 693 different massages from the iTRACK messaging app during the initial exercises/testings of the iTRACK system. The messages were labelled to the appropriate type of threat or the "others" category. Similar to previous data, the messages were divided into training and validation sets (70% for training, 30% for validation)

VI.2 Decision support

For mitigation planning and recommendations, we created a digital repository based on a set of Standard Operating Procedures (SOPs) adopted by humanitarian organizations operating in unstable and high-risk areas. The SOPs were extracted from the existing documents from UN agencies and humanitarian organizations, such as the United Nations Department of Safety and Security [5] or best practices and guidelines available on the European Inter Agency Security Forum [6]. For an overview of all considered SOPs and the underlying methodology, see the iTRACK

[5] https://training.dss.un.org/
[6] https://www.eisf.eu/resources-library/

Table E.1: Performance of the NER

	Target	**Attacker**	**Location**	**Number of killed**	**Number of injured**	**Number of kidnaps**
Precision	0.80	0.86	0.91	0.76	0.80	0.74
Recall	0.62	0.82	0.87	0.72	0.71	0.61
F-measure	0.70	0.84	0.89	0.62	0.75	0.66

project deliverable[7].

The set of SOPs was categorized on the basis of the types of threat type, context, impact, and attack vector (e.g., travel SOP for convoy situations or office SOPs for mob violence and attacks). The SOPs also contain a level tag that indicates at which level of the organization the procedures needs to be implemented (strategic, tactical, operational), and implementation stage tag which indicated at which stage of the mission to implement the procedure (precautionary, responsive).

We created a dataset that assigns the appropriate SOPs to each threat in the dataset used for threat detection on the basis of categorization described previously. Similar to threat detection, the decision support dataset was divided into training and validation sets (70% for training, 30% for validation) to train and evaluate the decision support.

VI.3 Results

For the experiments, we used 300 hidden units for the word embedding, 80 nodes for the feed froward network. All weights and the memory were initialized using a Xavier initialization. The network was trained with RMSprop stochastic gradient descent with a learning rate of 0.001 a batch size of 32.

Our model is evaluated by precision (ratio of true threats detected), recall (ratio of detected threats that are actually true), and F-measure (harmonic mean between precision and recall). Tables E.1 show the performance of the NER, and Table E.2 shows that of the threat detection and decision support.

The NER is evaluated on the basis of its accuracy in detecting the annotated name in the test set. For example, in case of location, if the NER detects the annotated name of the city, country, or both in the article then is it is a success. We do not consider deitetic terminology. If multiple locations are detected in an article, we consider the location with the highest probability as the correct location. Table E.1 shows that on average, 80% of the threat attributes from the NER extracts are

[7] https://www.itrack-project.eu/page/media_items/d2.6---policies-for-contingency-planning-and-case-brief35.php

Table E.2: Performance of the threat detection and decision support

	Threat detection on news reports	**Threat detection on messages**	**Decision support**
Precision	0.91	0.94	0.98
Recall	0.84	0.92	0.80
F-measure	0.87	0.93	0.88

correct. However, the low recall of 0.7 indicates that the information retrieval is far from complete, and 30% of the threat attributes in the test data are not retrieved.

Furthermore, Table E.2 shows that the threat detection component can achieve a high F-measure on both the data from the news report and the messaging app because it relies on the message encodings and the threat attributes. The threat attributes are used as supporting information for the final layer of the threat detection component. When these pieces of information (especially the number of injured, killed, and kidnapped) are detected, they will help a great deal in positioning the message in the write category. However, when this information is not available or incorrect, the model can still classify the article correctly by encoding the message. During the learning phase, the neural network also learns how much importance it should give to the information received by the NER and the coding. It is important to note that the threat detection performance with regard to all categories of threats and the category "others".

The decision support component shows a very high precision of 0.98, but a much lower recall of 0.80 (a difference of 0.18). The high precision is evidence that all decisions presented are correct, but the smaller recall indicates that many correct decisions are not presented by the system. Given that the decision support bases its decision not only on the type of threat but also on the threat attributes, the output depends on the threat attributes.

In many situations, the location of the threat influences the correct mitigation approach, e.g., a threat in Iraq will require different mitigation measures than a threat in Egypt. The dynamic nature of threat detection and mitigation hinders the development of a complete automated decision support system.

VII. Conclusion

Humanitarian aid in conflict zones face great challenges in term of operational coordination and security risks management. Current approaches in event detection from social media lack in three different aspect crucial to humanitarian aid worker operating in conflict zones: First, information about the event presented to

the aid worker in a precise and brief manner. Second, the scalability of classification framework. Finally, a mitigation plan needs to be quickly available to support the humanitarian aid worker in their decision making process.

In this paper, we presented a tool for supporting collaborative threat detection and decision support system that aims for enhancing safety and security of operational conditions in high-risk areas. The proposed solution is a combination of different artificial intelligence techniques and data classification. The tool testing results showed high precision in identifying threats from textual messages and in proposing the most relevant decisions to mitigate that threat. The proposed tool is a step forward to close an existing information gap between strategic and operational/field levels in the humanitarian aid operations by facilitating collaborative decision making as it communicates detected incidents and proposed mitigation plans in real-time to relevant personnel in the organization for further actions. It also serves as a decentralized information and decision support system platform for automatically informing humanitarian workers regardless of their rank about a threat and the ways to deal with the threat.

PAPER E: REFERENCES

[1] Nezih Altay and Melissa Labonte. 2014. Challenges in humanitarian information management and exchange: evidence from Haiti. *Disasters* 38, s1 (2014).

[2] Zahra Ashktorab, Christopher Brown, Manojit Nandi, and Aron Culotta. 2014. Tweedr: Mining twitter to inform disaster response. *Proc. of ISCRAM* (2014).

[3] Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics* 22, 1 (1996), 39–71.

[4] Michele Berlingerio, Francesco Calabrese, Giusy Di Lorenzo, Xiaowen Dong, Yiannis Gkoufas, and Dimitrios Mavroeidis. 2013. SaferCity: a system for detecting and analyzing incidents from social media. In *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*. IEEE.

[5] Tina Comes. 2016. Cognitive biases in humanitarian sensemaking and decision-making lessons from field research. In *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*. IEEE, 56–62.

[6] Tina Comes, Olga Vybornova, and Bartel Van de Walle. 2015a. Bringing structure to the disaster data typhoon: an analysis of decision-makers information needs in the response to Haiyan. In *2015 AAAI Spring Symposium Series*.

[7] Tina Comes, Niek Wijngaards, and Bartel Van de Walle. 2015b. Exploring the future: Runtime scenario selection for complex and time-bound decisions. *Technological Forecasting and Social Change* 97 (2015), 29–46.

[8] Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing social media messages in mass emergency: a survey. *ACM Computing Surveys (CSUR)* 47, 4 (2015).

[9] Muhammad Imran, Carlos Castillo, Ji Lucas, Patrick Meier, and Sarah Vieweg. 2014. Aidr: Artificial intelligence for disaster response. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*.

[10] Kirill Kireyev, Leysia Palen, and Kenneth Anderson. 2009. Applications of topics models to analysis of disaster-related twitter data. In *NIPS Workshop on Applications for Topic Models: Text and Beyond*, Vol. 1. Canada: Whistler.

[11] Mehdi Ben Lazreg and Morten Goodwin. 2018. Combining a Context Aware Neural Network with a Denoising Autoencoder for Measuring String Similarities. *arXiv preprint arXiv:1807.06414* (2018).

[12] Aibek Musaev, De Wang, and Calton Pu. 2014. LITMUS: Landslide detection by integrating multiple sources. In *11th International Conference Information Systems for Crisis Response and Management (ISCRAM)*.

[13] Jakob Rogstadius, Maja Vukovic, CA Teixeira, Vassilis Kostakos, Evangelos Karapanos, and Jim Alain Laredo. 2013. CrisisTracker: Crowdsourced social media curation for disaster awareness. *IBM Journal of Research and Development* 57, 5 (2013).

[14] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*.

[15] Red Cross Societies et al. 2013. World disaster report. *Geneva: International Federation of Red Cross and Red Crescent Societies (http://www. ifrc. org)* (2013).

[16] Bartel Van de Walle, Bert Brugghemans, and Tina Comes. 2016. Improving situation awareness in crisis response teams: An experimental analysis of enriched information and centralized coordination. *International Journal of Human-Computer Studies* 95 (2016), 66–79.

[17] Bartel Van de Walle and Tina Comes. 2015. On the nature of information management in complex and natural disasters. *Procedia Engineering* 107 (2015), 403–411.

[18] Jie Yin, Andrew Lampert, Mark Cameron, Bella Robinson, and Robert Power. 2012. Using social media to enhance emergency situation awareness. *IEEE Intelligent Systems* 27, 6 (2012).

Paper F

Title: Semantic Decay Filter for Event Detection

Authors: Mehdi Ben Lazreg[†], Usman Anjum [‡], Vladimir Zadorozhny [‡], and Morten Goodwin[†]

Affiliation: [†] Dept. of Information and Communication Technology, University of Agder (UiA), N-4898 Grimstad, Norway
[‡] Dept. of Informatics and Networked Systems, School of Computing and Information, University of Pittsburgh, PA 15260, USA

Conference: *The* 17th *International Conference on Information Systems for Crisis Response and Management (ISCRAM 2020)*, Virginia, USA, May 2020.

F

Semantic Decay Filter for Event Detection

Mehdi Ben Lazreg, Usman Anjum, Vladimir Zadorozhny, and Morten Goodwin

***Abstract* — Peaks in a time series of social media posts can be used to identify events. Using peaks in the number of posts and keyword bursts has become the go-to method for event detection from social media. However, those methods suffer from the random peaks in posts attributed to the regular daily use of social media. This paper proposes a novel approach to remedy that problem by introducing a semantic decay filter (SDF). The filter's role is to eliminate the random peaks and preserve the peak related to an event. The filter combines two relevant features, namely the number of posts and the decay in the number of similar tweets in an event-related peak. We tested the filter on three different data sets corresponding to three events: the STEM school shooting, London bridge attacks, and Virginia beach attacks. We show that, for all the events, the filter can eliminate random peaks and preserve the event-related peaks.**

Keywords—String metric, Event detection, Crisis management.

I. Introduction

The past few years have shown a rise in the use of various micro-blogging services like Twitter for people to share information [1]. In Twitter, individuals, organizations, and governments spread and collect information to obtain an accurate and complete picture of significant happenings in the world. It is, therefore, no surprise that many researchers use tweets to detect and gather information about several events [1].

An event is a real-world one-time occurrence usually defined based on specific spatial and temporal properties [1]. Some types of events that have been studied in the literature include natural disasters (e.g., earthquakes, floods, fires and typhoons), sports events, infectious diseases outbreaks, traffic incidents, riots, terrorist acts, weather updates, conferences, exhibitions, and festivals. An event is a rare occurrence that will change the normal tweeting behavior. By observing these patterns and looking at changes in the tweeting behavior, information about events can be obtained. Finding information about an event allows local authorities to take timely response to deal with the event and inform the public.

The purpose of this paper is to differentiate patterns specific to an event from those of routine tweeting behavior. We are thereby allowing a way to find the time of an event. Discriminating event signature and patterns from standard tweeting patterns is not straight forward. Previous researchers have approximated standard tweeting patterns by using different distributions, and deviations from this distribution were used to discriminate an event pattern. For example, the work in [2] used exponential distribution to approximate the tweeting pattern in a time series of tweets. Another work showed that peaks in a time series of hash count and topic count of tweets can be used to discriminate event patterns from standard patterns [3]. The rise in the number of tweets after an event is attributed to people discussing an event to express their opinions or inform others. Consequently, the position of the peak could be used as an indicator that an event has occurred and can be used to find the time and location of an event. However, a problem with using peaks to differentiate event-related behavior from standard tweeting behavior is how to determine which peak corresponds to an event. A time series of tweets could have multiple peaks due to routine tweeting behavior. For example, there could be more tweets in the day than in the night as people are more active in the day, which would manifest as a peak. Therefore, it becomes necessary to find a method to find the peak that can identify an event.

For this purpose, we propose a similarity-based method to accurately determine if a peak in time series of tweets corresponds to an event or is just part of routine tweeting behavior. We hypothesize that if an event occurs, the number of similar tweets in the peak following the event would be higher than other peaks occurring due to standard tweeting behavior. We propose a string metric that measures the similarity between tweets ad gives a higher value for two that are more similar. Further, we introduce a similarity threshold: Two tweets that have a similarity measure higher than the threshold are considered similar. The proportional decrease in the number of similar tweets in a specific peak as the similarity threshold increases is referred to as decay. We use that decay as a part of a function to detect which peak corresponds to an event. The function is a Semantic Decay Filter SDF that eliminates random peaks in the number of tweets and preserver the event-related peaks. Therefore, the function SDF would have a peak around the same time as an event-related peak. Figure F.1 gives an example of such a function, assuming that the event-related peak occurs around 23h00. We empirically test SDF on three different twitter data sets related to the STEM school shooting, London bride attacks, and Virginia beach attacks.

In summary, our goals in this paper are as follows:

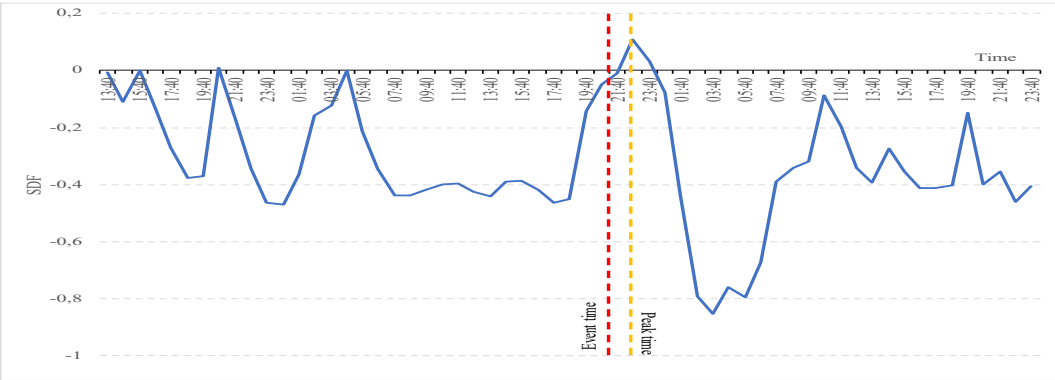


Figure F.1: SDF as a function of time for the London bridge attacks: The SDF function reaches a peak (a value above 0) during the period of the peak containing event-related tweets (22h40)

Formulation and Algorithm: We propose finding a methodology to discriminate patterns generated due to events from patterns related to standard tweeting behavior. We assume occurrence of peaks as manifestation of an event. However, to determine the peak corresponding to an event we propose a novel approach of using similarity analysis of the texts within the tweets.

Accuracy: Based on our experiments, we are accurately able to find peaks very close to an event at different granularities.

Generality: Our proposed models can be applied to other micro-blogging systems and domains that have both time-series data as well as be used to identify location of an event.

II. Literature Review

II.1 Event detection

Event detection looks at how to detect and analyze situations, such as a crises, using twitter data when an event occurs. An important aspect of event detection is to identify patterns by looking at the behavioral change that occurs whenever an event takes place. For example [4] looks at different ways people may respond to a terrorist attack and their behavior during the recovery process. They use the Paris 2015 attack to verify the different hypotheses. The behavioral change of the people is manifested through various ways in tweets, and this information can be used to detect events. We consider the occurrence of a peak as the behavioral change when an event occurs.

There are many works found in the literature that look at event detection through tweets. A list of surveys of techniques focusing on event detection in Twitter can be found in [5], [1], [6], [7], [8], [9], [10], [11] and [12]. Each of these surveys

focuses on a specific aspect of event detection. Steiger et al. [5] categorize papers according to their academic discipline, application, and methods and classify existing research into event detection, location inference, and social network analysis. Atefeh et al. [1] event detection techniques are classified based on the type of event, detection method, and detection task. Events are categorized as specified or unspecific detection methods as supervised and unsupervised and detection tasks as new event detection or retrospective event detection. They identify sparseness in tweets and vocabulary as a major challenge in detecting new events. The survey in [6] and [7] provided an overview of event detection techniques in social media. Imran et al. [8] surveyed event detection techniques based on their role in emergencies and disaster management.

Most of the detection methods focus on the content of the tweets and use classification or clustering to detect events. They improve detection by applying some form of modification in the text analysis part. For example, data fusion has been discussed in [13] to improve event detection. However, they implement data fusion by combining text and images to improve accuracy. Another example is the work in [14] that used text mining for event detection by grouping a set of words with similar burst patterns and then applying wavelet transform and [15] again used text analysis by focusing more on large-scale events. Another collection of works that combine text analysis and clustering but not mentioned in the above surveys include [16], [15], and [17].

Some works combined spatial and temporal aspects of tweets with different levels of scales for event detection. For example, Abdelhaq et al. [18] divide the region of interest into grids, and then keywords are extracted based on their temporal and geospatial properties and are then clustered. A cluster is defined as a localized event if its keywords have a high burstiness degree. The *Eyewitness* [19] algorithm looks through a corpus of geotagged tweets over localized regions in space and time for unusual spikes in tweet counts. They discretize with a hierarchical triangular mesh and time as periods of lengths, which means there can be different spatial and temporal resolutions. An event is defined as a peak above a baseline tweet count, which is obtained through regression. The event is located within a triangle and time window. However, during pre-processing, they removed retweets and repeat tweets, which we believe plays a significant role in event detection and used the texts of the tweets within the peak to identify the events. The real-time version of [19] was implemented in [3] but they mostly focused on *hashtags*. The work in [20] presented a geo-social event detection method focusing on the geographical regularities of local crowd behaviors to detect events. However, they had a fixed

time window, and their geographic grids are created based on a clustering-based space partition method.

There is one work found in the literature that used similarity analysis in finding the time and location of an event. Dong et al. [21] a stream of tweets is taken at different spatial and temporal resolutions, similar words are extracted to create a time series and a wavelet-based method is applied to measure the similarity of the time series. Then different clusters are created that correspond to events of different temporal and spatial resolutions. The work in [22] identifies different levels of news coverage and its relationship with countries looking at different spatial resolutions. Shao et al. [23] also looks at finding the location and time of an event as well as predicting its occurrence using graphs. However, they require keywords to locate an event, which means unknown events whose keywords are not known may not be detected.

Sakaki et al. [2] used the spatial-temporal information from tweets to find the epicenter of an earthquake and trajectory of typhoons. First, semantic analysis of the texts in the tweets is done to extract the relevant tweets. The authors assume that tweets follow an exponential distribution with time. The spatial information about the event is estimated using Kalman filters and particle filters. Kalman filters and particle filters have also been used for data fusion in sensors, e.g., in multisensor target tracking [24]. Another work estimated the event's location by assigning probabilities using DempsterShafer (DS) theory to find an event's possible locations based on geotags, texts in tweets, and user profile [25], [26]. However, they only considered two levels of granularity and require coordinates and names for assigning probabilities. This paper was extended to include fine-grained event localization in [27].

II.2 String metrics

An important part of our approach proposed in the introduction is the string metric that computes the similarly between tweets. A string metric or string distance function, defines a distance between every element of a set of strings A.Over the years, several attempts at defining an all-encompassing string metric have been carried out. The most well-known of these is the edit distance (Levenshtein distance) [28]. It counts the minimum number of operations (deletion, insertion and substitution of a character) required to transform one string into another. It also assigns a cost to each operation. The edit distance is called a simple edit distance when all operations have the same cost and a general edit distance when operations have different costs. The edit distance has four notable variants: Longest common subsequence (LCS) [29], the Hamming distance [30], the Damerau-Levenshtein

Distances [31], and the episode distance.

The the q-gram distance is based on counting the number of occurrences of common q-grams (strings of length $q \in \mathbb{N}$) in each string, the strings having a closer distance the more q-grams they have in common [32]. The N-gram distance is an extension of the edit and LCS distance to consider the deletions, insertions, and substitutions of N-grams. The use of N-grams enabled a certain number of new statistical methods for string metrics originating from the field of samples and sets. The use of N-gram introduces the notion of statistical string metrics, which are metrics that measure the statistical properties of the compared strings like for example, the Sorensen-Dice coefficient and the Jaccard Index [33] [34],

Machine learning techniques have also been used to learn an embedding of words capturing the similarity between them. As example, a method called local linear embedding computes a low-dimensional, neighborhood-preserving embedding of high dimensional input. The method is applied to generate a two-dimensional embedding of words that preserves their semantics [35].

Furthermore, feedforward neural networks have been used to generate a distributed vector representation of words [36]. By predicting the next word giving the previous words in the context, the neural network learns a vector representation of the words in its hidden layer [37]. Another word vector representation variant learns for each word a low dimensional linear projection of the one-hot encoding of a word by incorporating the projection in the energy function of a restricted Boltzmann machine [38] [39]. Finally, GloVe learns a log-bi-linear model that combines the advantages of global matrix factorization and local context window to produce a vector representation of word based on the word count [40].

Traditional similarity methods only look at the intrinsic properties of words. Word embedding methods, on the other hand, do not take into account a second constraint, namely that non-standard spellings of a word should also have similar vector representations. In Section IV, we introduce the similarity measure that we will use in this work, which balances between both constraints.

III. Significant peak detection

Figure F.2 show a time series of number of tweets for the three considered event. We can observe that the occurrence of an event (red vertical line in Figure F.2) is followed by a peak as there is a large change in the number of tweets along the temporal dimension around. We define a peak as a local maximum. We also consider a point as a peak if it is greater than n points before it and smaller or equal to n points after it. For simplicity we consider only adjacent points, i.e. $n = 1$.

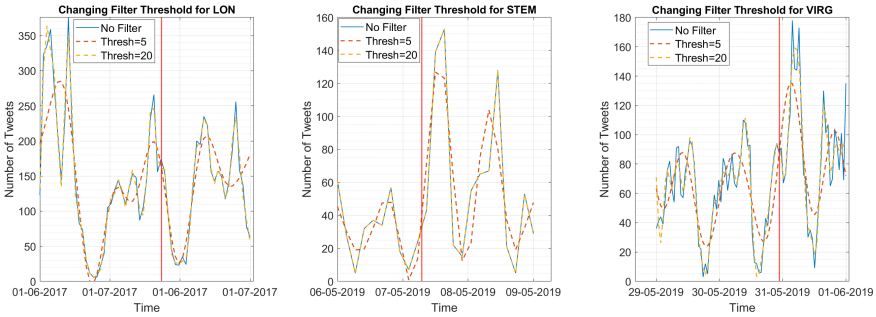


Figure F.2: Effect of Filter Threshold

If there is a sequence of numbers of tweets varying with time or space, the position of the peaks can determine the time or location of the event. However, not all the peaks are indicative of an event. Some peaks could also be due to routine Twitter behavior and are part of a standard recurring Twitter pattern. These peaks can be removed by passing the tweet count through a low pass filter. The low pass filter should remove any minor variations and large peaks would be kept. We refer to such peaks as *"significant peaks"*. However, a significant peak is not necessary an event related peak. As Figure F.2 shows, multiple significant peaks are present after different filter thresholds are applied. Ideally all significant peaks should be caused by events. However, that is not always the case. For the London bridge attack for example, peaks occurring before the event are bigger than the peaks that could be caused by the events. Therefore, low pass filter may not be ideal to identify peaks cause by events. The significant peaks that are due to events are called *event-related peaks*. Event-related peaks can be used to identify the time (and or location) of an event. Differentiating event-related peaks from significant peaks can be complicated and we propose using a similarity based analysis to identify event-related peaks.

IV. Methodology

IV.1 Similarity methodology

As low pass filters are in some cases unable to find event-related peaks due to the random peak problem, the objective of our methodology is to propose a filter that can remedy that problem. We hypothesise that the event-related peaks would have a lot of similar tweets as users will refer more about an event in their tweets. To achieve this goal, we define a semantic decay filter SDF based on a string similarity measure.

For the similarity measure we use the function F [42] that maps words into real vector space $\mathbb{R}^n$ in such a way that the distance between two similar words

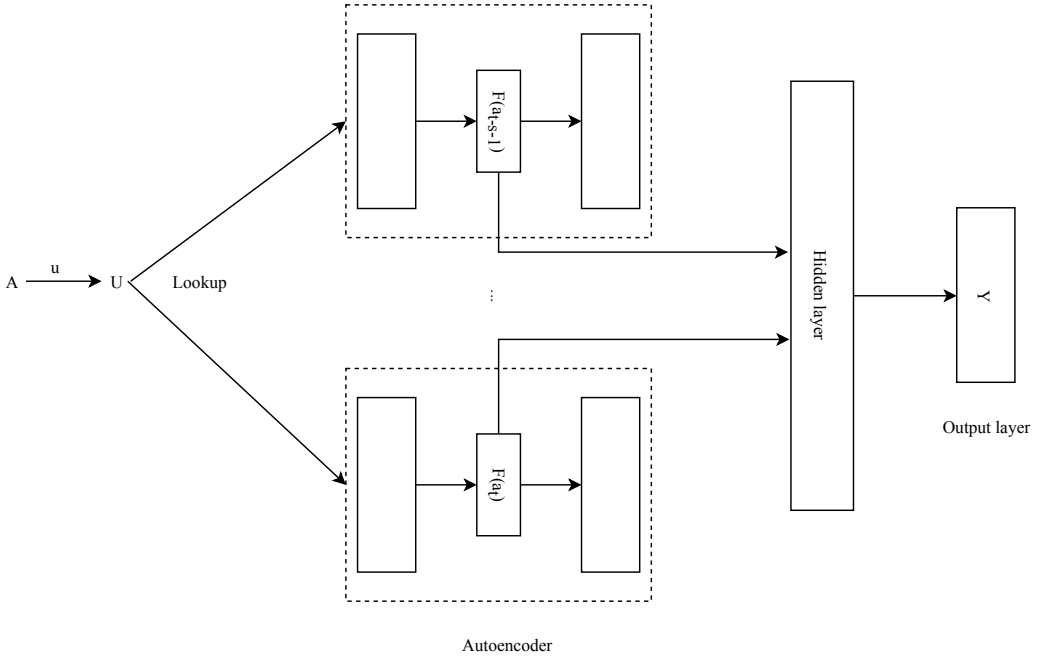


Figure F.3: Overall architecture of the autoencoder in combination with the context encoder to find the similarity between the words. In this figure, A is a vocabulary, $a_i \in A$, u is an initialization faction: $A \longrightarrow \mathbb{R}^n$, U is a real numbers matrix, and y is softmax activation

(i.e., non-standard spellings of the same word, or words used in the same context) will be the shortest distance between the corresponding mapping in the real vector space. F obeys two constraints. The first constraint is that the distance in real vector space between the mapping of a word and its non-standard versions must be shorter than the distance between that word and non-standard versions of other words. The second constraint is that the distance in real vector space between the mapping of words with similar meanings must be shorter than the distance between words with dissimilar meanings.

To implement the first constraint, we use a denoising autoencoder. A denoising autoencoder is a neural network that takes as input a vector with added noise and tries to reconstruct the original vector. By doing so, it capture features and patterns in the vectors. If we consider the non-standard version of a word to be the noisy input of the autoencoder than the latest will be able to capture pattern of relation between the non-standard and standard versions of the same word in its hidden layer. For the second constraint we use a neural network that predicts a word in sentence given its surrounding words (context). The word embeddings are considered to be the a weight matrix in the first layer of the neural network. The embedding are learned to maximizes the log-likelihood of predicting the correct words which will assure they contain patterns about the relationship of the word and its context.

We call this part a context encoder. Figure F.3 shows the overall architecture of the neural network. The combination of both the denoising autoencoder and the context encoder yield the function F.

The function F is most useful in the context of Twitter data because of the number of non-standard spelling and spelling mistakes in the platform.

IV.2 Semantic decay filter

Let T_1 and T_2 be two tweets composed of words $[t_1^1, ..., t_1^m]$, and $[t_2^1, ..., t_2^k]$ respectively, and D be a similarity measure in $\mathbb{R}^n$. The similarity between two tweets is the distance between the average of the embedding of the words in both tweets:

$$Similarity(T_1, T_2) = D(\sum_{i=1}^{m} F(t_1^i), \sum_{i=1}^{k} F(t_2^i)). \tag{F.1}$$

We choose D to be the cosine similarity which makes $Similarity(T_1, T_2) \in [-1, 1]$. A similarity of 1 will mean that both tweets are completely identical. If we set a threshold $a \in [-1, 1]$ and we consider T_1 and T_2 to be similar only if $Similarity(T_1, T_2) > a$, then the number of tweets that are similar to T_1 will decrease as a approaches 1. Let $A = a_0, ..., a_M$ be a series of number in $[0, 1]$ where:

$$\begin{aligned} &a_0 = 0 \\ &\forall i < i; a_i < a_j \\ &a_M = 1. \end{aligned}$$

Our hypothesis is that the the tweets in an event-related peak will share a common topic and similarities as compared to tweets in a random peak. If T_1 is an event-related tweet in an event-related peak, the decay in the number of tweets similar to T_1 as $a_k \in A$ approaches one will be lower than in the case T_1 is not event-related. Thus, the aggregated decay in similarity over all the tweets in an event-related peak will be slower than in a non-event-related peak.

Let N_t^k be the aggregated number of similar tweets is a set of tweets $T_t = \{T_1, T_2, ..., T_n\}$ posted during a time window t for a threshold a_k. N_t^k is given by the following equation:

$$N_t^k = \sum_{i=1}^{n} \sum_{j=1}^{n} 1_{Similarity(T_i, T_j) > a_k}. \tag{F.2}$$

Where:

Table F.1: Summary of Real Data

Event Name	Event Date and Time	Event Location (latitude, longitude)	Reference Name	Data Collection Location (latitude, longitude)
London Bridge Attacks	06-03-2017 2216 hrs	51.508056, -0.087778	LON (Location 0) LON (Location 30)	51.508056, -0.085717 51.50605, -0.080155
STEM School Shootings	05-07-2019 1353 hrs	39.556, -104.9979	STEM (Location 1) STEM (Location 2) STEM (Location 3)	39.58482, -104.99790 39.58096, -104.97928 39.55599, -104.96067
Virginia Beach Shootings	05-31-2019 1644 hrs	36.7509, -76.0575	VIRG	36.77974, -76.05750

$$1_{Similarity(T_i,T_j)>a_k} = \begin{cases} 1 & \text{if } Similarity(T_i, T_j) > a_k \\ 0 & \text{otherwise} \end{cases}$$

The decay λ_t^k is then calculated using the following equation:

$$\lambda_t^k = \frac{1}{a_k - a_0} \log \frac{N_t^k}{N_t^0}. \tag{F.3}$$

A special case can occur in which, during the time window t, the number of posted tweets is low. Nevertheless, those tweets are similar or identical. In this case, will observe a low decay corresponding to depression in the number of tweets. Thus, using the decay alone as an indicator is not sufficient. Our indicator is a linear combination of the normalized decay and the normalized number of tweets $|T_t|$ in a specific time window:

$$SDF_t = \alpha Norm(|T_t|) + \beta Norm(\lambda_t^k) \tag{F.4}$$

where $Norm$ is a normalization function, α, and β are real numbers.

F

V. Results and discussion

V.1 Data

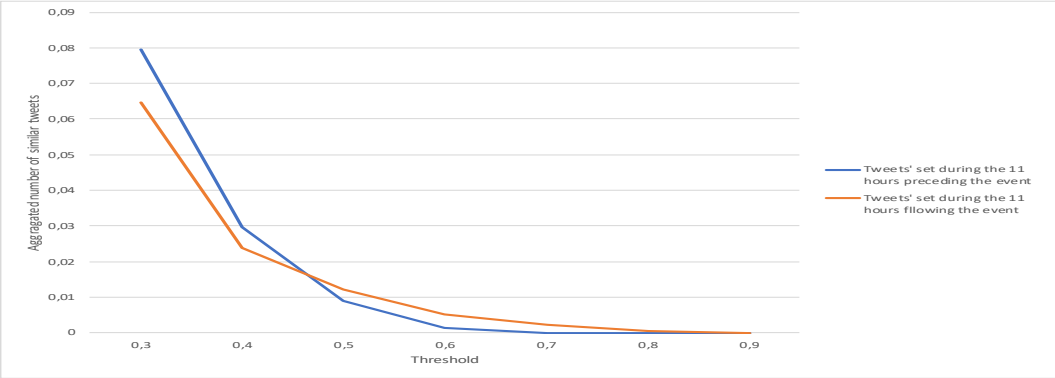


Figure F.4: Normalized aggregated number of similar tweets as a function of the threshold for STEM school shooting in location 0

The raw tweets were collected directly from the Twitter API using the 'TwitteR' package in R [43]. The data set with their details are summarized in Table F.1. The table shows the event name, the coordinates of the event and the time at which the event occurred. The reference name refers to the name used for the data sets in the experiments. The tweets were collected at different distances from the actual coordinate of the event. The "Data Collection Location/Distance" column gives the coordinate at which the tweets were collected and the distance from the event location. The tweets are aggregated into specific time windows. Figure F.2 shows the plots of the data set and how they change for different time windows. The increasing time windows are representation of the number of tweets at coarser granularity.

V.2 Similarity decay

The first objective of this section is to verify the first hypothesis enounced in Section IV; namely to confirm if the decay in the aggregated number of similarities (Equation F.3) in an event-related peak is lower than the decay in non-event peaks. To do so, we compare such decay on three different events described in Section V.1. We collected tweets form three different locations for the STEM school shooting, one location for the Virginia shooting, and two locations for the London attacks. At each location, we compare the decay obtained on the cluster of tweets shared in the 11 hours before the event and the 11 hours following the event for the STEM school shooting, in the 16 hours before the event and the 16 hours following the event for the Virginia shooting, and in the 11 hours before the event and the 11 hours following the event for the London attacks. The duration over which we made the measurements was predicated by the duration of the event peak i.e., the time frames correspond to the length of the event-related peaks.

Figures F.4 to F.7 show the aggregated number of similar tweets (Equation F.2)

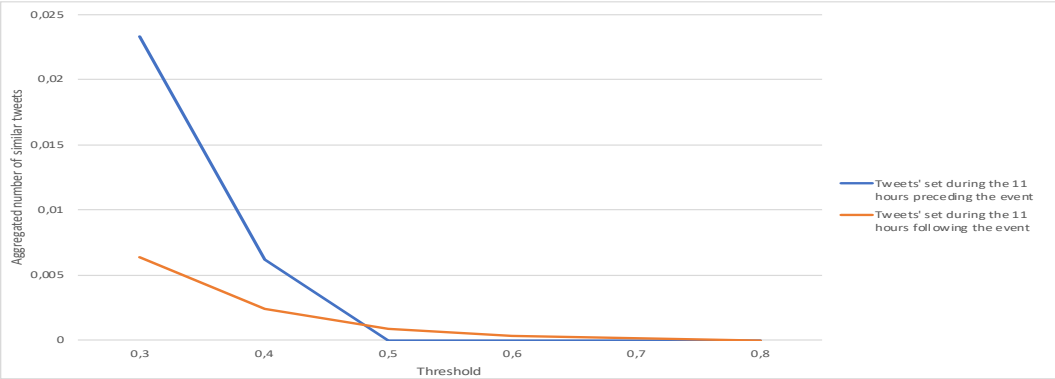


Figure F.5: Normalized aggregated number of similar tweets as a function of the threshold for STEM school shooting in location 30

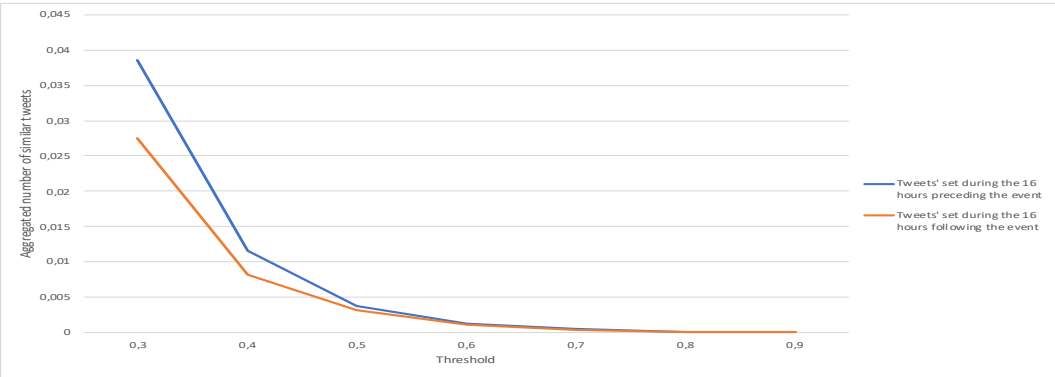


Figure F.6: Normalized aggregated number of similar tweets as a function of the threshold for Virginia attacks

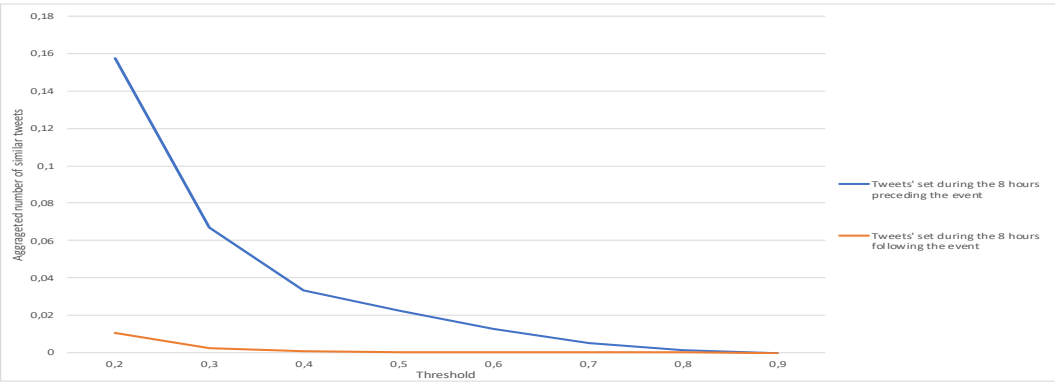


Figure F.7: Normalized aggregated number of similar tweets as a function of the threshold for London attacks in location 3

F

Table F.2: Decay in number of similarity

Data	λ pre-event	λ post-event
STEM school shooting location 1	8.45	1.97
STEM school shooting location 2	3.01	1.37
STEM school shooting location 3	5.74	4.26
London bridge attacks 0	3.91	3.21
London bridge attacks 30	4.20	4.83
Virginia shooting	4.63	2.52

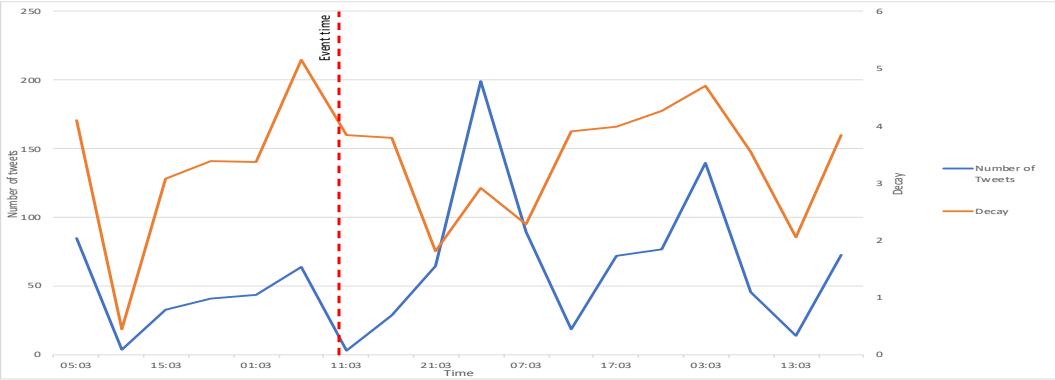


Figure F.8: Decay as a function of time for the STEM school shooting

for each event at each location for a similarity threshold a_k going from 0 to 0.9. The aggregated number of similar tweets was normalized to values between 0 and 1 for the sake of presentability and comparability. For example, at the first location in the STEM school shooting (Figure F.4), the number of aggregated similar tweets went down before the event from 1558 for a threshold of 0 to 0 for a threshold of 0.9. After the event, it went down for 33656 for a threshold of 0 to 226 for a threshold of 0.9. During the Virginia shooting, the number of aggregated similar tweets went down before the event from 338028 for a threshold of 0 to 8 for a threshold of 0.9. After the event, it went down for 993755 for a threshold of 0 to 1437 for a threshold of 0.9. On average, the number of aggregated similar tweets at the 0.9 threshold is at 23.66 before the event, and at 478.75 after the event. This result indicated that the number of nearly identical tweets is 20 fold higher in an event-related peak.

Table F.2 compares the decay (Equation F.3) in the number of aggregated similar tweets as the threshold increases pre and post-event (The decay of the curves in Figures F.4 to F.7). As the table shows, for all the data we tested, the decay is, for all but one location in the London attacks, lower during event-related peaks. On average, the decay is 3.02 for event-related peaks and 4,99 pre-event, which represents a 1.65 fold decrease. This result confirms our hypotheses.

Figures F.8 to F.10 divide the time over which the data collection was made

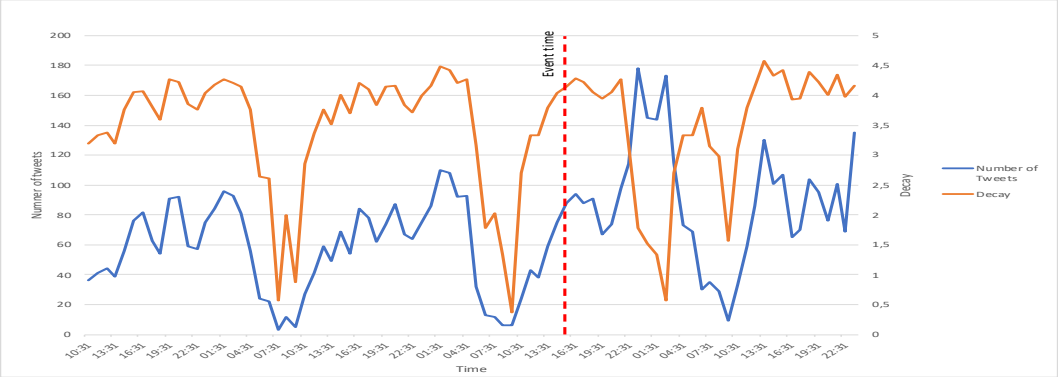


Figure F.9: Decay as a function of time for the Virginia attacks

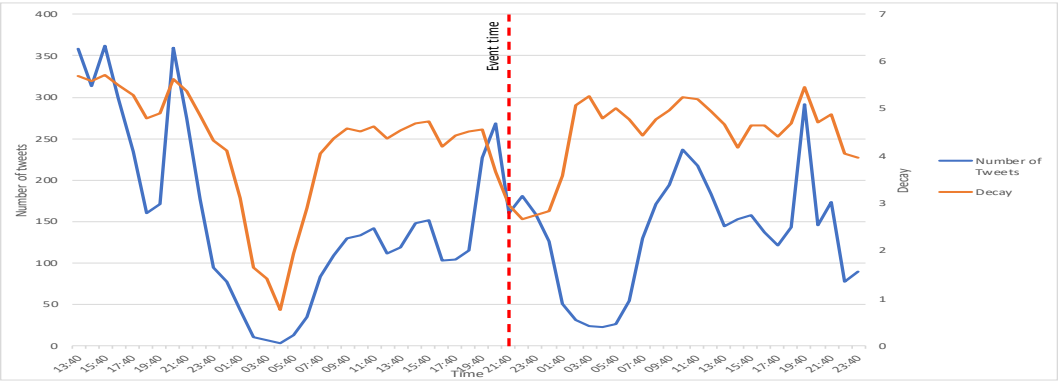


Figure F.10: Decay as a function of time for the London bridge attack

F

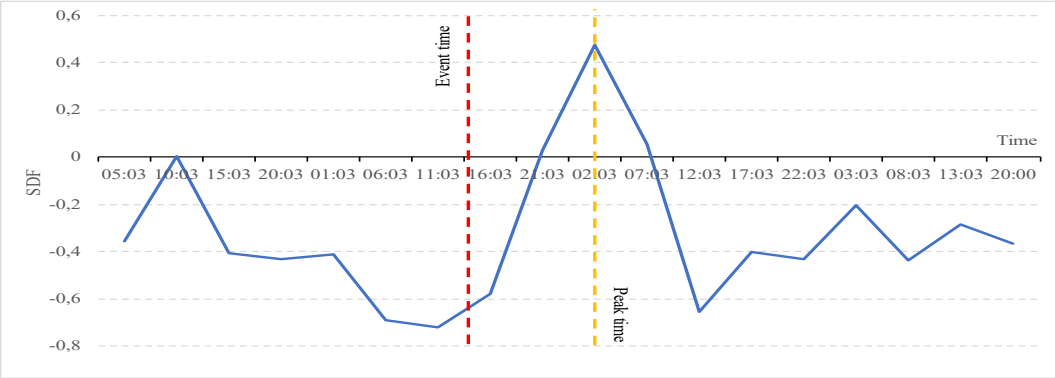


Figure F.11: SDF as a function of time for the STEM school shooting

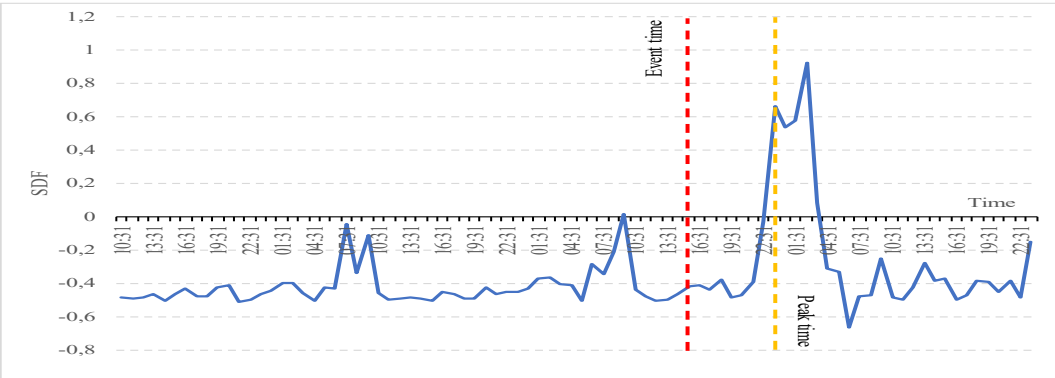


Figure F.12: SDF as a function of time for the Virginia attacks

into windows t of length 1 hours (Figure F.10 and F.9), and 5 hours (Figure F.8). We choose those windows for each dataset based on the limitation of the required minimum number of tweets (no less than 2) present in each time window. The figures show the number of tweets (blue line) together with the decay (Equation F.3) in the number of similar tweets (orange line) in each time window. The figures show a depression of the decay coinciding with an event-related peak. It is important to note that this depression in the decay does not occur around other peaks in the number of tweets outside the event-related peak. However, the figures show regular depressions in the decay occur every day in the early morning (between 5 and 7 am). This depression coincides with a low traffic time on Twitter at which a minimum amount of tweets is shared in the areas of data collection (less than ten tweets). Upon examination of those tweets, we noticed that the majority are automated and identical weather reports, thus the low decay. We mentioned the drawback of only using the decay as a filter at the end of the Section IV which led us to define the function SDF.

Figures F.1, F.11, and F.12 show the function SDF as a function of time for the considered event. For all figures, we chose the parameters $\alpha = 1$ and $\beta = -1$ in

Table F.3: Event time according to the significant peak and the peak in SDF

Event	Event time	Significant peak time	SDF peak time	Difference between peaks
STEM school shooting	13h53	2h03	2h03	0h00
London bridge attacks	22h06	19h40	22h40	21h00
Virginia shooting	16h04	23h31	23h31	0h00

Equation F.4. Those parameters grantee that SDF peaks around the event-related peak. As the Figures show, SDF reaches a value higher than zero only around the event-related peak for all the events and thus can be viewed as an indicator. Table F.3 compares the real-time of the event with the time at which the significant peaks occur (see Section III) and SDF peaks. Note that, since we use a 5 hour time window for the STEM school shooting, for example, the time of the peaks is the end of the time window (2h03 for the STEM school shooting). However, the tweets were taken between 21h33 and 2h03. The Table also shows, in the case of the London bridge attack where the event-related peak is submerged by other peak, if we follow the significant peak approach, the event-related peak would be located 12h after the event. However, the SDF can accurately detect the cluster of tweets related to the event, and peaks 36 minutes after the event.

VI. Conclusion

This paper proposes to solve the randomly occurring peak-problem in analytics of social media posts. This problem affects event detection methods from social media based on peak detection. We defined a semantic decay filter to identify event-related peaks. The filter was tested on data sets from three different events at different granularities: the STEM school shooting, London bride attacks, and Virginia beach attacks. Despite the small sample size of the number of events, the filter showed remarkable performance in detecting the event-related peaks depicting emergency events. In the future we wish to test the semantic decay filter on other different data sets and different time series data to identify patterns that could identify an event. We also propose to extend the SDF to identify the location of an event at different granularities. Finally, we believe that semantic analysis can also be done to find the type of an event.

PAPER F: REFERENCES

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

[2] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 851–860.

[3] C. Comito, D. Falcone, and D. Talia, "A peak detection method to uncover events from social media," in *Data Science and Advanced Analytics (DSAA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 459–467.

[4] X. Wen and Y.-R. Lin, "Sensing distress following a terrorist event," in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*. Springer, 2016, pp. 377–388.

[5] E. Steiger, J. P. De Albuquerque, and A. Zipf, "An advanced systematic literature review on spatiotemporal analyses of t witter data," *Transactions in GIS*, vol. 19, no. 6, pp. 809–834, 2015.

[6] M. Cordeiro and J. Gama, "Online social networks event detection: a survey," in *Solving Large Scale Learning Tasks. Challenges and Algorithms*. Springer, 2016, pp. 1–41.

[7] M. Garg and M. Kumar, "Review on event detection techniques in social multimedia," *Online Information Review*, vol. 40, no. 3, pp. 347–361, 2016.

[8] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, p. 67, 2015.

[9] O. Ajao, J. Hong, and W. Liu, "A survey of location inference techniques on twitter," *Journal of Information Science*, vol. 41, no. 6, pp. 855–864, 2015.

[10] M. Hasan, M. A. Orgun, and R. Schwitter, "A survey on real-time event detection from the twitter data stream," *Journal of Information Science*, vol. 44, no. 4, pp. 443–463, 2018.

[11] O. Ozdikis, H. Oğuztüzün, and P. Karagoz, "A survey on location estimation techniques for events detected in twitter," *Knowledge and Information Systems*, vol. 52, no. 2, pp. 291–339, 2017.

[12] X. Zheng, J. Han, and A. Sun, "A survey of location prediction on twitter," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1652–1671, 2018.

[13] S. M. Alqhtani, S. Luo, and B. Regan, "Fusing text and image for event detection in twitter," *arXiv preprint arXiv:1503.03920*, 2015.

[14] J. Weng and B.-S. Lee, "Event detection in twitter." *ICWSM*, vol. 11, pp. 401–408, 2011.

[15] N. Alsaedi, P. Burnap, and O. Rana, "Can we predict a riot? disruptive event detection using twitter," *ACM Transactions on Internet Technology (TOIT)*, vol. 17, no. 2, p. 18, 2017.

[16] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, "Tedas: A twitter-based event detection and analysis system," in *Data engineering (icde), 2012 ieee 28th international conference on*. IEEE, 2012, pp. 1273–1276.

[17] N. D. Doulamis, A. D. Doulamis, P. Kokkinos, and E. M. Varvarigos, "Event detection in twitter microblogging," *IEEE transactions on cybernetics*, vol. 46, no. 12, pp. 2810–2824, 2016.

[18] H. Abdelhaq, C. Sengstock, and M. Gertz, "Eventweet: Online localized event detection from twitter," *Proceedings of the VLDB Endowment*, vol. 6, no. 12, pp. 1326–1329, 2013.

[19] J. Krumm and E. Horvitz, "Eyewitness: Identifying local events via space-time signals in twitter feeds," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 20.

[20] R. Lee and K. Sumiya, "Measuring geographical regularities of crowd behaviors for twitter-based geo-social event detection," in *Proceedings of the 2nd ACM SIGSPATIAL international workshop on location based social networks*. ACM, 2010, pp. 1–10.

[21] X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard, "Multiscale event detection in social media," *Data Mining and Knowledge Discovery*, vol. 29, no. 5, pp. 1374–1405, 2015.

[22] M. Quezada, V. Peña-Araya, and B. Poblete, "Location-aware model for news events in social media," in *Proceedings of the 38th International ACM SIGIR*

Conference on Research and Development in Information Retrieval. ACM, 2015, pp. 935–938.

[23] M. Shao, J. Li, F. Chen, H. Huang, S. Zhang, and X. Chen, "An efficient approach to event detection and forecasting in dynamic multivariate social media networks," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1631–1639.

[24] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[25] O. Ozdikis, H. Oguztuzun, and P. Karagoz, "Evidential location estimation for events detected in twitter," in *Proceedings of the 7th Workshop on Geographic Information Retrieval*. ACM, 2013, pp. 9–16.

[26] O. Ozdikis, H. Oğuztüzün, and P. Karagoz, "Evidential estimation of event locations in microblogs using the dempster–shafer theory," *Information Processing & Management*, vol. 52, no. 6, pp. 1227–1246, 2016.

[27] Z. K. Shahraki, A. Fatemi, and H. T. Malazi, "Evidential fine-grained event localization using twitter," *Information Processing & Management*, vol. 56, no. 6, p. 102045, 2019.

[28] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," in *Soviet physics doklady*, vol. 10, 1966, p. 707.

[29] D. Bakkelund, "An lcs-based string metric," *Olso, Norway: University of Oslo*, 2009.

[30] R. W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, vol. 29, no. 2, pp. 147–160, 1950.

[31] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, vol. 7, no. 3, pp. 171–176, 1964.

[32] E. Ukkonen, "Approximate string-matching with q-grams and maximal matches," *Theoretical computer science*, vol. 92, no. 1, pp. 191–211, 1992.

[33] T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons," *Biol. Skr.*, vol. 5, pp. 1–34, 1948.

[34] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.

[35] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[36] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[38] G. E. Dahl, R. P. Adams, and H. Larochelle, "Training restricted boltzmann machines on word observations," *arXiv preprint arXiv:1202.5695*, 2012.

[39] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in neural information processing systems*, 2009, pp. 1607–1614.

[40] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–43.

[41] A. Cherniak and V. I. Zadorozhny, "Signature-based detection of notable transitions in numeric data streams," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2867–2879, 2013.

[42] M. B. Lazreg, M. Goodwin, and O.-C. Granmo, "Combining a context aware neural network with a denoising autoencoder for measuring string similarities," *Computer Speech & Language*, p. 101028, 2019.

[43] J. Gentry, "Twitter package, http://lists.hexdump.org/listinfo.cgi/twitter-users-hexdump.org.