SPATIOTEMPORAL DATA MINING FOR SITUATION AWARENESS IN MICROBLOGS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

ÖZER ÖZDİKİŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

JUNE 2016

Approval of the thesis:

SPATIOTEMPORAL DATA MINING FOR SITUATION AWARENESS IN MICROBLOGS

submitted by **ÖZER ÖZDİKİŞ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Pınar Karagöz _____
Supervisor, **Computer Engineering Dept., METU**

Prof. Dr. Halit Oğuztüzün _____
Co-supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering Dept., METU _____

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Dept., METU _____

Prof. Dr. Fazlı Can
Computer Engineering Dept., Bilkent University _____

Prof. Dr. Ferda Nur Alpaslan
Computer Engineering Dept., METU _____

Assist. Prof. Dr. Mehmet Tan
Computer Engineering Dept., TOBB ETÜ _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: ÖZER ÖZDİKİŞ

Signature :

ABSTRACT

SPATIOTEMPORAL DATA MINING FOR SITUATION AWARENESS IN MICROBLOGS

ÖZDİKİŞ, ÖZER

Ph.D., Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Pınar Karagöz

Co-Supervisor : Prof. Dr. Halit Oğuztüzün

June 2016, 171 pages

Detection of real-world events using messages posted in microblogs has been the motivation of numerous recent studies. In this thesis, we study spatiotemporal data mining techniques to improve situation awareness by detecting events and estimating their locations using the content in microblogs, particularly in Twitter. We present an enhancement to the clustering techniques in the literature by measuring associations between terms in tweets in a temporal context and using these associations in a vector expansion process to improve the accuracy of online tweet clustering and event detection. Moreover, we propose a method using the Dempster-Shafer theory to estimate the locations of the detected events. We utilize three basic location-related features in tweets, namely the latitude-longitude metadata in geotagged tweets, the location names mentioned in the tweet content and the location attribute in the user profile, as independent sources of evidence. We apply combination rules in the Dempster-

Shafer theory to fuse them into a single model, and estimate the whereabouts of a detected event. We demonstrate the results of our experiments for event detection and location estimation using public tweets posted in Turkey. Our experiments indicate higher success rates than those obtained by the state of the art methods.

Keywords: Event Detection, Location Estimation, Microblogs, Dempster-Shafer Theory, Statistical Text Analysis

ÖZ

MİKROBLOGLARDA DURUM FARKINDALIĞI İÇİN LOKASYON VE ZAMAN TABANLI VERİ MADENCİLİĞİ

ÖZDİKİŞ, ÖZER

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Assoc. Prof. Dr. Pınar Karagöz

Ortak Tez Yöneticisi : Prof. Dr. Halit Oğuztüzün

Haziran 2016 , 171 sayfa

Mikrobloglarda yazılan mesajları kullanarak gerçek hayatta yaşanan olayların bulunması pek çok güncel çalışmanın konusu olmuştur. Bu tez çalışmasında, mikrobloglarda, özellikle de Twitter platformunda oluşturulan içeriği kullanarak olayları bulup yerlerini tahmin ederek durum farkındalığını artıran lokasyon ve zaman tabanlı veri madenciliği teknikleri araştırılmıştır. Çevrimiçi kümeleme ve olay bulma işlemlerinin doğruluk derecesini artırmak amacıyla, tweet'lerde geçen kelimeler arasındaki ilişkileri zamansal bir bağlamda ölçen ve bu ilişkileri vektör genişletme işleminde kullanan bir yöntem sunulmaktadır. Ayrıca bulunan olayların yerlerinin tahmin edilmesi için Dempster-Shafer teorisinin uygulandığı bir yöntem önerilmektedir. Bu yöntemde, coğrafi etiketi bulunan tweet'lerdeki enlem-boylam bilgisi, tweet içeriğinde geçen yer isimleri ve kullanıcı profilinde belirtilen lokasyon alanı birbirinden bağımsız üç farklı bilgi kaynağı olarak kulla-

nılmış; ve Dempster-Shafer teorisindeki kombinasyon kuralları ile tek bir model halinde birleştirilerek ilgili olayın konumu tahmin edilmiştir. Olay bulma ve yer tahmini için önerilen çözümler Türkiye içinden gönderilen tweet'ler üzerinde uygulanarak, elde edilen sonuçlar gösterilmiştir. Yapılan deneyler, literatürdeki mevcut çözümler ile elde edilenden daha yüksek bir başarı oranına işaret etmektedir.

Anahtar Kelimeler: Olay Bulma, Lokasyon Tahmini, Mikrobloglar, Dempster-Shafer Teorisi, İstatistiksel Metin Analizi

To my parents

ACKNOWLEDGMENTS

First and foremost I am greatly thankful to my supervisors Dr. Pınar Karagöz and Prof. Halit Oğuztüzün for their support, guidance, suggestions, and for motivating me throughout my PhD studies.

I would like to show my gratitude to the members of my thesis monitoring committee, Prof. Nihan Kesim Çiçekli and Prof. Fazlı Can, for their valuable comments and feedback on my thesis work. I am also grateful to defense jury members, Prof. Ferda Nur Alpaslan and Dr. Mehmet Tan, for reviewing and evaluating my thesis.

I take this opportunity to record my gratitude to Scientific and Technological Research Council of Turkey (TÜBİTAK) (program 1001, grant number 112E275) and ICT COST Action (IC1203) for supporting this research.

I would like to thank my parents, my mother Nalan Özdikiş and my father Muhsin Özdikiş, for their love and endless support during my education. Last but not the least, I thank all my friends for their encouragement and patience.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xvi

LIST OF FIGURES xviii

LIST OF ABBREVIATIONS xx

CHAPTERS

1 INTRODUCTION 1

1.1 Overview 1

1.2 Motivation 2

1.3 Contributions 6

1.4 Organization of the Thesis 8

2 BACKGROUND 11

2.1 Social Media, Microblogs, and Twitter 11

2.1.1 Data Collection from Twitter 12
2.1.2 Data Types in Twitter 13
2.2 Statistical Relationships of Words 15
2.3 Dempster-Shafer Theory 17
3 RELATED WORK 21
3.1 Event Detection in Twitter 21
3.2 Incremental Clustering Approaches 23
3.3 Similarity Analsysis and Vector Expansion 25
3.4 Geospatial Analysis in Social Networks 27
3.5 Applications of Dempster-Shafer Theory 30
4 ONLINE EVENT DETECTION IN TWITTER 33
4.1 Data Collection and Modeling 34
4.2 Clustering 36
4.2.1 IC: Incremental Clustering 36
4.2.2 ICVE: Incremental Clustering with Vector Expansion 37
4.2.2.1 Identifying Terms to Compare . . . 38
4.2.2.2 Finding Co-occurrence Vectors . . . 39
4.2.2.3 Calculation of Term Similarities . . 42
4.2.2.4 Vector Expansion 42
4.3 Burst Detection 43

4.3.1 Detection of bursty terms in clusters 44
4.3.2 Extraction of Event Features 45
4.3.3 Assigning tweets to events 46
5 EVALUATION OF INCREMENTAL CLUSTERING WITH VECTOR EXPANSION 47
5.1 Evaluation Setting 47
5.2 Ground Truth Annotation 48
5.3 Analysis of Clustering Accuracy 49
5.4 Implications of Clustering on Event Detection Accuracy 55
6 LOCATION ESTIMATION TECHNIQUES FOR EVENTS DETECTED IN TWITTER 59
6.1 Event Types and Location Granularities 59
6.1.1 Event Characteristics in Localization 60
6.1.2 Granularity of Estimated Locations 61
6.2 Spatial Features for Location Estimation 63
6.2.1 Analysis of Spatial Features in Tweets 64
6.2.2 Usage of Spatial Features in Tweets 70
6.3 Location Estimation Methods 73
6.3.1 Event-Pivot Methods 73
6.3.1.1 Basic Statistics 74
6.3.1.2 Spatial Clustering 76
6.3.1.3 Probabilistic Approaches 78

6.3.1.4 Bayesian Filters 79

6.3.2 Location-Pivot Methods 80

6.3.2.1 Location-Oriented Tweet Collection 81

6.3.2.2 Activity Analysis in Partitioned Regions 83

6.3.2.3 Burst Detection 85

6.3.2.4 Spatial Clustering 86

6.4 Evaluation Metrics 88

6.5 Human-Computer Interaction 90

6.6 Discussion 92

7 EVIDENTIAL LOCATION ESTIMATION FOR EVENTS DETECTED IN TWITTER 95

7.1 Spatial Information for Location Estimation 95

7.2 Location Mapping Functions 96

7.3 Basic Probability Assignments for Locations 100

7.4 Combination of BPAs 101

7.5 Location Selection 101

7.6 Association Of Evidence 102

7.7 Normalization of GPS 103

7.8 Graphical Presentation for the Combined Evidence . . . 105

8 EVALUATION OF THE PROPOSED LOCATION ESTIMATION SOLUTION 107

8.1 Evaluation Setting . . . 107
8.2 Ground Truth Annotation . . . 108
8.3 Evaluation Metrics . . . 109
8.4 Evaluation of Combination Methods . . . 110
8.5 Evaluation of City-Town Association . . . 115
8.6 Comparison with Baseline Methods in the Literature . . . 117
8.6.1 Baseline Methods . . . 117
8.6.2 Evaluation Results for Baselines . . . 121
8.6.3 Evaluations on Event Categories . . . 124
8.7 Evaluation of GPS Normalization . . . 128
8.8 Analysis of Earthquakes . . . 130
8.9 Graphical Presentation for the Combined Evidence . . . 132
8.10 Limitations of the Proposed Methods . . . 134
9 CONCLUSION AND FUTURE RESEARCH . . . 137
REFERENCES . . . 143
APPENDICES . . . 157
A CLUSTERING EVALUATION RESULTS . . . 157
B LOCATION ESTIMATION EVALUATION RESULTS . . . 159
CURRICULUM VITAE . . . 169

LIST OF TABLES

TABLES

Table 5.1 Constants and Thresholds in Experiments 48

Table 5.2 Cluster-target event contingency table 50

Table 5.3 Detected similarities for the term "Sneijder" using SO, SSO and SVD 52

Table 5.4 Precision, Recall and F_1-scores per each event, using the baseline IC and the enhanced ICVE-SO algorithms 54

Table 6.1 Targeted Event Types and Granularity of Estimated Locations 63

Table 6.2 Advantages and Challenges of Spatial Features in Tweets . . . 66

Table 6.3 Usage of Spatial Features (P):Primary, (S):Secondary. 71

Table 6.4 Event-Pivot Techniques 75

Table 6.5 Location-Pivot Techniques 82

Table 8.1 Categories of Detected Events in Ground Truth 109

Table 8.2 Average error distances (in kilometers) for different sized data sets 111

Table 8.3 Average error distances (in kilometers) using city-town association 116

Table 8.4 Average error distances per event type using different estimation methods (in kilometers) 125

Table 8.5 Average error distances per event type using GPS-normalization (in kilometers) 129

Table A.1 Precision, Recall and F_1-scores using IC and the three settings of ICVE 158

Table B.1 City-level average error distances using baseline estimation methods (in kilometers) 160

Table B.2 City-level match rates using baseline estimation methods . . . 161

Table B.3 Town-level average error distances using baseline estimation methods (in kilometers) 162

Table B.4 Town-level match rates using baseline estimation methods . . 163

Table B.5 City-level average error distances using the proposed DS methods (in kilometers) 164

Table B.6 City-level match rates using the proposed DS methods 165

Table B.7 Town-level average error distances using the proposed DS methods (in kilometers) 166

Table B.8 Town-level match rates using the proposed DS methods 167

LIST OF FIGURES

FIGURES

Figure 4.1 Online Event Detection Stages 33

Figure 4.2 Proposed Online Event Detection Process using Vector Expansion 36

Figure 5.1 Cluster accuracies for IC, ICVE-SO, ICVE-SSO and ICVE-SVD using different merge threshold values 51

Figure 5.2 Execution times of clustering algorithms 53

Figure 5.3 Event Detection Accuracy Analysis on Annotated Events . . 56

Figure 5.4 Distribution of the term "gol" in tweet stream and clusters before and after the goal event 57

Figure 8.1 Match rates and average error distances at city level 121

Figure 8.2 Match rates and average error distances at town level 123

Figure 8.3 (a) Average error distances for earthquakes using different location estimation methods, (b) Distances between the epicenters of four sample earthquakes and the GPS coordinates of the corresponding tweets 131

Figure 8.4 City level estimations for an earthquake in the town Gaziemir,Izmir on 26 May 2013, at 8:31am (3.5 M_L in magnitude). 132

Figure 8.5 Town level estimations for the earthquake in Gaziemir,Izmir. (a): DP, (b): NormDP, (c): NormADP 133

Figure 8.6 People traveling from Izmir to Istanbul to join the Gezi Park protests on 31 May 2013. 134

LIST OF ABBREVIATIONS

API	Application Programming Interface
BPA	Basic Probability Assignment
DP	Dubois and Prade
DRC	Dempster's Rule of Combination
DS	Dempster-Shafer
EM	Expectation Maximization
GPS	Global Positioning System
IC	Incremental Clustering
ICVE	Incremental Clustering with Vector Expansion
LSA	Latent Semantic Analysis
NB	Naïve Bayes
NER	Named Entity Recognition
NLP	Natural Language Processing
PCA	Principal Component Analysis
POI	Point of Interest
POS	Part-of-Speech
SIS	Sequential Importance Sampling
SVD	Singular Value Decomposition
TDT	Topic Detection and Tracking
VGI	Volunteered Geographic Information
YR	Yager's Rule

CHAPTER 1

INTRODUCTION

1.1 Overview

Social networks, particularly microblogs, are increasingly used as communication platforms by millions of people worldwide. The availability of Internet access and affordable smart phones allow people to easily access online services to share photos, videos, and text messages with others in their network. With its innovative microblogging concept, Twitter is among the most popular social networking platforms enabling users to post 140 character text messages as tweets, and share them with their followers. Since real world events usually have a direct impact on the content of tweets, it is a useful medium to learn about the latest news, follow popular events, and keep up with hot topics [61].

In this thesis, we aim to detect events using tweets and estimate the locations of the detected events accurately with no prior assumption about their topics. An important novelty in the thesis is the application of evidential reasoning techniques, namely the Dempster-Shafer (DS) theory, for the location estimation problem [36, 37, 110]. We utilize three basic location-related features in tweets, namely the latitude-longitude metadata provided by the GPS sensor of the user's device, the textual content of the post, and the location attribute in the user profile, as three independent sources of evidence. Considering this evidence in a complementary way, we apply combination rules in the DS theory to fuse them into a single model, and estimate the whereabouts of a detected event.

Locations are treated at two levels of granularity, namely, city and town. Using the DS theory to solve this problem allows uncertainty and missing data to be tolerated, and estimations to be made for sets of locations in terms of upper and lower probabilities. We use Twitter as a representative of microblogs, due to its widespread usage and easily accessible public data. However, we believe our proposed solution is also adaptable to other microblogs.

The location estimation method proposed in this thesis requires that a group of tweets posted about an event has been identified. One approach for collecting tweets about an event is to use specific query terms [82, 108]. However, this may be useful when the user knows what to search for. If the facts are not known in advance, it is not realistic to expect a user to define a query [136]. An alternative approach is to collect random samples of public tweets provided by Twitter and cluster them according to their content similarity, which can detect various types of events without any restriction on the topic [109, 137]. In this thesis, we adopt the latter approach and use an incremental clustering method to identify events and event-related tweets becasue of the effective processing of online tweet stream without any prior selection of event-specific keywords. Since identifying event-related tweets is an important component in the overall process, we also study incremental clustering methods in the literature and propose a method to enhance their accuracy using vector expansion.

1.2 Motivation

Data mining techniques on content-rich textual documents have long been studied as a part of the field of Topic Detection and Tracking (TDT) [11, 43]. The increasing usage of Twitter as a communication platform has led to TDT techniques that were previously applied for newspaper articles and blog posts being extended and adapted to perform event detection using tweets. In these studies, an event is defined as an activity that happens at a specific time and place. Event detection is an important objective in TDT studies, which aims to ana-

lyze texts automatically, generate summaries for people, and enhance situation awareness.

Automatic event detection techniques in social networks have been employed to detect various types of events, such as mass emergencies [60], earthquakes [90, 108], landslides [84], disasters and crises [79, 137], sports events [1], epidemics and diseases [2, 94], political topics [104], crimes, accidents and public unrest in early stages [74, 123]. These efforts usually aim to provide curated information for the public, help decision makers and local authorities take timely response actions in case of emergencies, and alert journalists and community about the latest news. Detection of such events and estimating their locations significantly improve the level of public awareness and provide the means to perform a visual analysis on real-world activities [131].

Clustering documents according to their semantic similarity around topics is a widely adopted approach to detect events [109, 113, 137]. In order to achieve effective clustering in Twitter one of the basic challenges to be overcome is related to content. Twitter has distinct characteristics that differentiate it from text in newspapers and blogs [66]. The limitation of the tweet length to 140 characters, and idiosyncratic spellings due to uncontrolled and spontaneously generated content are two major reasons for the existing methods to be enhanced. People make spelling mistakes, follow non-traditional writing conventions, and abbreviate long words because of the character limitation. Even when they refer to the same reality, they may express it in many different ways. Therefore, we claim that, clustering performance can be improved if similar terms in tweets are discovered and utilized in a vector expansion process.

The content in Twitter is highly driven by the community rather than being based on a thesaurus such as WordNet[1] or Wikipedia[2], therefore a static thesaurus cannot effectively cover this user generated content [8, 127]. Moreover, the similarity of two terms may change depending on the time and context. As a

[1] https://wordnet.princeton.edu [accessed 01 June 2016]
[2] https://www.wikipedia.org [accessed 01 June 2016]

consequence, we propose a method to identify similar terms using co-occurrence based statistics on the tweet content and score term similarities online within a time window.

A clustered set of tweets may not necessarily indicate a new event in Twitter. A noticeable number of tweets have been reported not to be related to a real-world event but are concerned with the ordinary daily activities of people in the form of chat, personal updates, conversations, and spam [97, 133]. Burst detection can be applied to distinguish event-triggered patterns in tweet traffic by analyzing temporal frequency distributions in specific features of tweets (e.g., frequency of a term, number of tweets per unit time) [46, 72, 134]. Therefore, as similar tweets are grouped around topics by clustering, we also apply a burst detection method to detect new events by analyzing surges in term frequencies in near real time. We aim to show that the proposed enhancement using online similarity analysis of terms and using them in vector expansion yields more coherent clusters and helps accurate identification of event-related tweets.

Tweets can contain geographical footprints, also referred to as ambient geographical information [27, 79, 117]. The most explicit and precise location can be obtained from the *geotagged tweets*, i.e., tweets with geographical coordinates in terms of latitude and longitude. Another source of geographical evidence is the *tweet text* itself, which requires the identification of location names contained in a tweet. Additionally, the location attribute in the *user profile* allows Twitter users to specify their home location in a free-form text field. Despite these various sources of spatial information in tweets, each attribute poses different challenges to be useful in location estimation. For example, although GPS coordinates provide a precise geographic position on earth in terms of latitude and longitude, this location and that of the event mentioned in the tweet may not be the same. Once an event is broadcast in the media or shared among people via phone calls or retweets, the GPS coordinates of the recently posted tweets may quickly spread across locations that are further away, making the event location more difficult to detect [31, 108]. Challenges related to the tweet con-

tent and location attribute of user profiles mostly concern text processing and geoparsing, i.e., relating a given text to spatial locations [54]. These attributes are uncontrolled free-text fields. Their quality of content is not as good as that contained in news articles due to the idiosyncratic spellings, unusual writing conventions and abbreviations. Therefore, state of the art Natural Language Processing (NLP) tools do not perform as accurately on tweets [76]. In addition to these challenges, users do not have to reveal their GPS locations, or their city of residence in their profiles. They also do not have to mention any location name in their tweets. As a result, uncertainty and a lack of rich and reliable data is a major common problem to be overcome.

Geographical traces in user generated content in social networks have been utilized to solve a variety of spatial analysis problems. Some studies aim to infer the location where a photo or tweet has been posted even if the user did not share the GPS data of the mobile device [70, 125, 132]. Similarly, there are efforts to assign geographical coordinates to textual resources, such as Wikipedia articles [126]. Estimating locations of users by utilizing the content in social networks is another active research area [25, 28]. In [118], the authors reviewed the literature and suggested that the spatiotemporal analysis of tweets is a promising but still an underexplored field for researchers. In this thesis, we focus on estimating locations of events detected in microblogs. Given a set of tweets that are presumably about an event, we aim to estimate the location of this event by exploiting the available evidence in the tweets and in the profiles of the users who posted the tweets. This problem is also referred to as *event localization* in [48, 49].

In this thesis, the proposed method to the event localization problem is the DS theory. The DS theory is a generalized form of Bayesian inference based on existing evidence. Considering three tweet features as three separate information sources, it provides means to define belief intervals for sets of possible discrete locations and various rules to combine them into a single model. It enables the represention of indifference, which is, in our opinion, very suitable to analyze

tweets since not all tweets can be expected to provide spatial information in all of their location-related attributes. We demonstrate the results of the proposed solution using public tweets posted in Turkey. The experimental evaluations conducted on a wide range of events including earthquakes, sports, weather, and street protests indicate higher success rates than the existing state of the art methods.

1.3 Contributions

The primary goal in this thesis is to accurately estimate event locations detected in microblogs. Moreover, since the proposed solution requires a set of event-related tweets in order to estimate event location, we also propose an enhancement to the state of the art incremental clustering methods to improve the event detection and clustering accuracy. Therefore, we consider the contribution in this thesis in two parts.

In the first part, the contributions of the proposed solution to enhance the performance of incremental clustering can be summarized as follows:

- We leverage online incremental clustering methods by automatically extracting and scoring term similarities in a temporal locality to be used in a vector expansion process, which we call *Incremental Clustering with Vector Expansion (ICVE)*.
- Our methods are unsupervised and do not rely on an existing thesaurus. We extract and utilize term similarities using statistical methods, therefore they can be applied to any language. We make no *a priori* assumption about the number of clusters or their topics.
- The proposed solution can be efficiently executed online, making a single pass on the incoming tweet stream without any post-processing. We consider clusters as evolving topics containing zero or more newsworthy events.

The second part is related to the localization of detected events. Before presenting the details of the proposed location estimation solution using the DS theory, we review the state of the art event locatization techniques for microblogs in the literature. We analyze these techniques with respect to the targeted event type, granularity of estimated locations, location-related features selected as sources of spatial evidence, and the method used to make aggregate decisions based on the extracted evidence. We discuss the strengths and advantages of alternative solutions to various problems related to location estimation, as well as their preconditions and limitations.

Then we describe our proposed location estimation using the DS theory. The contributions of our location estimation solution can be summarized as follows:

- The problem of location estimation for events is investigated using the DS theory, which allows us to use the existing evidence pertaining to the location of event in a complementary way and extract belief intervals for the candidate locations.
- The proposed method is not specific to the event type thus, it does not require any prior event annotation for training. It is experimentally evaluated on a set of tweets posted in Turkey about events of different types, including concerts, sports, street protests, accidents, and earthquakes. The results show that the proposed method can estimate the location of events with higher accuracy in comparison to the existing state of the art methods.
- Estimations are made for locations at multiple granularities, namely at the city and town levels. Accordingly, we define an association of evidence between coarse-grained and fine-grained data based on the mixed class hypothesis in the DS theory.
- We demonstrate that the contribution of each attribute in the location estimation problem may change temporally depending on the event type. For some events, GPS coordinates are very reliable for the first few tweets,

but they diffuse over time as more tweets are received. For some other types of events, the location references in the tweet content turns out to be more accurate source of evidence over time.

- Since the DS theory yields probability intervals for each discrete geographical entity in the domain, all the locations related with a given event are marked accordingly on a map. This view offers an intuitive graphic representation of the geography of the event.

We would like to note that except for the stemming library in Turkish [23], the online event detection methods presented in this thesis do not depend on the language. Moreover, although we evaluate the proposed methods using tweets collected by the Streaming API of Twitter[3] using the geographical coordinates of Turkey as the filtering criteria, by adjusting the boundary coordinates of this filter, these methods can be applied to other countries and regions as well.

1.4 Organization of the Thesis

This thesis presents a solution to improve the accuracy of online tweet and event detection, and another solution to estimate the locations for the detected events using clustered tweets. These solutions are presented in two parts. Accordingly, the remainder of this thesis is organized as follows:

Chapter 2 covers background information about microblogs, focusing mostly on Twitter. It describes basic data collection methods and types of data that can be collected from Twitter. Background information related to word co-occurrences and to the DS theory that are often referred in the remainder of the thesis are presented.

Chapter 3 reviews the current state of the art on clustering and event detection methods in microblogs, which is followed by efforts that aim to handle spelling variances in documents, particularly the ones that adopt term-level similarity

[3] https://dev.twitter.com/streaming/overview [accessed 01 June 2016]

analysis and vector expansion. We discuss the existing research that performs geospatial analysis in social networks and that estimates event locations in microblogs. Finally, we address several applications that previously used DS theory to solve various problems.

Chapter 4 and Chapter 5 are devoted to the enhancement that we propose for online event detection and its evaluation, respectively.

Before introducing the proposed event localization method based on DS theory, a detailed analysis of the literature on event localization methods in microblogs is presented in Chapter 6. This chapter illustrates the types of events that have been localized, including a categorization of their spatial granularity, analyzes spatial features in tweets together with their advantages and challenges, and introduces a classification of existing location estimation techniques explaining their essential characteristics, strengths and limitations.

Chapter 7 describes the proposed evidential location estimation method for the events detected in Twitter. The solution applies DS theory principles to the event localization problem. We explain different settings to improve the accuracy of the estimation.

Chapter 8 gives the results of the experiments using the proposed location estimation method and presents comparisons with several baseline methods. Effect of different settings in the solution are discussed and their performance for specific event types are analyzed.

Chapter 9 concludes the thesis by a summary of our findings, suggestions for further research, and concluding remarks.

CHAPTER 2

BACKGROUND

In this chapter, we give background information about the most widely used concepts in the remainder of the thesis. Section 2.1 presents an overview about microblogs, particularly about Twitter and its content. Section 2.2 addresses the types of co-occurrence based relationships between terms in a document collection. Section 2.3 explains the fundemantal concepts and three combination rules in DS theory.

2.1 Social Media, Microblogs, and Twitter

Social media services encompass a variety of platforms where people can connect with each other, and publish and share content such as text messages, photos, videos, or location information within their network [60, 65]. Microblogging platforms, such as Twitter, Tumblr, Weibo, constitute an important part of social media services acting as message broadcasting platforms that provide the means of information sharing, leading to the concept of citizen journalism [34, 69]. In addition to microblogs, social media services include other application domains, which can be listed as social networking (e.g., Facebook), multimedia sharing (e.g., Flickr, Instagram, YouTube), and location check-in services (e.g., Foursquare) [106, 119, 124]. The vast amount of user-generated content in these platforms has been an attractive resource for researchers. Numerous studies have utilized the content in microblogs to detect real-world events [15, 118]. Event

detection using the tags supplied by users to annotate their photos in Flickr has been studied in [26]. The correlation between traffic congestion and locations of the user check-ins at Foursquare and Instagram has been analyzed in [103]. In [95], the authors combined Twitter and Flickr data using the tweet content to locate events and analyzing Flickr images to delineate the impact area of these events. Social media is even integrated with other information sources such as news channels, knowledge bases, as well as physical sensors in order to detect events and increase situation awareness [57, 63, 84].

Since the content in different social media platforms might have common properties and similar attributes, such as user profiles or geotagged items, a solution applied to a problem in one platform can also be applicable to another. However, at the same time, each platform poses its own challenges because they provide their content in diverse structures, formats and qualities [95]. Additionally, the number of contributors concerned with an event is not the same in all platforms. According to our observation, the majority of studies related to information extraction in social media are based on the content in microblogs, particularly in Twitter. The reasons for this use of Twitter can be listed as its widespread adoption, brief textual content, and non-reciprocal and asymmetric nature (a user can follow any other user without being followed back) as well as the easy distribution of messages via retweeting, the responsiveness of tweets, and the accessibility of Twitter data through its public APIs [69]. Therefore, in this thesis, we also focus on the location estimation methods for events detected in Twitter.

2.1.1 Data Collection from Twitter

Twitter provides two types of APIs[1] that support collecting tweets programmatically, namely the Search (Query) API and the Streaming API [60, 89]. The Search API allows developers to retrospectively search for tweets by specifying a set of query criteria, such as a time period, expected words in the tweet content,

[1] https://dev.twitter.com/overview/api [accessed 01 June 2016]

a list of users to follow, or tweet locations described by bounding boxes in the form of latitude-longitude coordinates. The Search API functions in a request-response manner returning the list of requested objects that satisfy the criteria at the time of the query. The most important disadvantage of using this API is the limitation on the number of queries that can be made per unit time, where each query returns a limited number of tweets. It also requires executing the queries periodically in order to follow the most recent tweets posted in Twitter.

The Streaming API, on the other hand, is designed to operate in a more online fashion. A client application can collect tweets from an online tweet stream by establishing a connection to the Twitter service. As long as the connection is alive, Twitter keeps sending a random sample of tweets in a callback mechanism. Additionally, similar to the Search API, in the Streaming API, it is possible to define criteria for the tweets to be received from the stream. Challenges related to searching and collecting tweets from Twitter, together with a comprehensive description of systems, tools, and libraries are given in [89].

2.1.2 Data Types in Twitter

Data retrieved from Twitter consists of four main types of objects, namely tweets, users, places, and entities. The attributes of a tweet object include a unique id, tweet text (content), creation time, and coordinates in terms of latitude-longitude assigned by the client at the time of the post. The user object associated with a tweet describes the properties of the user who posted the tweet, such as his unique id, name, location, language, and time zone. The attributes of users are kept in user profiles. Place objects represent the location definitions in Twitter, described by a unique id, name, and an array of latitude-longitude pairs corresponding to the location's boundary coordinates. Twitter allows users to attach place objects to their tweets. Furthermore, entity objects in the form of hashtags, user mentions, and URLs are used to provide additional information for the associated tweet [58, 69]. A hashtag is a word in a tweet preceded by a # character, which is used as a convention to annotate tweets

around a thread of discussion. A user can be addressed as a mention in a tweet by typing @ before the user identifier. It is also possible to re-post someone else's tweet as a retweet.

In this thesis, in addition to the time of the post, we use various tweet attributes to detect events and estimate their locations. These attributes and their most important properties are listed as follows:

- ***Tweet Text:*** Tweet text is a free-text field containing up to 140 characters. It presumably constitutes the most important feature to understand the topic of the tweet and extract information about an event. Since locations mentioned in a tweet can be related to an event described in that tweet, message-mentioned locations identified in the tweet are also widely used for event localization [123]. However, the task of textual analysis can become complicated due to factors such as the brevity of content, spelling idiosyncrasies, grammatical errors, and abbreviations that may not exist in any vocabulary.

  In order to surpass the 140-character limit, tiny URLs have been created to shorten long URLs to texts with 10-20 characters. These links can be easily resolved to the original URLs of the external web pages and allow people to link their posts with web resources with larger content [109]. A Twitter user can also configure his account to work in integration with his accounts in other social networks. For example, a check-in posted via Foursquare[2] can automatically be displayed in a user's tweet in a specific format; e.g., "I am at *TimesSquare* w/ 2 others", which may ease the identification of place names in tweet texts [132].

- ***GPS Geotag:*** Coordinates in terms of latitude and longitude can be associated with a tweet at the time of posting if supported by the user's GPS-enabled device and its software. However, even if the hardware and software support sharing this data, users can disable this feature and stop

[2] https://foursquare.com [accessed 01 June 2016]

sharing the coordinates in their tweets. Tweets associated with this information are referred as GPS geotagged, or simply, geotagged [60]. Although geotagged tweets are reported to be very rare in Twitter (around 2-3%), their ratio in a corpus is also affected by the selected tweet collection method [48]. For example, using geographical filters, it is possible to obtain high ratios of geotagged tweets.

- ***User Profile:*** Twitter users can specify their home location in their profiles with a text of up to 30 characters. Since the location in the user profile is a free-text field, it is possible to find unclear location references, multiple location names, or even fake locations and sarcastic comments [28, 52]. In [52], it is reported that 66% of the profiles contained valid geographical information, 16% had non-geographic texts, and the remaining 18% did not specify anything related to the user's location. Moreover, even if the location information is provided in the profile, most people do not update it every time they relocate. Thus, the spatial evidence obtained from a user's profile may not reflect the most recent location of that user [31].

In addition to the three attributes, other types of data, such as the preferred language and time zone specified in the user profile can provide further spatial evidence. However, these attributes are not as useful since they provide relatively more coarse-grained information covering very large geographical scopes [9, 82].

2.2 Statistical Relationships of Words

Since tweets are maximum 140 characters long and there are a lot of spelling differences, in order to effectively find the similarities in tweets, we propose to enrich tweet texts using term-level similarities before analyzing tweet-level similarities for clustering. Similarities between terms can be obtained from online thesauri, such as WordNet or Wikipedia. However, such online resources may not be mature for all languages. More importantly, the content in Twitter is highly driven by the community rather than being based on a thesaurus. There-

fore, different writing conventions and violations of spelling or grammatical rules particularly make it difficult to benefit these resources. Moreover, similarities between two terms may also change depending on the context of usage. As a result, we investigate statistical methods based on word co-occurrences to extract term similarities.

Statistical relationships between term pairs are classified as first-order, second-order, and higher-order relationships [24, 93, 101, 102]. These relationships can be described as follows:

- ***First-order:*** A strong first-order relationship is observed when two terms appear frequently together in texts [93, 102]. This relationship is also known as syntagmatic relationship. People's first and last names, or word pairs such as *birthday-party* can be considered to exhibit this kind of association.

- ***Second-order:*** Two terms have a second-order relationship if they frequently appear together with the same set of terms, i.e., having similar lexical neighborhood. Therefore, methods that aim to find terms with second-order relations in text consider the mutuality of co-occurrences with other words. For example, *photo-photograph* or *black-white* are such word pairs that most probably co-occur with the same words. Second-order associations are also referred to as paradigmatic relations. Two terms that frequently co-occur with the same set of other terms are expected to be used interchangeably, possibly changing the meaning of the sentence, but without affecting the structure and grammar [102]. Hence, in this thesis, we focus mostly on the second-order relationships and use them for tweet enrichment.

- ***Higher-order:*** A similar logic can be applied to obtain terms with higher-order relationship. For example, if the number of co-occurrences of the term pairs t_1-t_2, t_2-t_3 and t_3-t_4 are relatively high, then t_1 and t_4 can be considered to have a third-order association [93]. Higher-order relations

can be measured by plotting the terms as nodes on an undirected graph and connecting them with edges if they co-occur in documents [24]. The number of distinct paths of length n between two nodes gives a score for the n^{th} order relationship between them.

2.3 Dempster-Shafer Theory

Arthur P. Dempster [36, 37] introduced a generalized Bayesian inference model based on evidence and evidential reasoning which was further extended by the work of Glenn Shafer [110] and is known as the Dempster-Shafer (DS) theory. The first step in applying this theory to a problem is defining the propositional space of possible solutions, the *Frame of Discernment*, denoted by Θ. For the targeted location estimation problem, this set is composed of all the locations (e.g., cities, towns) where an event might have happened.

The set of all subsets of Θ are denoted by 2^{Θ}. Applying DS theory to a problem necessitates the definition of *Basic Probability Assignments* (BPAs), which are basic probability numbers assigned to the elements in 2^{Θ} according to the evidence obtained from the information sources. BPAs are represented by *mass functions* m, where $m : 2^{\Theta} \rightarrow [0, 1]$, $m(\emptyset) = 0$ and $\sum_{A \subseteq \Theta} m(A) = 1$. The mass value $m(A)$ is A's *basic probability number* (or *probability mass*), and it measures the belief that is committed exactly to A, according to the evidence obtained from a data source. The subsets of Θ with non-zero mass assignments are called *focal elements*, and the union of focal elements is termed the *core* of a mass function. Assigning $m(\emptyset) = 0$ means a "closed world assumption", i.e., there has to be a solution to the problem in Θ [114]. Since the evidence can support multiple possible explanations of the problem, probability numbers can be assigned for sets of elements in Θ. This is called the *mixed class hypothesis* in DS theory and it is particularly useful for the described event localization problem, since a tweet may refer to multiple locations, or users may type multiple location names in their profiles.

In DS theory, the *belief function* $Bel(A)$ is defined as given in Equation 2.1. Unlike a probability function in probability theory, the belief function in DS theory is generally not additive, i.e., $Bel(A \cup B) \geq Bel(A) + Bel(B)$. A belief function is also known as a *support function*, since it indicates the degree to which the evidence supports that proposition [78].

$$Bel(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B) \tag{2.1}$$

DS theory also provides the means to evaluate the credibility of a proposition. It is defined by the *plausibility function* $Pl(A)$, given in Equation 2.2. It represents the degree to which the evidence fails to refute a proposition [78]. The belief and plausibility values for a proposition are also interpreted as its lower and upper probabilities. In other words, $Bel(A)$ represents the currently available evidence in the environment that supports that a solution is in A. As more evidence supporting A is received, this value approaches $Pl(A)$.

$$Pl(A) = 1 - Bel(\overline{A}) = \sum_{B \subseteq \Theta, A \cap B \neq \emptyset} m(B) \tag{2.2}$$

The total probability mass that can move freely to every point of A is measured by the *commonality function* in Equation 2.3. More specifically, given two sets of elements A and B in Θ where $A \subset B$, the mass assigned to B could move into A if more evidence was provided to support it, but the current evidence is not sufficient to assign it more precisely than to B. The commonality function in Equation 2.3 measures the total mass that can move to A.

$$Q(A) = \sum_{A \subseteq B} m(B) \tag{2.3}$$

While the Bayesian statistics represent the full ignorance as a uniform distribution and assign a probability to each element in Θ, the belief functions in DS theory require no global probability [77]. Assignments are made merely on

observations, allowing the representation of ignorance. In the case in which a tweet does not provide any location data, the probabilistic models either discard such tweets [108], or apply the *Principle of Insufficient Reason* [114]. DS theory, on the other hand, can be indifferent to such incomplete data. If a tweet does not provide any evidence, it is handled as evidence for all possible locations in Θ, without any positive or negative support for a specific location.

The belief distribution over Θ using a source of evidence is called a *body of evidence*. DS theory supports fusion of multiple bodies of evidence via combination rules to obtain combined mass values. There are several methods that can be used for this combination [39, 110, 125, 135]. The most widely used is Dempster's rule, given in Equation 2.4 for a subset C of Θ where $C \neq \emptyset$. Considering the location-related features of tweets as independent sources of evidence, the corresponding basic probability assignments can be combined in a single model through Dempster's rule of combination. The combination operator in Dempster's rule, denoted by $\oplus$, is commutative and associative. It can be applied pairwise on multiple bodies of evidence in any order as long as their weight of conflict is finite [77]. Conflicting evidence obtained from two sources are handled by normalizing the mass values using the denominator in Equation 2.4.

$$(m_1 \oplus m_2)(C) = \frac{\sum\limits_{C=A\cap B} m_1(A) \times m_2(B)}{1 - \sum\limits_{A\cap B=\emptyset} m_1(A) \times m_2(B)} \tag{2.4}$$

Other combination methods differ from the Dempster's rule in how they handle the conflict between bodies of evidence [39]. For example, Yager's rule [135] adds the mass that represents the degree of conflict to the mass of ignorance $m(\Theta)$, rather than using it for normalization as in Dempster's rule. More specifically, when combining m_1 and m_2, it first applies the combination operator given in Equation 2.5.

$$(m_1 \odot m_2)(C) = \sum_{C=A\cap B} m_1(A) \times m_2(B) \tag{2.5}$$

This combination operator allows for a non-zero mass for the empty set $\emptyset$ [114]. As a final step, Yager's rule moves the combined mass assigned for $\emptyset$ to the mass assigned for Θ, as shown in Equation 2.6. The combination operator in Yager's rule is denoted by $\oplus'$. Unlike in Dempster's rule, this operator is not associative.

$$(m_1 \oplus' m_2)(C) = \begin{cases} (m_1 \odot m_2)(C) & \mathrm{C} \neq \emptyset, \Theta \\ (m_1 \odot m_2)(\Theta) + (m_1 \odot m_2)(\emptyset) & \mathrm{C} = \Theta \\ 0 & \mathrm{C} = \emptyset \end{cases} \tag{2.6}$$

Combining bodies of evidence, especially in the presence of conflicting information, has been extensively discussed in [39]. In that work, Dubois and Prade argue that in case of conflict, if we know that at least one of the sources is telling the truth, then a natural combination operator should use the disjunction of mass assignments. Accordingly, the authors proposed a combination rule that assigns the mass for conflict to set disjunctions, as given in Equation 2.7 [39].

$$(m_1 \oplus'' m_2)(C) = \sum_{C=A \cap B} m_1(A) \times m_2(B) + \sum_{C=A \cup B, A \cap B = \emptyset} m_1(A) \times m_2(B) \tag{2.7}$$

In this rule the combination operator is denoted by $\oplus''$. We explain how we use these combination rules for the event localization problem in Chapter 7, and discuss their results in our evaluations in Chapter 8.

CHAPTER 3

RELATED WORK

In this chapter, we present an overview of previous studies related to event detection in Twitter, incremental clustering approaches, similarity analysis and vector expansion, geospatial analysis in social networks, and applications of Dempster-Shafer theory.

3.1 Event Detection in Twitter

With the introduction of social networks, TDT techniques that were previously applied to large textual resources have been extended and adapted to perform event detection using tweets [21, 108, 109, 136]. In TDT parlance, the notions of "topic" and "event" are closely related. Topic is defined as an activity along with all related events and activities, whereas an event is a unique incident that happens at a specific time and place [11, 15]. In the literature, these two conceptes are sometimes used interchangeably. For example, in [109], tweets are clustered according to their content similarity to identify newsworthy topics, without making any specific distinction between topics and events. In [80], it is suggested that events can have a hierarchically nested structure (e.g., presidential elections can be considered an event, and a speech during the elections being a sub-event). Depending on the interpretation, such nested structures could also be handled as topics and events. In this thesis, topic is regarded as the subject of a text, whereas event is a specific activity that happens at a specific time and

place [38, 123].

The techniques for event detection in the literature can be classified in several ways. Depending on the *granularity of data* used for event detection, they can be categorized as document-pivot and feature-pivot methods. The former detect events by clustering tweets according to their similarity. For example, in [109], authors introduced a news processing system called TwitterStand[1] which employs an online clustering algorithm that measures the cosine similarity between the feature vectors of tweets, and clusters them into topics. A similar approach is adopted in [137] in which a single-pass incremental clustering algorithm is developed that automatically groups similar tweets into event-specific topics. In contrast, feature-pivot methods detect events by analyzing specific features in tweets, such as the frequencies of terms or their co-occurrences. Fung *et al.* propose a parameter-free probabilistic feature-pivot event detection method on text streams [46]. In that work, the authors find the expected probability for a document to contain a specific term, and compare it with the observed probabilities in order to detect bursty terms in a time window. A similar probabilistic burst detection technique was applied on tweets in [137] to determine which topic clusters can be associated with a real-world event. The Emergency Awareness System[2] presented in [99] analyses tweets in 5-minute windows in order to detect bursty keywords about emergency events, such as fires, earthquake and terrorist attacks. Identifying bursty terms for event detection has been investigated in numerous recent studies [72, 81, 133, 134].

By taking *tweet processing time* into consideration, algorithms can be classified as retrospective or online [11, 136]. Retrospective algorithms process a corpus of tweets to identify events discussed therein, while the objective of online algorithms is to identify events as they happen [16]. Due to a large volume of tweets arriving at a fast rate, online algorithms have to decide in a timely manner whether an event has just occurred and burst detection is a widely adopted method for this purpose [67, 140].

[1] http://twitterstand.umiacs.umd.edu/News [accessed 01 June 2016]
[2] https://esa.csiro.au [accessed 01 June 2016]

Regarding the *subjects of events*, event detection can be designed either to focus on a specific topic (usually implemented by querying specific terms in tweets) or applied to an open topic domain. For example, Sakaki *et al.* aim to detect earthquakes by collecting tweets containing the terms "earthquake" or "shaking" [108]. TwitInfo, another keyword based solution, is presented in [80], where tweets containing a given list of keywords are collected, and the peaks in their histograms are examined to detect events. Alternatively, TwitterStand uses tweet samples provided by Twitter services and the posts of a set of handpicked Twitter users without targeting a specific topic [109]. Similarly, Yin *et al.* utilise the tweets returned by Twitter's location based search API to apply an open-domain event detection [137].

Another aspect of event detection is the *temporal properties of events*. Algorithms can target instantaneous events, focusing on the instant they happened [108], or they can focus on activities that span a longer period with their beginning and ending times [140].

In this thesis, we cluster the tweets received from the Streaming API, and apply a burst detection method in order to detect events online. For burst detection, we analyse surges in the frequencies of terms in active clusters. Since we group similar tweets about a topic through clustering, our method does not require additional post-processing to discover event-related tweets. The proposed solution aims to detect instantaneous events on an open topic domain in near real-time. We would like to note that, although there is an expanding literature on event detection, examination of the effect of vector expansion on online event detection performance has not been a focus of attention thus far.

3.2 Incremental Clustering Approaches

The clustering of textual items retrieved from data streams, particularly from social networks, is an active research area [3, 4, 15, 60, 87, 113]. In [139], authors propose BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)

as an efficient incremental clustering algorithm for large databases, introducing the concepts of *cluster centroid* and *cluster feature vector*. Incremental clustering is regarded as an appropriate method for grouping continuously received textual items [15, 20, 112, 137]. In incremental methods, there are several ways to decide whether a newly arriving data point should join the closest cluster or initiate a new cluster. TwitterStand uses constant thresholds to make this decision [109]. Another approach is to calculate the mean and standard deviation of similarities between tweets, and compare the similarity of a recently posted tweet with these values [6]. In [112], authors define a minimum bounding similarity measure as the weighted average of similarities between the cluster's centroid and the tweets in the cluster, and use it to decide whether to add a tweet to an existing cluster. De Boom *et al.* train a logistic regression classifier to make this decision [33].

Constraints concerning memory size and processing capacity can become an issue to resolve in online clustering on continuous data streams [109, 112]. In particular, for high rates of streaming input data, a decision must be made to decide which stale clusters to delete from the memory, or which clusters should be allocated memory in order to start processing. A straightforward method is to remove the least recently updated stale cluster when the capacity limit is reached [6, 7]. In TwitterStand, the activity of clusters are periodically checked, and those with no recent change are removed [109]. Similarly, the average timestamp of recent arrivals in each cluster can also be used to identify stale clusters. This is facilitated in [5] by the use of a micro-clustering approach, in which the definition of a cluster feature vector is extended with the timestamp components of the data points in the cluster. In [112], the most similar clusters are merged when the upper limit for the number of clusters is reached, and a cluster is deleted if the average timestamp of the latest 10 percent of its tweets is more than 3 days old. In [33], an event is assumed to have a timespan of one day; thus, all active clusters are simply deleted when a new day begins. Alternatively, instead of keeping all the clusters in the memory and removing the stale ones when necessary, it is possible to select only the event-related clusters for maintenance at their creation. This approach is adopted in [137] by

allowing only the tweets that contain a bursty term to form clusters.

3.3 Similarity Analsysis and Vector Expansion

The problem of handling the spelling variances in documents to be clustered is discussed in numerous studies. In [115], the authors describe a Latent Semantic Analysis (LSA) based k-means clustering method for clustering a collection of documents. They aim to find relevant documents in the corpus that do not necessarily share any common words by reducing the dimension of document vectors. Similarly, Jun *et al.* build a clustering method by combining k-means clustering with dimension reduction through Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) [64]. In these studies, the dimensions of document vectors are reduced in a pre-processing phase before applying the clustering algorithm. However clustering an online stream of voluntarily generated content with no prior assumption about the vocabulary and topic introduces certain challenges due to noisy data, diverse topics, and frequent spelling mistakes [86]. New terminology can emerge when different topics are discussed by users in their posts [127]. This can result in each new post bringing about changes in the modelled vector space, which can limit the applicability of static training-based solutions. Furthermore, data pre-processing, formation of the vector representations, and clustering algorithms must be executed in a timely manner to keep pace with the continuously incoming data items.

Among the efforts to tackle the problem of resolving spelling variances in tweets, Kim *et al.* define a set of rules to handle abbreviations, minor typing errors, and terms separated by spaces [66]. They apply these rules on terms that either start with a capital letter or are enclosed by quotation marks. In [98], authors identify the proper nouns in tweets by applying Named Entity Recognition (NER) and boost the effect of these proper nouns. In [129], vector expansion is proposed as a solution to the issue of word mismatches in documents. The authors use a thesaurus, namely WordNet, to find the synonyms of the words in a query

text. However, due to the noisy lexical nature of micro-posts with frequent irregularities and incorrect spelling, it might not be possible to identify entities in a pre-existing knowledge source, such as a thesaurus [127].

As an alternative to using manually compiled thesauri, similar words can be identified by analyzing their distribution in large text corpora. A comparison of distributional and WordNet-based similarity measures is given in [8]. The authors state that the distributional similarity was effectively used to cover the items that did not exist in the WordNet vocabulary. A similarity thesaurus, which includes term-term similarity values extracted from the co-occurrence statistics, was constructed in [100]. In another study, Lin *et al.* focus on finding synonyms by analysing the results obtained from search engines and evaluating the commonality of translations in bilingual dictionaries [75]. Considering the dynamic content in Twitter that might not match the content in a dictionary, and the expressions with different meanings in different time contexts, we chose to extract similarity information from co-occurrence statistics rather than using a static dictionary.

Vector expansion has been applied in various studies to solve similar problems. For example, Cao *et al.* use it to expand query terms in order to improve the query results, but they claim that not all expansion terms are useful to improve the results [22]. A co-occurrence based term suggestion technique for e-commerce sites is presented in [59]. The earthquake prediction system in [88] employs a classifier to decide whether a tweet containing a query word really concerns an earthquake. The classifier makes use of the words before and after the query words as the context. Another study offers an insight to users in relation to why there is a surge in the popularity of a given keyword by finding the words that are commonly used together with the keyword [16]. We applied expansion methods to detect retrospective events in [92, 93]. We also investigated the extraction of associations between hashtags [91]. However, applying the proposed methods for online event detection brings new challenges. In this thesis, we present an improved version of the earlier methods for online event detection combined with

the clustering and burst detection approaches. The presented methods in this thesis can be executed to perform retrospective analysis, as well.

3.4 Geospatial Analysis in Social Networks

Recently, numerous studies have been conducted to detect real-world events by collecting public tweets in Twitter [15]. These include methods to identify earthquakes [31, 107, 108], disasters and crises [137], sports events [30], epidemics and diseases [2, 94], crime and accidents [74]. Although there are numerous efforts that aim to detect events in microblogs, not all of them propose a solution to estimate event locations. Usually, the main objective in these studies is performing the event detection task accurately in a timely manner, dealing with the barriers about performance and scalability [80, 133]. The problem of location estimation has previously been studied mostly in the context of newspaper texts and web pages. For example, a location is assigned to the clustered news articles collected through RSS feeds from online news sources in [122] . In [12], authors present a system for associating geography with web pages. Compared to newspaper articles and web pages, Twitter has several advantages. Spatial features in addition to textual content and the greater number of users acting as social sensors can be listed among the major advantages. On the other hand, Twitter is an uncontrolled, noisy, and sometimes unreliable environment. The large amount of information can also be a handicap considering possible conflicts and inconsistencies in the available data. Last but not least, the tweet volume and velocity can impose performance constraints, especially on real-time applications. These factors require special techniques to be developed for event localization in Twitter.

Among the studies that perform event localization using geotagged items in microblogs, Sakaki *et al.* estimated the epicenter of earthquakes in Japan using Kalman filters and particle filters, two widely used variants of Bayes Filters [44, 107, 108]. The authors searched for relevant tweets periodically that con-

tained earthquake related keywords in the content, and used the GPS metadata of geotagged tweets for estimating the locations of earthquakes. In a similar study [2], influenza-related tweets are collected using specific keywords, and GPS coordinates of these tweets are utilized to track the regional situation of the spread of the virus. In [137], authors captured tweets for a specific area of interest and executed clustering, burst detection, and classification algorithms on the tweet stream in order to detect emergency situations, namely, disasters and crises. In that work, location information about the detected events was presented to the end users by plotting the geotagged tweets about the event on a map according to their GPS coordinates.

An in-depth study on user profiles [52] suggests that the location field in user profiles should not be presumed to be strongly typed geographic information. The study reports that, since this attribute is a free-text field limited to 30 characters, it may contain multiple location names or even fake locations and sarcastic comments. In [74], tweets about crime and disaster are collected using keyword-based searches, and they are displayed on a map based on locations in user profiles. According to the observations in that study, only 12% of users specified a location in their profile. Hence, the authors predicted the user locations by analyzing users' previous tweets and locations of their friends. Sending this text to the public geocoding web services to obtain latitude-longitude pairs has been applied in [2, 137]. The location attribute in user profile has also been utilized in lieu of GPS data for non-geotagged tweets in several studies [2, 107, 108, 137].

In [82], it is argued that GPS data and user's profile location is not available most of the time, and therefore the authors chose to analyze the tweet content for location references by implementing a geoparser. A NER system for targeted tweet streams is presented in [73]. The problem of toponym recognition particularly in tweets has been investigated in [76]. In that work, the authors proposed an improvement on the conventional NER tools, which were trained on news data with formal text. The disambiguation of location names in tweet

content was discussed in [96]. Furthermore, Amitay *et al.* argued that, if there is a gazetteer at hand, a simpler and faster method than NLP techniques is to search for the place names in the text and resolve the possible ambiguities and this requires no training data [12]. The authors applied this method to assign a geographical focus for web pages using a heuristic for scoring place names based on their population and hierarchical relations. In [109], authors used a gazetteer for toponym recognition and described a heuristic for toponym resolution in tweets. The geographic focus for the clustered tweets are then determined as the most frequently mentioned toponym in tweet contents and user profiles. Similarly, Unankard *et al.* detected emerging events in Twitter by clustering, and they designated the most frequently mentioned location as the event location [123].

In [14], Ao *et al.* extracted geographical coordinates from the locations mentioned in microposts, their geotags, and the registered locations in user profiles. In order to estimate the location for a set of posts related to an event, they applied hierarchical clustering on these coordinates according to their Euclidean distances with each other. In another study, given a set of event-related tweets, majority voting based on location names in user profiles has been applied to determine the coarse-grained location (e.g., city, state, or country) for the event [48]. For fine-grained location, the authors applied part-of-speech (POS) tagging on tweet content to recognize the names of landmarks and addresses, and searched these labels in public geotagging services to retrieve geographical coordinates. The event location is then calculated as the average of these coordinates. The joint location estimation method proposed in [49] uses an expectation maximization approach for event localization using tweet contents and users. Given a set of events, some of which are approximately localized at the start, the relationship between events, users, and location references in tweets are modeled as a graph. The authors aim to estimate the unknown parameters in the graph that maximize the probability of observations, and thus estimate the most likely location for each event.

In this thesis, we propose a method that systematically combines the attributes in a single model using the theory of belief functions. The proposed method does not require any prior knowledge about the event. The primary novelty of the proposed method is devising a way to use the DS theory to combine three evidence sources in tweets and estimate the location of a given event.

3.5 Applications of Dempster-Shafer Theory

DS theory has been widely used in decision-making and classification problems, especially those that incorporate uncertainties [77]. In [116], authors used the theory to manage uncertainties for fraud risk assessment in financial statements. A stock trading expert system was developed in [40]. In that work, authors present a decision making system to generate advices to traders to buy, sell or hold some stocks using DS theory. An evidential reasoning approach based on belief structures in order to solve group decision analysis problems has been proposed in [45]. DS theory has also been used in decision making in hypothetical legal situations [32].

Using evidential reasoning and Bayesian approaches to solve target identification and tracking problems in military applications is explained in [18]. Coombs *et al.* identified types of ballistic missiles by fusing imperfect data retrieved from multiple sensors through Dempster's rule of combination [29]. Combining multiple classifiers in a single solution using DS theory was presented in [10]. A method to detect water contamination events by fusing data retrieved from sensors is given in [56].

DS theory was also proposed as a solution for grouping users' web search activities into sessions by combining two sources of evidence in web search logs [51]. In [125], the theory has been used to estimate the locations of Flickr images based on their meta-data in the form of textual descriptions. In that study, the authors train language models at multiple geographical granularities, and combine these models using the combination rules in DS theory. We believe that DS theory

can be applied effectively for the location estimation problem in Twitter and other social networks that exhibit similar location-related characteristics. This would help overcome challenges such as incomplete data, conflict in evidence, and ambiguous references.

CHAPTER 4

ONLINE EVENT DETECTION IN TWITTER

Our event detection method is composed of two major stages, namely a clustering stage and a burst detection stage. In order to enhance the accuracy of clustering and burst detection, we propose a method that analyzes the similarity of terms in tweets in a temporal context and uses these similarities for vector expansion. This is achieved by an additional stage of similarity analysis prior to the clustering and burst detection stages. An overview of stages for the proposed online event detection process is given in Figure 4.1.

In this chapter, after describing our data collection and modeling in Section 4.1, we present the details of baseline incremental clustering and our proposed enhancement in Section 4.2. The burst detection that we applied to select event-related clusters is given in Section 4.3. Our enhancement for incremental clustering can group similar tweets into clusters more coherently; and thus, it improves the accuracy of clustering and online event detection, as demonstrated in Chapter 5.

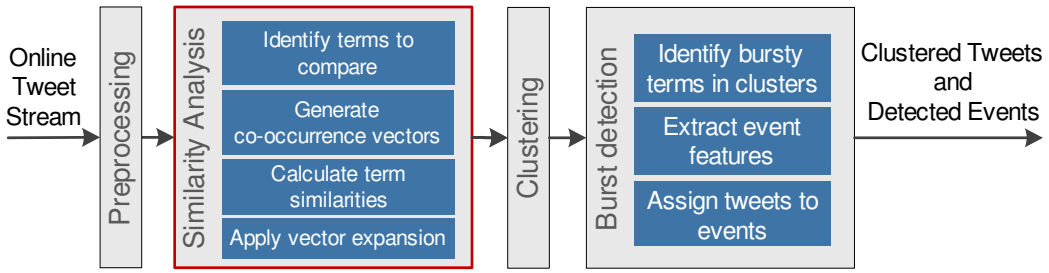

Figure 4.1: Online Event Detection Stages

4.1 Data Collection and Modeling

We collect tweets posted in Turkey online using the Streaming API of Twitter and its geographical filtering option. We define a geographical filter based on the boundary coordinates of the country in terms of latitude and longitude, and use the Java library of Twitter4j[1] for service connection and tweet collection tasks.

As new tweets are received from the stream, they are processed in sliding windows, which we call *tweet buckets* or simply *buckets*, denoted by B. In the literature, the size of the sliding windows is mostly determined either in terms of the clock time or in terms of the number of tweets [46, 80, 136]. In this thesis, we devise a hybrid bucket size model that employs both time and tweet count. In the hybrid model, the time window for a bucket is determined as one minute. However, if the tweet count in a bucket is below a certain threshold after one minute since the receipt of the first tweet in that bucket, the system waits for more tweets to accumulate. This threshold is set to 200, which is approximately the average number of tweets per minute received by the configured streaming client in a day.

The reason for devising the hybrid bucket size model is twofold. Firstly, the flow of tweets received by the aforementioned streaming client considerably changes depending on the time of the day. Specifically, the number of tweets received at midnight is usually very low, whereas a popular event at daytime can result in high tweet rates. As a result, if the bucket size was determined in terms of a fixed time interval, such disparities in tweet rates would make it harder to detect surges in tweet counts and manage cluster lifetimes. Secondly, if buckets were generated from a fixed number of tweets, then tweets about a minor event could be dispersed at distant buckets, probably due to a major event allocating a larger portion of the buckets. This would inherently limit the number of concurrent events that can be detected. The hybrid bucketing approach using both time

[1] http://twitter4j.org/en/index.html [accessed 01 June 2016]

and tweet count yields a more homogeneous bucket size throughout the day, and does not require any *a priori* assumption about the number of concurrent events.

When the time and tweet count requirements are satisfied to generate a bucket, a bucket B_i is formed as a list of n tweets denoted by $[B_{i,1}, ...B_{i,n}]$. Tweets are modelled in the standard vector space model based on the bag-of-words representation of their textual content [136]. In the pre-processing phase, given a bucket B_i, the stemming and stop word removal procedures are applied to its tweets (including the removal of punctuation), and the vocabulary statistics (i.e., term and document frequencies) are updated. Associated with each tweet $B_{i,j}$, we keep its unique id, its posting time, the list of its stemmed terms with their frequencies, and a tweet vector $\vec{B_{i,j}}$ as the normalized tf-idf values of the terms in its content. A tf-idf vector is represented as the tuples of the form $\langle x, w \rangle$ where x is a term in the tweet and w is its normalized tf-idf value [21]. After pre-processing, the bucket is fed into the event detection engine.

The temporal context for a bucket at time t can be described by the tweets posted before and after t. By keeping a number of tweets in a *tweet history* after they are processed, we can easily access the past portion of the time context. The future portion of the context requires "foreseeing" the tweets to be processed later. For this purpose, we introduce the concept of *look-ahead cache*. When a bucket of tweets is received from the tweet stream, rather than immediately applying the clustering and burst detection methods, we first keep this bucket in the look-ahead cache to be used in the context description. In other words, event detection follows the online tweet stream one-cache size behind. The overall event detection process is depicted in Figure 4.2. The following sections present the details of clustering, burst detection and similarity analysis executed on the tweets in bucket B_t.

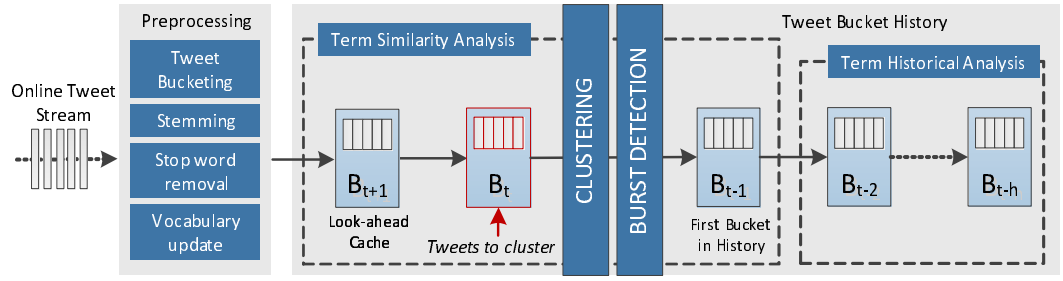

Figure 4.2: Proposed Online Event Detection Process using Vector Expansion

4.2 Clustering

The clustering technique we propose is an enhancement on the incremental clustering, an approach widely employed on streaming data [20, 109, 137, 139]. Considering their handling of the tweet stream (online vs. offline), topic independence, and clustering algorithm characteristics, we selected the incremental clustering method presented in [137] as a comparable baseline clustering algorithm. Due to several implementation details that were not clarified in that work (e.g., the cluster termination condition, vocabulary management), we referred to other studies in the literature [3, 15, 109, 136] and implemented the clustering algorithm given in Section 4.2.1. This baseline incremental clustering algorithm will be referred to as IC in the remainder of the chapter. In Section 4.2.2 we introduce how to extract term similarities in a time context and use them in the vector expansion process to develop Incremental Clustering with Vector Expansion (ICVE).

4.2.1 IC: Incremental Clustering

The clustering algorithm executes on the most recent bucket in a single pass, as the tweets are continuously received from the tweet stream and accumulate in a new bucket. It maintains the list of active clusters and updates these clusters incrementally. A cluster c is represented by a cluster centroid vector $\vec{c}$, its creation time, and the list of terms in the cluster with their frequencies [139]. The centroid vector $\vec{c}$ is the normalized vector sum of all tweet vectors

in the cluster [137]. We also keep the list of tweet ids added to the cluster grouped by their buckets. An element in this list is represented by B_i^c, meaning the set of tweets received in bucket B_i and added to the cluster c. This bucket information helps us obtain a term frequency histogram in the clusters and apply burst detection techniques, as will be discussed in Section 4.3.

When a tweet $B_{i,j}$ is to be clustered with incremental clustering, its tweet vector is compared with the centroid vectors of the active clusters by using cosine similarity [109, 111]. If the most similar cluster has a similarity score that is greater than a predefined threshold, called the *merge threshold*, the tweet is assigned to that cluster [98]. This assignment causes the cluster features to be updated using the features of this recently added tweet. Otherwise, if no similar cluster is found for $B_{i,j}$, a new cluster is created using this tweet's features. There is also a termination condition for a cluster, also referred as *cluster death* [3]. After the processing of each bucket, the active clusters are checked whether they have been stale, and a cluster is terminated if no tweet has been added to it for the last three buckets consecutively. The termination process can store the cluster data to a persistent storage for reporting purposes, or simply discard it depending on the number of tweets collected in that cluster.

4.2.2 ICVE: Incremental Clustering with Vector Expansion

The proposed enhancement on the traditional incremental clustering algorithm discovers and measures similarities between terms, and uses these similarities for applying a vector expansion process prior to the clustering. Considering the process flow in Figure 4.2, where B_t represents the bucket of tweets to cluster, the proposed enhancement introduces an expansion of tweet vectors in bucket B_t and the centroid vectors of active clusters. This is achieved by leveraging the contextual information in the neighboring buckets of B_t in the following four steps, as presented in Figure 4.1:

1. Identify terms in B_t to compare for their pairwise similarities,

2. Find co-occurrence vectors for terms in B_t using the co-occurrence statistics in B_{t+1}, B_t, and B_{t-1},

3. Calculate term similarities as real numbers in the range [0,1] using the co-occurrence vectors,

4. Expand tweet and active cluster vectors using extracted similarities.

These four steps are explained in the following sections. After the expanded vectors are obtained for the tweets in B_t and for the active clusters, incremental clustering described in Section 4.2.1 is executed on these expanded vectors. For that reason, we refer to this clustering technique as Incremental Clustering with Vector Expansion (ICVE).

4.2.2.1 Identifying Terms to Compare

The importance of selecting discriminative terms for vector expansion is discussed in [22]. In that work, authors argue that not all expansion terms are useful for improvement. In other words, analyzing similarities for each pair of terms mentioned in tweets of B_t might not result in an improvement. Moreover, the number of distinct terms in a bucket can be very high, and thus comparing each of them with the other for their similarity can be a time-consuming task that is not suitable for online stream processing. Therefore, at this step, we aim to select the most discriminative terms in B_t to be analyzed for their similarities. We refer to these terms as *terms to compare*.

The selection of terms to compare is done by a *concept decomposition* process [3]. For the concept decomposition, we apply a separate clustering procedure only for the tweets in B_t with a high merge threshold to obtain coherent tweet collections. Our intuition is that the most descriptive terms for the detected concepts should also represent the most discriminative terms in B_t. Therefore, the terms with high tf-idf values in the centroids of these clusters are selected as terms to compare. We would like to note that, the clusters mentioned here

are used only for concept decomposition to determine terms to compare. That means, they are not included in event detection process, and therefore they are discarded after the similarity analysis.

The procedures in finding *terms to compare* can be exemplified on an example scenario for earthquake. In case of an earthquake event, it is likely to observe a number of tweets in B_t that contain the terms "earthquake" or "shake". Considering the tweets containing either of these terms as two separate sets, the concept decomposition process applied on B_t is expected to group these tweets in two separate clusters. Normally, these two terms would have the highest tf-idf scores in the corresponding cluster centroids, and thus "earthquake" and "shake" would be included in the set of terms to compare to be analyzed for their statistical similarities.

4.2.2.2 Finding Co-occurrence Vectors

Since two terms that are strongly related to each other in one context may be unrelated in another, temporal locality should be taken into account when extracting similarities between terms [16]. As described in Section 4.1, the temporal context for bucket B_t is defined as the tweets posted before and after the bucket in B_{t-1} and B_{t+1}, respectively. More specifically, before applying the event detection process on a bucket B_t, we discover the contextual similarities between the terms to compare in B_t by using the tweets in B_{t+1}, B_t, and B_{t-1}. In this sequence, B_{t+1} represents the most recent bucket received from the tweet stream that is kept in the look-ahead cache as shown in Figure 4.2. B_{t-1} is the last analyzed bucket for event detection before B_t.

Term associations are useful in a vector expansion process if the expansion adds useful information to the vector that may not be explicitly specified in the corresponding text. In this thesis, our primary focus is the second-order relations, which are supposed to identify terms that can be used interchangeably in a given text [102]. By preparing co-occurrence vectors and measuring their similarities,

similarity scores are obtained in the range of [0,1] between the term pairs. There are several ways to produce co-occurrence vectors. We experiment with three of them, which we call Second Order, Strict Second Order, and dimension-reduced using Singular Value Decomposition (SVD). Once the co-occurrence vectors are generated by one of these methods, similarity of two co-occurrence vectors is measured by using cosine similarity.

Second Order (SO): This method generates co-occurrence vectors by using the number of times that two terms appear together in tweets. Using the terms and tweets in B_{t+1}, B_t, and B_{t-1}, a binary term-tweet matrix A is formed where $A[i][j]$ is set to one if the term corresponding to the i^{th} row appears in the tweet corresponding to the j^{th} column. Otherwise, $A[i][j]$ is set to zero. Multiplying A with its transpose A^T gives the pairwise co-occurrence counts of terms. In order to speed up the process, since we need the co-occurrence vectors only for the *terms to compare* identified in Section 4.2.2.1, we process only the non-zero entries in matrix A and obtain the co-occurrence vectors without applying matrix multiplication. The co-occurrence of a term with itself is discarded since it is trivial. The computational complexity of this process is then $\mathcal{O}(mnd)$, where mn is the dimension of A and d represents the number of terms to compare (with $d \ll m$). While comparing the co-occurrence vectors of two terms for their similarity, their co-occurrence values with each other are set to zero and the vectors are finally normalized to be used in the similarity score calculation.

Strict Second Order (SSO): The idea of this method is based on the fact that co-occurrence vectors generated by SO implicitly inherit the first-order similarities. If two terms x_i and x_j co-occur very frequently, their co-occurrence vectors are expected to have almost the same values. As a result, comparison of these term vectors results in a high similarity score, which might not necessarily mean that they can be used interchangeably in texts. In order to eliminate the effect of first order co-occurrences, for finding the co-occurrence vectors for x_i and x_j to compare with each other, we exclude tweets that contain both x_i and x_j. That means, the co-occurrence vector of a term depends on which term it

is compared to. Therefore compared to SO, the computational complexity is increased by a factor of d, which results in a cost of $\mathcal{O}(mnd^2)$.

Dimension-reduced using SVD (SVD): SVD is a method that decomposes a matrix A into three matrices U, S, and V with specific properties [17, 35]. Matrix S has the property that, if k is the rank of matrix A, only the first k values in the diagonal of S are non-zero. Moreover, a reduction can be made on S by keeping the first few values in the diagonal and setting the remaining values to zero. If we call this reduced matrix S_R, a reduced approximation of A, call A_R, can be produced by taking the product US_RV^T. By reducing the dimension of the term-tweet matrix A to A_R using SVD, trivial and incidental co-occurrences can be filtered out and only the significant ones can be obtained. The reduced term co-occurrence matrix is then generated from the product of A_R with A_R^T, given in Equation 4.1. While comparing the co-occurrence vectors of two terms x_i and x_j, their co-occurrence values with each other are set to zero. The co-occurrence vectors are finally normalized to be used in the similarity score calculations.

$$A_R A_R^T = US_RV^T(US_RV^T)^T = US_R^2U^T \tag{4.1}$$

The extent of dimension reduction can be defined by changing the reduction ratio, i.e., the ratio of the rank of S_R to the rank of S. In the experiments, an emprically determined reduction ratio of 80% is used. That means the top 20% of the diagonal in S is kept in S_R. In the implementation, we used the JAMA[2] library for SVD operations. The computational complexity of the decomposition of matrix A is $\mathcal{O}(m^2n + mn^2)$ [17]. The co-occurrence vector generation also includes multiplication of decomposed matrices in $US_R^2U^T$, which is $\mathcal{O}(mn^2 + m^2n)$.

[2] http://math.nist.gov/javanumerics/jama [accessed 01 June 2016]

4.2.2.3 Calculation of Term Similarities

After generating the co-occurrence vectors using one of the co-occurrence vector generation methods presented in Section 4.2.2.2, these vectors are used to find the similarities between pairs of terms to compare identified in 4.2.2.1. To this end, we calculate cosine similarities between the co-occurrence vectors, which represent the similarity scores between the corresponding pair of terms. Given the normalized co-occurrence vectors, the calculation of similarities for d^2 term pairs is $\mathcal{O}(md^2)$.

For each term to compare, we keep only the top few similarity tuples with the highest similarity scores above a certain threshold. We refer to the number of similarity tuples we keep for a term as the *term similarity count*, and the minimum similarity threshold between two terms is called as *term similarity threshold*.

The output of this third step in term similarity analysis is a list of tuples of the form $\langle x_i, x_j, s_{i,j} \rangle$, where x_i and x_j are two terms both of which are included in the terms to compare that were found in Section 4.2.2.1, and $s_{i,j}$ represents their similarity score using one of the co-occurrence vector generation methods described above (with $s_{i,j} \geq$ *term similarity threshold*) . In accordance with the names of the co-occurrence vector generation methods, we label three variations of ICVE with respect to these three vector generation options as ICVE-SO, ICVE-SSO, and ICVE-SVD, and evaluate their effect in accuracy seperately.

4.2.2.4 Vector Expansion

Pairwise similarity scores between the terms to compare are used in the expansion of tweet vectors and active clusters' centroid vectors before applying the incremental clustering procedures. Given a tf-idf vector $\vec{v}$ with term-weight tuples of the form $\langle x_i, w_i \rangle$ and a list of similarity scores between the term pairs denoted as $\langle x_i, x_j, s_{i,j} \rangle$, the expansion algorithm first initializes an expanded tf-

idf vector $\vec{v^e}$ with the values in $\vec{v}$, and for each $\langle x_i, w_i \rangle$ in $\vec{v}$, it finds other terms x_j that are similar to x_i with $s_{i,j} > 0$, and updates the weight of the similar term x_j in the expanded vector $\vec{v^e}$ by adding $w_i s_{i,j}$ to its previous weight w_j. In other words, the weight of a term in the tf-idf vector is updated in the expansion process using the weights of similar terms and their similarity scores. After all terms in $\vec{v}$ are processed, the expanded vector is normalised. Given the similarity scores between term pairs to compare, the time complexity of the expansion for the tweet vectors in bucket B_t is $\mathcal{O}(mnd^2)$.

Once expanded tweet and cluster vectors are obtained, the incremental clustering method described in Section 4.2.1 is applied on these vectors. In other words, while clustering a tweet $B_{t,i}$ using ICVE, comparisons are performed between its expanded tweet vector and the expanded centroid vectors of the active clusters. The clusters generated as a result of this vector expansion are found to be more coherent and less fragmented, as will be demonstrated in Chapter 5.

4.3 Burst Detection

Burst detection stage aims to detect sudden surges in the frequency of terms received from the tweet stream. A burst is basically identified by comparing the frequency of terms in the recent buckets with their average frequencies in the history. A remarkable increase in the frequency can be interpreted as an indication of an event. However, a term that exhibits bursty statistics with respect to the tweet stream may be dispersed in different clusters. Therefore, it does not necessarily indicate a newsworthy event. In order to mark an event about a topic, we additionally expect a bursty term in tweet stream to be bursty in one of the active clusters, as well. The process of detecting bursts and marking new events in active clusters is executed in three steps as follows.

4.3.1 Detection of bursty terms in clusters

For the detection of bursty terms, we adopt a method similar to the selection of bursty keywords in [66]. As depicted in Figure 4.2, after the tweets in bucket B_t are processed by the clustering algorithm, the burst detection method is executed on the active clusters. Burst detection starts with finding average frequencies of the terms in the oldest (h-1) buckets of the bucket history. Let $B_{t|x}$ represent the set of tweets in bucket B_t that contain the term x. Then, the mean frequency $\mu_t(x)$ at time t of a term x in history is defined as in Equation 4.2. In this equation, if the cardinality $|B_{t-i|x}|$ is 0 for some i, we assume a frequency of 0.5 for this term in that bucket. The reason for this smoothing is to prevent spurious increases in the freqeuncy to be interpreted as bursts and to avoid zero-division in the calculation of the burst ratio. $\mu_t(x)$ is used in the calculation of burst ratios for the buckets B_t and B_{t-1}, as will be explained shortly.

$$\mu_t(x) = \frac{\sum_{i=2}^{h} max(|B_{t-i|x}|, 0.5)}{h-1} \tag{4.2}$$

If the number of tweets that contain x in buckets B_t and B_{t-1} are significantly higher than $\mu_t(x)$, we mark x as a bursty term at time t. In order to measure the significance, we calculate the ratios of these numbers (i.e., the cardinalities of $B_{t|x}$ and $B_{t-1|x}$) to $\mu_t(x)$, which we call *burst ratio* of x. If the burst ratios of a term x are above a threshold Δ for the two consecutive buckets B_t and B_{t-1}, then x is selected as a bursty term at time t. The reason for requiring high burst ratios in both B_t and B_{t-1} (not in B_t only) is to minimize noises and spams. In order to handle events that happen at a time close to the middle of a bucket window, we further relax this condition by expecting a burst ratio of $\Delta/2$ for one of these two buckets.

This process finds the bursty terms in the tweet stream without considering the distribution of these terms in active clusters. However, tweets that contain a bursty term might be related to different topics, mentioned in tweets for different

purposes. Therefore, for each bursty term identified in the tweet stream, we also find their distributions in active clusters. Reminding that B_t^c represents the set of tweets in bucket B_t that are added to the cluster c, it is straightforward to identify the set of tweets in B_t^c that include the term x, denoted by $B_{t|x}^c$. Burst ratios of x with respect to a cluster c can be found in a similar way, i.e., by dividing the cardinalities of $B_{t|x}^c$ and $B_{t-1|x}^c$ to $\mu_t(x)$. If these ratios are also above the thresholds, we signal the detection of a new event, and the cluster with a bursty term is marked as an *event cluster*. These bursty terms are then used to extract event features from the event cluster.

4.3.2 Extraction of Event Features

Once a burst is detected in a cluster, the corresponding event features are extracted from tweets to obtain a human understandable description of the event. If a term x is identified as bursty in cluster c, then the tweets $B_{t|x}^c \cup B_{t-1|x}^c$ are marked as the event's bursty tweets. If multiple bursty terms are detected at time t in an event cluster, tweets that include at least one of them in the cluster constitute the set of the event's bursty tweets. That means, bursty terms detected at the same time and in the same event cluster are treated as descriptors of the same event. Event's bursty tweets are important since they are the first and probably the most relevant tweets that describe the event.

An event is described by three features, namely the event time, event centroid vector, and the best tweet for the event. The posting time of the first bursty tweet is assigned as the *event time*. An *event centroid vector* is defined as the normalized vector sum of all vectors corresponding to the event's bursty tweets. By measuring the similarities between the event centroid vector and vectors of the event's bursty tweets, the tweet with the highest similarity is designated as the *best tweet* to describe the event.

4.3.3 Assigning tweets to events

The final step in the event detection process is to assign tweets to the detected events. Once an event is detected in a cluster, the tweets collected in that cluster for a duration of time after the detection of the event is considered to be mentioning that event. It is possible to have several events at different but close buckets in a cluster. For example, two goals can be scored in the same game resulting in two distinct bursts. In that case the correct event for a tweet in the cluster is selected by comparing the tweet vector with the event centroids. The event with higher similarity is selected for that tweet.

These three steps of burst detection are executed for B_t after the clustering stage. The described burst detection method is advantageous for several reasons. Firstly, it does not require any prior annotation, supervision, or training to learn the historical frequency distribution of terms. Analyzing frequency distribution of terms automatically provides an inherent adaptation to changes in the vocabulary of new tweets. Secondly, it does not require substantial processing power or memory, which makes it suitable for online analysis of tweets. Thirdly, since similar tweets about a topic are already clustered before burst detection, event-related tweets can easily be obtained without searching for further correlations between bursty terms and tweets. Tweets in the cluster of a bursty term can be considered to constitute a coherent tweet collection about the detected event. Last but not least, by changing Δ, the preferred awareness level can easily be maintained. Low thresholds could generate a higher number of alarms, part of which could be duplicates or false positives. High thresholds can be preferred if only the major events are of interest to the user.

CHAPTER 5

EVALUATION OF INCREMENTAL CLUSTERING WITH VECTOR EXPANSION

In this chapter, we present the results of the proposed online vector expansion method on clustering and event detection accuracy. We describe the setting of the experiment and data set that is used in the evaluation in Section 5.1 and Section 5.2, respectively. In Section 5.3, we evaluate accuracy of clustering using the baseline incremental clustering method and using our enhancement. Implication of the enhancement in clustering on event detection accuracy is demonstrated in Section 5.4.

5.1 Evaluation Setting

Our evaluations focus on the implications of the proposed vector expansion on two aspects of event detection, namely clustering accuracy, and event detection accuracy. The clustering accuracy refers to the accuracy of grouping similar tweets in generated clusters. An accurate clustering algorithm should collect as many relevant tweets as possible in the same cluster. Event detection accuracy analysis addresses the correct detection of events with minimum false alarms and missed events. We perform and evaluate the proposed event detection method on tweets posted in Turkey between May 1 and May 10, 2014 that are received by our streaming client. This tweet set is composed of more than 2.3 million tweets posted by almost 250,000 distinct users. The values for the constants and

Table5.1: Constants and Thresholds in Experiments

Minimum Bucket Size	=	200 tweets
Cluster Maximum Idle Time	=	3 buckets
Merge Threshold in Concept Decomposition	=	0.5
Term Similarity Threshold	=	0.5
Term Similarity Count	=	10
Number of Buckets in History h	=	5

thresholds that we used in our executions are presented in Table 5.1.

5.2 Ground Truth Annotation

As a reference annotation, we select and annotate a set of *target events* and their associated *target tweets* based on our tweet set. In order to minimize the effect of human interpretation while making these annotations, we devise a partially automated way. It is based on the topic-specific tweet collection method applied in [108], where authors use specific keywords to collect tweets about earthquakes, and additionally use a classifier to minimize false positive tweets due to ambiguous usages. We adopt a similar approach for the annotation of tweets in our dataset. We select earthquakes and goals in soccer games as two target topics in our evaluations because 1) they are real world events with specific time and place to set as ground truth, 2) the relevance of their tweets are often clear, and 3) their tweets are easier to select using keywords. Events about other topics detected by our topic-independent event detection algorithm are not included in the evaluation.

Considering earthquakes and goals as two specific topics, we first select *topic-specific keywords* that best describe these topics. For earthquakes, we set "deprem" and "salla" (Turkish equivalents of "earthquake" and "shake", respectively) as the topic-specific keywords. Goals are described by the terms "gol", "gool" and "goal" ("gol" is the Turkish spelling for "goal"). Ground truth an-

notation is carried out in three steps. In the first step, we process tweet buckets, and count tweets that contain topic-specific keywords in each bucket. If a remarkable number of tweets is observed, we search the past news sources to learn about the details of these events. This helps us extract *event-specific keywords* and determine the *event times*. Event-specific keywords for an earthquake are the names of the city and town of the earthquake's epicenter. For a goal event, event-specific keywords are the names of the competing clubs and the name of the scoring player. The result of the first step is the list of target events with their times and descriptive keywords. The second step in annotation makes a second pass on tweets, marking a tweet as target tweet if it contains either a topic-specific or an event-specific keyword, and assigns these tweets to the corresponding target events. The third and final step is a human controlled elimination of false positives in target tweets, with the same purpose of the classifier presented in [108]. Crooks *et al.* argue that, as time passes after an event, the tweets can turn out to be "about the news of the event", rather than being "about the event" [31]. This makes it harder to decide the relevance of tweets with the event. Therefore we execute our evaluations on the tweets posted within 10 minutes after the event time. Finally, by discarding the minor events with few tweets, we obtained 3 earthquakes and 28 goals as target events in our tweet set.

5.3 Analysis of Clustering Accuracy

For the evaluation of clustering accuracy, we employ the methods based on cluster-event contingency tables and precision-recall analysis of the clustered tweets [72, 136]. Our intuition is that the cluster about an event should group as many relevant tweets as possible, and with minimum false positives. This requires identifying the best cluster generated as a result of a target event. Therefore, for a target event E_m with the set of target tweets T_m, the cluster having the highest number of target tweets is designated as the best matching cluster for that target event. If a cluster C with the set of tweets T_c is found to

Table5.2: Cluster-target event contingency table

	in E_m	**not in** E_m
in C	$T_c \cap T_m$	$T_c \setminus T_m$
not in C	$T_m \setminus T_c$	$\emptyset$

match the target event E_m, the corresponding cluster-event contingency table is formed, given in Table 5.2. Based on the values in this contingency table, precision and recall values for the cluster C are found as in Equation 5.1. For the overall accuracy of the cluster, F_1-score is calculated using Equation 5.2.

A cluster can be matched with multiple target events, as in the case of two consecutive events about the same topic with similar content. If C was matched with E_{m_1} and E_{m_2}, then the ground truth tweet set would be $T_{m_1} \cup T_{m_2}$. In general, if C is found as the best cluster for k different target events $E_{m_1}...E_{m_k}$, T_m in Equation 5.1 is substituted by $T_{m_1} \cup T_{m_2}... \cup T_{m_k}$. To measure the overall performance of an algorithm, we employ the macro-average approach and find the average values of the per-cluster scores [136].

$$prec(C) = \frac{\mid T_c \cap T_m \mid}{\mid T_c \mid} \qquad recall(C) = \frac{\mid T_c \cap T_m \mid}{\mid T_m \mid} \tag{5.1}$$

$$F_1 score(C) = \frac{2 \times prec(C) \times recall(C)}{prec(C) + recall(C)} \tag{5.2}$$

In order to evaluate the improvement obtained by our enhancement based on vector expansion, we execute IC and ICVE on the same dataset and compare their results. We use three methods to generate co-occurrence vectors as presented in Section 4.2.2.2. We call the proposed enhanced clustering algorithms ICVE-SO, ICVE-SSO, and ICVE-SVD, in accordance with the names of the co-occurrence vector generation methods.

The accuracy of the incremental clustering is strongly related with the merge threshold parameter. Since no specific merge threshold value was given for the incremental clustering in [137], we analyzed the performance of the clustering

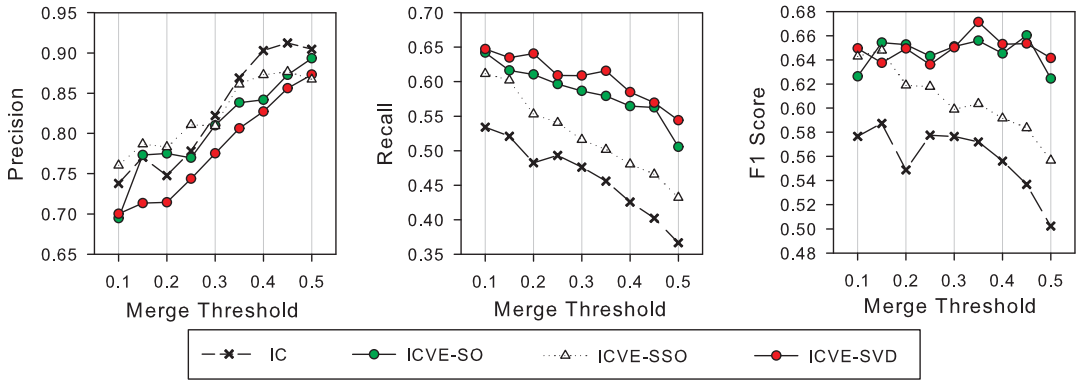

Figure 5.1: Cluster accuracies for IC, ICVE-SO, ICVE-SSO and ICVE-SVD using different merge threshold values

algorithms under different merge thresholds ranging between 0.1 and 0.5. The results of the precision, recall and F_1-score analysis are presented in Figure 5.1. The graphs show that precision is usually similar for all algorithms. However, recall is remarkably higher than the baseline clustering algorithm when we apply similarity analysis and vector expansion. Its effect is also remarkable on the F_1-scores.

While all vector expansion enhancements on the baseline clustering algorithm outperform IC according to the F_1-scores, the best results are obtained by ICVE-SVD and ICVE-SO. We can thus conclude that the implicit first-order relationships in the co-occurrence vectors are useful for finding similar tweets. It is also notable that, no matter how we change the merge threshold, all F_1-scores obtained by ICVE-SVD and ICVE-SO are higher than the maximum F_1-score that can be achieved by IC.

Another observation is that while high merge threshold values have a positive effect on the precision, they cause tiny fragmented clusters and low recall. More importantly, fragmented clusters may not survive for a long time and finally be discarded at their termination if they contain a small number of tweets. Therefore, to determine the best merge threshold for an algorithm, we require all target events to be matched with a cluster with the highest precision possible. These conditions are satisfied by using 0.25 as the merge threshold for IC, ICVE-

Table5.3: Detected similarities for the term "Sneijder" using SO, SSO and SVD

	Similar terms and their similarity scores
SO	wesley(0.96), sneijderr(0.95), sniejder(0.94), sneıjder(0.94), goll(0.92), sneijdeer(0.92), sniper(0.92)
SSO	sneijderr(0.95), sniejder(0.94), sneıjder(0.94), wesley(0.93), sneijdeer(0.92), sniper(0.92), goll(0.91)
SVD	wesley(0.96), sneijderr(0.95), sneıjder(0.93), goll(0.92), sniejder(0.92), sniper(0.91), weesley(0.91)

SSO and ICVE-SVD. The merge threshold to satisfy these conditions for ICVE-SO is found as 0.30. It is noteworthy that even with a higher merge threshold, ICVE-SO yields a higher recall than IC does.

We present an example for the similarities discovered automatically in Table 5.3. The table shows the similarity scores identified for the term "Sneijder" by using the three co-occurrence vector generation methods. The similarities are calculated by the time that a goal was scored by the Dutch player named "Wesley Sneijder" playing in the Turkish league. As we previously discussed, SO and SVD include an implicit first-order relationship. Therefore, the first name of the player "Wesley" is found as the most similar term for his surname, since they usually co-occur in tweets. On the other hand, SSO yields a list with different spellings of the term "Sneijder" having higher similarity scores.

As we have discussed in Section 4.2, the proposed vector expansion process introduces additional computational complexity to the online clustering. Among the three ICVE methods, ICVE-SVD has the highest computational cost while ICVE-SO is the most efficient one. We show this in practice by measuring the time it takes to process a bucket. In our tweet set, the total number of buckets is 8187 with at most 2469 tweets in a bucket. We performed our evaluations on a personal computer with a 3.4Ghz Intel Core i5 processor and 8GB of DDR3 memory. The average and maximum execution times of a bucket are presented in Figure 5.2(a) and Figure 5.2(b), respectively. These figures show that the enhancements introduced by ICVE-SO and ICVE-SSO do not incur any remarkable cost to prevent online processing of the tweet stream. On the other

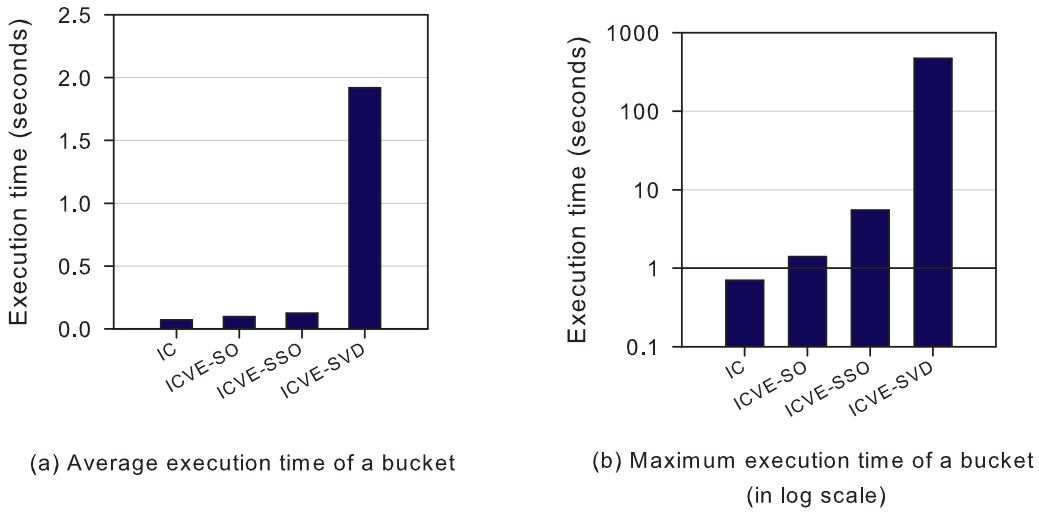

Figure 5.2: Execution times of clustering algorithms

hand, ICVE-SVD may take much longer time to process a bucket. In Figure 5.2 (b), we see that the time for ICVE-SVD to process a bucket can reach up to 470 seconds. Therefore, because of the efficient and accurate clustering achieved by ICVE-SO, in the rest of our evaluations, we compare IC with ICVE-SO under their best merge thresholds (i.e., 0.25 and 0.30, respectively).

In addition to the macro-average accuracy results presented in Figure 5.1, we evaluate the precision, recall, and F_1-scores for each of the 31 target events, and present the results in Table 5.4. If a cluster is matched with multiple events, we use the accuracy scores of the cluster for each of these events. The table shows that the accuracy is higher with ICVE-SO for most of the events. The significance of the improvement is analyzed through an unpaired t-test[1]. For both recall and F_1-scores, the t-test score yields a value lower than 0.05, which indicates that the improvement is statistically significant. The results obtained by each of the experimented settings are given in Table A.1

[1] http://www.socscistatistics.com/tests/studentttest/Default2.aspx [accessed 01 June 2016]

Table5.4: Precision, Recall and F_1-scores per each event, using the baseline IC and the enhanced ICVE-SO algorithms

Target Event	**Target Tweets**	Precision		Recall		F_1-score	
		IC	ICVE-SO	IC	ICVE-SO	IC	ICVE-SO
EQ#1	34	0.900	0.900	0.794	0.794	0.844	0.844
EQ#2	40	0.569	0.576	0.825	0.850	0.674	0.687
EQ#3	21	0.708	0.783	0.810	0.857	0.756	0.818
GOAL#1	42	0.095	0.922	0.254	0.836	0.138	0.877
GOAL#2	499	0.963	0.922	0.577	0.836	0.722	0.877
GOAL#3	29	0.095	1.000	0.254	0.379	0.138	0.550
GOAL#4	134	1.000	0.668	0.634	0.828	0.776	0.740
GOAL#5	40	0.958	0.767	0.575	0.575	0.719	0.657
GOAL#6	31	1.000	1.000	0.323	0.323	0.488	0.488
GOAL#7	28	1.000	1.000	0.607	0.607	0.756	0.756
GOAL#8	20	1.000	1.000	0.350	0.400	0.519	0.571
GOAL#9	36	1.000	0.895	0.333	0.472	0.500	0.618
GOAL#10	20	0.750	0.857	0.300	0.300	0.429	0.444
GOAL#11	51	1.000	1.000	0.745	0.686	0.854	0.814
GOAL#12	297	0.857	0.802	0.444	0.872	0.585	0.836
GOAL#13	248	0.672	0.853	0.315	0.867	0.429	0.860
GOAL#14	19	0.929	0.938	0.684	0.790	0.788	0.857
GOAL#15	14	0.875	0.875	0.500	0.500	0.636	0.636
GOAL#16	17	0.001	0.111	0.059	0.059	0.003	0.077
GOAL#17	10	1.000	1.000	0.700	0.900	0.824	0.947
GOAL#18	14	0.240	0.308	0.403	0.286	0.301	0.296
GOAL#19	12	0.240	0.758	0.403	0.431	0.301	0.550
GOAL#20	46	0.240	0.758	0.403	0.431	0.301	0.550
GOAL#21	33	0.717	0.709	0.328	0.595	0.450	0.647
GOAL#22	98	0.717	0.709	0.328	0.595	0.450	0.647
GOAL#23	42	1.000	0.906	0.405	0.691	0.576	0.784
GOAL#24	2679	0.740	0.688	0.453	0.497	0.562	0.577
GOAL#25	55	0.844	0.778	0.614	0.764	0.711	0.771
GOAL#26	33	0.844	1.000	0.614	0.152	0.711	0.263
GOAL#27	61	1.000	0.966	0.426	0.459	0.598	0.622
GOAL#28	80	0.414	0.624	0.363	0.663	0.387	0.642
Mean for all events		0.722	0.809	0.478	0.590	0.546	0.655
Unpaired t-test result		0.206		0.041		0.046	

5.4 Implications of Clustering on Event Detection Accuracy

We apply the burst detection method presented in Section 4.3 for the clusters generated by using IC and ICVE-SO on the same tweet set with 31 ground truth events (3 earthquakes and 28 goals). Since the described event detection method is not specific to a topic, this process results in events from a wide range of topics, such as political elections, TV shows, street protests, and news about celebrities. The results also include trendy topics or non-real world events. For example, a controversial tweet posted by a celebrity or a campaign started in Twitter can turn into a bursty topic. In our evaluations, we focus on the ground truth events by performing precision-recall analysis based on the false alarms and missed target events.

For this evaluation, we need to decide which detected event corresponds to which target event in the ground truth. Therefore, we check the detected events within two buckets after the time that a target event has happened and compare the top terms of the detected events with the topic and event-specific keywords of the target event. If there is a match, detected event is accepted to be the result of that target event. If no such event is found among the detected events, then this target event is considered as a "miss". On the other hand, there can be multiple events detected for a target event. The excess detections are counted as "false alarms". This would usually happen in case of fragmented clusters about an event. A single bursty term can be dispersed in multiple clusters, or distinct bursty terms about the same event may gather in distinct clusters. ICVE is expected to ameliorate such issues.

Because the accuracy of the burst detection algorithm depends on the *burst ratio threshold* Δ, we experimented with several values for it on the same dataset. The results of the accuracy analysis are presented in Figure 5.3. The figure shows that, miss rates obtained by using IC and ICVE-SO are usually close to each other. On the other hand, for all Δ values in our experiments, the number of false alarms is drastically reduced by using ICVE-SO.

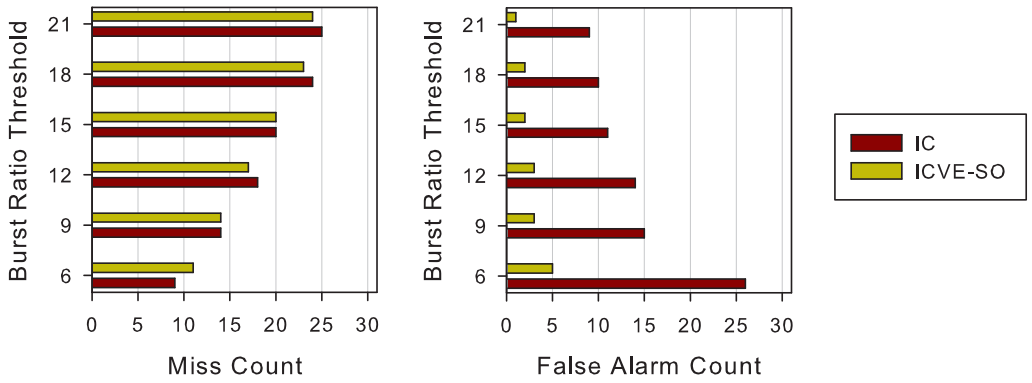

Figure 5.3: Event Detection Accuracy Analysis on Annotated Events

We present two illustrative examples to highlight the cases regarding the false alarms. The examples are selected from the burst detection results obtained by using $\Delta = 9$. The first example in Figure 5.4 shows histograms for the frequency of term "gol" in the tweet stream and three clusters. In the figure, the histogram (A) represents the number of tweets mentioning "gol" in each bucket received from the tweet stream for about 30 minutes. The goal event happens at around 19:45, which causes the burst in that histogram. The histograms (B) and (C) are the histograms for two event clusters associated with that goal event that are generated by the IC clustering method. One of these clusters has a centroid vector with a high weight for the term "gol". The second cluster's centroid vector has most of the weight on the name of the player. As a result of this fragmentation, the tweets that contain the term "gol" are grouped in two separate clusters, resulting in two separate alarms for the same event. One of these alarms is considered a false alarm. On the other hand, the event detection method using ICVE-SO generates one event cluster, with the histogram (D) in Figure 5.4. We observed that its centroid vector has close weights for "gol" and the name of the player. Compared to the histogram (A), it shows that almost all tweets containing the term "gol" could accurately be collected in a single cluster, resulting in the detection of the target event with no false alarms.

The second example presents the effect of cluster fragmentation on the results of event detection due to spelling variances. The target event in the example is

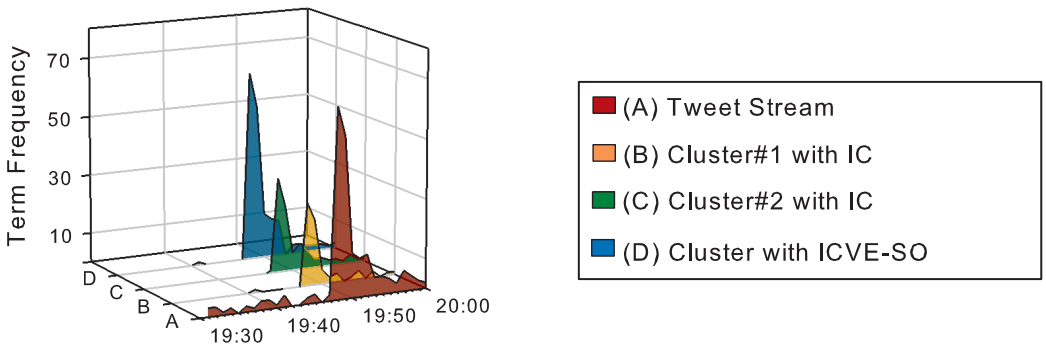

Figure 5.4: Distribution of the term "gol" in tweet stream and clusters before and after the goal event

a goal scored by a player named "oğuzhan" that resulted in a score of 2-0 in the game. It is observed that some Twitter users type the letter "g" instead of the letter "ğ" in the Turkish alphabet. Executing event detection using IC yields four separate event clusters about this event, with the centroid vectors mostly focusing on four different terms, namely "oguzhan", "oğuzhan", "gol", and "ozi". On the other hand, using ICVE-SO, we observe that a single event is detected by gathering relevant tweets in the same cluster with a centroid vector that has proportional weights for these event-related terms. This example suggests that ICVE-SO can successfully identify the differences in typing and group similar tweets about an event in the same cluster, resulting in increase in event detection accuracy.

CHAPTER 6

LOCATION ESTIMATION TECHNIQUES FOR EVENTS DETECTED IN TWITTER

This chapter presents a review of the current state of the art related to the location estimation techniques for events detected in Twitter. The analysis is conducted from several aspects. In Section 6.1, we present an overview of event types and the granularity of locations estimated for these events. We discuss the advantages, strengths, and challenges of spatial features in tweets, and different ways of using these features for event localization in Section 6.2. In Section 6.3, we introduce a classification of event localization methods in the literature and explain how we position our proposed location estimation method in this classification. We conclude this chapter by evaluation methods for location estimation in Section 6.4, examples of applications with a user interface in Section 6.5, and our discussion in Section 6.6.

6.1 Event Types and Location Granularities

In the implementation of an event localization system, it is necessary to determine the types of events to localize and the required spatial precision. Therefore, in this section, we first review and categorize the types of events targeted for localization in the literature and then categorize the types of their location according to their granularity.

6.1.1 Event Characteristics in Localization

Methods that aim to estimate event locations can be designed based on the topics of events since the topic affects the content and characteristics of spatial features in tweets. Examples of well-known topics in event localization from microblogs include natural disasters (e.g., earthquakes, floods, fires, and typhoons), sports events, outbreak of infectious diseases, traffic-related events, civil unrest, terrorist acts, weather events, conferences, exhibitions, and festivals. Solutions can also be proposed for and experimented with more generic event types, targeting multiple topics. For example, the term *local events* (also called *city events*) has been used to describe events restricted to a certain region [19, 48, 132]. These events can be related to various topics, such as local festivals, art exhibitions, or traffic accidents. Alternatively, the objective can be to localize topics that are discussed by many users in Twitter, without necessarily restricting the domain to a predetermined topic. This approach can be referred to as *open domain*, since it usually consists of topics mentioned on a large geographical scale. The location estimation method proposed in this thesis is also designed to operate in an open domain, and thus, experimented on events about various topics.

According to their temporal characteristics, events can be categorized as *breaking news*, *running events*, and *scheduled events* [53]. Another classification in [60] considers events as *forewarned* or *unexpected*. Temporal aspects of events are important for their localization, particularly for selecting the appropriate methods for data collection and analysis. For example, when an event is scheduled or predicted to happen in a region, the focus would be on tweets posted in that region before that event time (e.g., the Olympic games in a country or the weather forecast of a city). On the other hand, unexpected events and breaking news, such as earthquakes and traffic accidents, may require monitoring the tweet activity without any prior knowledge about the time or exact location of the event.

Another important set of criteria is the performance constraints related to the detection and localization of events. Online methods aim to process event-

related tweets in real-time usually to generate alerts shortly after an event has occurred. To this end, tweets can be retrieved from an online tweet stream or collected through frequent queries using the Search API. In both cases, the large volume of data arriving at high velocity is a major challenge to overcome. Alternatively, offline methods can be used to retrospectively identify events on an archived corpus of tweets. This is particularly useful in tracking and studying health-related incidents such as the spread of influenza. Offline methods are considered to have more flexibility since large volume of online data arriving at high velocity is not a major concern. There are also hybrid approaches that periodically collect and analyze tweets in sliding windows, such as in hourly or daily chunks [60, 94, 123]. Systems collecting and processing tweets immediately after natural disasters in sliding windows of several minutes can help authorities and rescue teams plan their actions accordingly in a timely manner.

6.1.2 Granularity of Estimated Locations

In addition to the determination of event types to localize, the granularity of the estimated location is a critical decision point in location estimation. The results can be at various granularity levels, ranging from a specific geographical coordinate to a larger region, such as a city or country. We classify the estimated locations into two groups according to their data types, namely *geographical coordinates* and *named locations*. Geographical coordinates describe the position of a location on Earth according to a coordinate system and named locations describe the locations by their names or addresses in a more human-understandable form [106]. Although transformation between these two types of locations is possible by using forward and reverse geocoding services such as GeoNames APIs[1], Yahoo Geo Services[2], and Google Geocoding Services[3], the targeted location type can affect the selection of spatial features and location estimation techniques.

[1] http://www.geonames.org [accessed 01 June 2016]
[2] https://developer.yahoo.com/boss/geo [accessed 01 June 2016]
[3] https://developers.google.com/maps/documentation/geocoding/intro [accessed 01 June 2016]

Geographical Coordinates: Geographical coordinates describe a location on Earth unambiguously and in a machine understandable way, which can be useful for plotting graphical representations of events on maps. These definitions can be further categorized as a single point, multiple points, or area, based on their granularity. In [48], authors use the term *point-events* for events that can be located at a point level, such as building fires, car accidents, and traffic congestion. The trajectory of a typhoon is studied in [108]. Although the location estimation technique in that work was designed to make single point estimations, when executed at discrete time intervals, it yields a list of points that can be interpreted as a trajectory. Event location at the granularity of an area can be defined in the form of a region (polygon) or a grid cell in a grid system. For example, in [13], the region of interest is mapped to a grid, and tweets are assigned to grid cells according to the location of the posts. The proposed method yields one of these grid cells as the event location. Similarly, Padmanabhan *et al.* display the event locations on multiple grid cells indicating where a disease was first reported and where it was later observed to provide an insight about the spread of diseases [94].

Named Locations: People generally refer to a location by its name, rather than its geographical coordinates. Therefore, depending on the application, it may be preferable and more practical to describe the location of an event in a human understandable form. A named location can be a country, city, town, street, or at the finest granularity, a Point-of-Interest (POI), such as a stadium or a concert venue. Named locations are typically the results of estimation methods that process spatial data in textual forms [82, 109]. For example, references to locations can be searched in the content of event-related tweets and the location that is mostly mentioned can be selected as the event's location [123]. Solutions that model the problem as a classification problem can also return a named location.

Table 6.1 presents a list of event types and granularity of estimated event locations targeted by the state of the art solutions. In order to avoid any ambiguity

Table6.1: Targeted Event Types and Granularity of Estimated Locations

Event Type	**Study**	**Granularity**	**Example Topics**
Open Domain	[42]	Grid/Region	TV shows, politics, sports
	[58]	City(s)	crimes, civil unrest, diseases
	[49]	Named Location	politics, epidemics, accidents
	[109]	Named Location	global news topics
	[38]	Named Location	TV shows, disasters, sports
	[123]	Named Location	diseases, disasters, cyberspace attacks
Local Events	[48]	Point	fires, traffic accidents
	[13]	Grid	public transport, weather, sewage, public safety
	[1]	Grid	sports, traffic accidents
	[132]	Grid	shop openings, market sales
	[19]	Region	parties, exhibitions, conferences, shows
	[27]	Region	disasters, sports, train delays
	[71]	Region	local festivals
Natural Disasters	[14]	Point	earthquakes
	[107, 108]	Point(s)	earthquakes, typhoons
	[82]	Named Location	floods, earthquakes
	[128]	Named Location	floods, fires
Epidemics	[94]	Grid	influenza like illnesses
Geopolitics	[117]	Region	civil unrest
Weather	[121]	Point	snow
	[120]	Region	snow, rain

related to event types that can be interpreted differently, we also list specific examples for targeted event topics as referred in the associated work.

6.2 Spatial Features for Location Estimation

As we mentioned in Section 2.1.2, tweets can contain several spatial features. However, the extraction of useful information from each of these attributes can pose different challenges. In this section, we first address the details of spatial features in tweets, discussing their strengths, weaknesses, and challenges. Then, we illustrate the different ways of using these features giving examples from previous studies.

6.2.1 Analysis of Spatial Features in Tweets

Among the listed three spatial features in tweets, GPS geotags are based on explicit geographical coordinates. On the other hand, tweet texts and user profiles require string processing for the extraction of useful spatial information. Each of these features exhibits certain characteristics that can affect the accuracy of location estimation and therefore require different spatial analysis techniques. Table 6.2 lists the advantages and challenges related to spatial features in event localization from tweets, and the remainder of this section discusses the methods to overcome these challenges.

Geographical coordinates: The most explicit and precise location of a tweet can be retrieved from its GPS geotag. Assuming that the tweets are posted by eye-witnesses or from places close to the event location, they can provide timely information even before the event is announced in other media channels. This information is useful to estimate the precise location of earthquakes or to locate city events such as traffic accidents or building fires in a timely manner.

One major problem about geographical coordinates in tweets is their *sparsity*. It has been reported that geotagged tweets constitute only a few percent of all tweets in Twitter [48, 60]. In fact, this ratio in a tweet corpus also depends on the way the tweets are collected. For example, tweets collected using geographical filters in the Search or Streaming APIs would contain a high ratio of geotagged tweets since the geographical filter looks for tweets satisfying the given criteria for coordinates. An alternative solution to compensate for the missing data due to the sparsity of geotagged tweets could be using the location attribute in the user profile for non-geotagged tweets, as suggested in [108].

Another challenge related to geographical coordinates is *information diffusion*, which, over time, affects the reliability of geotagged tweets for event localization [31, 108]. Basically, a user who posts a tweet about an event can be at a distant location at the time of the event [14, 48]. Once the event has been mentioned in Twitter or reported in traditional media channels, people at distant locations can

start posting tweets about that event. The location estimation method in [108] presumes that users are independent and identically distributed, meaning that their tweets do not affect each other. The authors analyzed information diffusion in several types of events based on the social relationship between the users, and suggested that the assumption of independent and identical distribution is valid for specific types of events such as earthquakes and typhoons. On the other hand, some studies have disagreed with such assumptions [27]. One straightforward solution to handle information diffusion in the localization of an event is to use only the first few tweets posted about that event [108]. Alternatively, searching for specific predetermined words in tweets is suggested in [53] in order to identify users that are eye-witnesses to an event.

The usability of geotagged tweets can also be hampered by differences in the *distribution of users* in a region [14, 90, 108]. Since the concentration of population in urban areas is much higher than that in rural areas, unless handled accordingly, methods employing geotag information would tend to favor densely populated regions as the locations of events. In order to minimize this bias, Sakaki *et al.* recalculated weights in their algorithm based on a distribution of randomly selected Twitter users that reflect the population distribution in the region of interest [108]. In [79], authors discussed measures that can be used for normalization, such as the population, number of Twitter accounts, or event-specific measures. The work in [71] handled the effect of heterogeneity in population distribution differently by defining possible event locations as regions of clustered coordinates. As a result of this clustering, densely populated areas were represented by small fine-granular regions whereas sparsely populated areas were covered by larger region definitions. In this thesis, we also address the effect of population differences on the evidence obtained from GPS coordinates in a region (see Section 7.7), and apply a normalization for the probability values when estimating an event's location [90].

Tweet Text: Location name, an important component for describing an event, can frequently be mentioned in tweets. Since a tweet text is a free-text field

Table6.2: Advantages and Challenges of Spatial Features in Tweets

Feature	**Advantages**	**Challenges**
Geographical Coordinates	+ timely information + precise	– sparsity of geotagged tweets – information diffusion – heterogeneous distribution
Tweet Text	+ various granularity levels + can provide more than one location + part of event description + can be more useful over time	– geoparsing – non-standard writing – ambiguity
User Profile	+ shorter in length + designed for location information	– not strongly typed – mostly coarse-grained data – can be outdated – ambiguity

limited to 140 characters, people can type any location name related to the event, which may not necessarily be a single location. They can even describe the event referring to locations at various granularity levels, such as country, city, district, or even street name or block number, depending on the event type. Regarding the temporal aspects, information diffusion may not be a concern for tweet texts. On the contrary, with the diffusion of information over time, tweets become "news about an event" containing the location name even more frequently [31].

The challenges involved in estimating the location of an event from a tweet text are essentially related to accurately identifying the georeferences in the text, i.e., *geoparsing* [54]. Geoparsing techniques have long been studied as part of linguistics, machine learning, and information retrieval [12, 76]. The sub-problems in geoparsing can be listed as finding location names in a text (toponym recognition) and resolving possible Geo/Geo ambiguities (toponym resolution). However, these problems are even harder to solve in tweets due to their brevity, frequent spelling idiosyncrasies, deviation from grammatical rules, and non-standard abbreviations and writing conventions.

There are two basic approaches to identifying the place names in a given text; 1) using NLP techniques to analyze the structure of the text, and 2) scanning the text to search for place names listed in a glossary or gazetteer [9, 12]. The NLP

techniques generally aim to analyze each word (token) of a given sentence using a language-specific POS tagging process, and to detect groups of tokens that are likely to refer to named entities [47, 82]. The Stanford Named Entity Recognizer is widely used for this purpose [60]. However, tools that have been experimented with controlled and grammatically correct textual items may not perform well in tweets due to the distinct characteristics of tweets mentioned before. Despite the enhancements to the NLP methods to handle linguistic peculiarities, they still present several drawbacks in the Twitter environment [73, 76, 105]. Considering the high volume and velocity of the data to process, probable computational complexities in NLP techniques can make it difficult to keep up with the speed of the incoming tweet stream. Moreover, some of these techniques may require training specific to the language of the text, as well as to the poorly structured content [82].

Being simpler and usually faster than the NLP techniques, gazetteer-based approaches constitute practical and convenient alternatives for identifying the location names in tweets [12, 96]. Terms in a text can be looked up in a glossary of location names to find related references. In this regard, resources that provide Volunteered Geographic Information (VGI) can be considered useful gazetteer databases. GeoNames, OpenStreetMap[4], and WikiMapia[5] are among well-known sources that allow downloading spatial definitions (e.g., cities, towns, streets, buildings, mountains, rivers) in many countries in a variety of data formats [50, 106]. However, if a location definition is not available in a gazetteer, it is not possible to find it in the text. Moreover, even if a location name is found in the text, it may have another non-geographical meaning in the context of that text [12, 62]. In order to resolve such Geo/Non-Geo ambiguities, a spatial indicator such as "in" before the ambiguous term can be searched [96]. Alternatively, an additional geocoding can be applied on the detected named entities to extract location references [82]. Further heuristics were described in [62] to resolve Geo/Non-Geo ambiguities. Although gazetteer-based approaches

[4] http://www.openstreetmap.org [accessed 01 June 2016]
[5] http://wikimapia.org [accessed 01 June 2016]

minimize language dependency, the complete ignorance of the grammar can restrict the geoparsing performance, especially for agglomerative languages such as Turkish [82]. Another argument against gazetteer-based approaches is related to the use of phrases such as "x miles away form y", which requires the analysis of the complete sentence structure and thus cannot be pinpointed to a location by simply looking up the location names in a database. On the other hand, it is arguable whether it is worth making a deep analysis of such phrases since they are rarely observed in tweets, and the location name alone can still be useful to provide an idea about an event [128].

Another ambiguity related to the location names in texts is the Geo/Geo ambiguity, which results from several distinct places having the same name, such as "Paris" having 140 distinct possibilities [109]. In order to determine which location is meant by the user in a specific tweet, Teitler *et al.* used geographical distance, document distance, and hierarchical containment between the locations mentioned in the text [122]. The authors' proposal was based on the idea that location references in an article should provide evidence that is consistent with each other. In [62], possible locations for an ambiguous term are ranked by scoring them according to their textual and geographical support for each other. In [109], authors used all the tweets in a topic cluster to resolve ambiguities based on the relationship between the possible interpretations of toponyms in that cluster, such as the geographic containment, document distance, and geographical distance. A similar approach is discussed in [48]. In that work, the authors also combined coarse-grained location information in the user profile with fine-grained location names in tweet texts to resolve any ambiguity in fine-grained definitions. In another study [96], distances between user locations and possible interpretations of the ambiguous location names are compared to perform resolution.

Finally, it should be considered that there may not be any useful location references in tweets, particularly for events that have only been reported by few users. Moreover, even if all locations in a tweet are identified, there can be noisy

tweets that can distract the event localization process. For example, a sports team of city A can play a game in city B, and their supporters can post tweets containing the name of city A, which is actually not the event location. Therefore, to improve the accuracy of results, location names can be supported by further evidence from other tweet features or by performing additional analysis based on the event type.

User Profile: The location attribute in the user profile allows Twitter users to specify their locations in a free-form text field. Different from the tweet text, this attribute can contain 30 characters at most. Since users are specifically asked to enter a location name in this field, and provided with a limited space, complicated sentence-like phrases are mostly avoided. This is also consistent with the findings in [52], reporting that 66% of profiles contain valid geographical information.

Despite the advantages of profile location, it should not be presumed that this field always contains accurate geographical information. Since it is a free-text field, users can enter non-geographical texts, and thus, issues related to geoparsing are still a concern. Due to its brevity, passing the content in the profile location to public geocoding services in order to search for geographical descriptions can provide useful results, as experimented in [2, 137]. Another characteristic of profile location is its granularity. It is suggested that people tend to specify coarse-grained locations in their profiles, such as city or state names [48]. Therefore, due to the lack of fine-grained information such as a street name, profile location may not provide satisfactory results in locating certain events such as building fires or traffic incidents.

The temporal characteristics of profile location mean that this information may not be updated frequently. In the profile location text field, the users may specify their hometown or the place they used to live, or even some other location, to which they feel attached. Therefore, there is a risk of obtaining outdated and misleading information from user profiles. On the other hand, the user profile can be considered to provide spatial evidence that does not heavily depend on

time. In other words, we can expect users to be interested in events occurring in the location specified in their profile. No matter when the event has occurred, users having the event's location in their profile tend to post tweets about that event, irrespective of their current location.

6.2.2 Usage of Spatial Features in Tweets

Our analysis of spatial tweet features revealed that it may not always be possible to obtain high quality and reliable spatial information from the selected features. That means, the possibility of missing information and possible inconsistencies between different information sources should be taken into account. Tweet features can be selected and used in different ways depending on the requirements and expectations. We classify the spatial tweet features as "primary" and "secondary" according to their usage in a location estimation algorithm. A primary spatial feature is the one that is first inspected in tweets and used in the location estimation algorithm. An application can choose to use a single primary attribute or a combination of multiple primary attributes. Secondary features, on the other hand, play a complementary role for the tweets that lack the expected primary attribute. For example, an application can primarily use the GPS geotags of tweets, and for the tweets that are not geotagged, it can exploit the user profile as a secondary feature to infer the location of that tweet [108]. We reviewed and categorized the work existing in the literature based on their choice of features as follows; 1) using a single (primary) feature, 2) using secondary features to handle the missing features, and 3) using a combination of multiple features as primary sources. In Table 6.3, we present the results of our analysis of the spatial features used in the studies we reviewed. This table also includes the selected tweet collection method, which may affect the quality and availability of content in spatial features.

Using a Single Feature: Methods based on the use of a single feature determine one of the features in tweets as the source of spatial evidence and only use this feature in location estimation. For example, in [123], only the location

Table6.3: Usage of Spatial Features (P):Primary, (S):Secondary.

Study	GPS Coor.	Tweet Text	User Profile	Tweet Collection Method
[1]	P	-	-	Stream (Geo.)
[13]	P	-	-	Stream (Geo.)
[19]	P	-	-	Stream (Geo.)
[27]	P	-	-	Stream (Geo.)
[71]	P	-	-	Search (Geo.)
[120]	P	-	-	*Not specified*
[121]	P	-	-	*Not specified*
[82]	-	P	-	Stream (Keyword)
[123]	-	P	-	Search (Geo.)
[128]	-	P	-	Search (Keyword)
[58]	P	S	-	Search (Keyword)
[132]	P	S	-	Stream
[42]	P	-	S	Stream (Keyword)
[107, 108]	P	-	S	Stream (Keyword)
[94]	P	-	-	Stream (Geo.)
[38]	P	-	-	Stream (Geo.)
[117]	P	P	-	Stream (Geo.)
[48]	-	P	P	Search (Keyword)
[49]	-	P	P	Search
[109]	-	P	P	Stream (User)
[14]	P	P	P	Search (Keyword)

names in tweet texts are used to determine the event location. A spatiotemporal clustering is applied to tweets using their GPS coordinates in [13]. Similarly, in [1], the authors utilized the GPS coordinates of event-related tweets to represent the spatial distribution of bursty keywords in order to detect events and estimate event locations.

Using Secondary Features: A location estimation method can benefit from multiple features in a way that if the primarily selected feature is not available in a tweet, other features can be used as secondary sources to extract the location information for that tweet. This arises from the fact that tweets can have missing or unusable values in a primarily selected spatial feature. Users can choose to hide their coordinates in their tweets, and they may not specify any location in their profiles. One of the common practices in using secondary features is utilizing the location attribute in the user profile in lieu of GPS data for non-

geotagged tweets [2, 108, 137]. That means, the coordinates of a non-geotagged tweet can be inferred from the coordinates of the location specified in the user profile. A variation in the usage could be to randomly select a latitude-longitude pair for a non-geotagged tweet among the coordinates of geotagged tweets having the same profile locations [42].

Location references in a tweet text can be considered an alternative to GPS geotag. For example, the coordinates of geotagged tweets are used as a primary source of information in [79], and for non-geotagged tweets, the authors detected geographical references in the tweet text and searched them in GeoNames to obtain the latitude-longitude values. Similarly, the location for a non-geotagged tweet was inferred using common hashtags and mentions with the geotagged tweets in [58].

Using Multiple Primary Features: A third option regarding the use of spatial features in tweets for event localization is utilizing multiple features as the primary sources of information. Such methods do not consider features alternatives to each other; rather, they combine them in a single model. For example, in [48], POS tagging is applied to tweet texts to extract fine-grained location information. Claiming that coarse-grained locations are usually available in the profile rather than in a tweet text, the authors combined profile location with fine-grained information in tweet texts for a more specific description of places. The location estimation method presented in [49] creates a graph containing the events, users, and location names in tweets based on their relationships. This graph is then used to make estimations for all events maximizing the probability of the observed evidence. In another study combining multiple spatial features [14], location estimation was carried out for events detected in Sina Weibo, a Chinese microblogging service similar to Twitter. The microposts in Sina Weibo are called weibos, which can be considered counterparts of tweets in Twitter. In that work, coordinates were extracted from weibos using their GPS geotags, location references in weibo texts, and user profiles. In the location estimation process for an event described by a set of weibos, coordinates extracted from

these features are treated equally in a hierarchical clustering algorithm. The location estimation method we propose in this thesis also uses multiple primary features, as explained in Chapter 7. We assign basic probability values to possible event locations based on GPS geotags, tweet texts, and the user profile, first separately; and then, combine these probability assignments in a single solution using combination rules in DS theory [90].

6.3 Location Estimation Methods

Having discussed the event types that can be localized, their location granularities, and the use of spatial features in tweets, in this section, we classify event localization techniques applied for Twitter, and discuss their strengths and limitations. In our classification, we consider these techniques in two main groups; event-pivot and location-pivot methods. This distinction is made according to the association of event detection and location estimation tasks, i.e., the order in which they are performed [53]. The event-pivot approach primarily focuses on the detection of events. Such methods perform spatial analyses to estimate the locations of detected events at a later stage, mostly by using a set of tweets identified for the events. The location-pivot approach first carries out a spatial analysis of the tweets, trying to detect any abnormal tweet activity associated with a region. Once an active region is identified, events are detected at a subsequent step. In the remainder of this section, we explain the specific techniques in these two groups in more detail.

6.3.1 Event-Pivot Methods

Event-pivot methods estimate a location for events after their detection. Tweets can be collected by one of the tweet collection methods in Twitter, e.g., using keywords of a specific topic [107, 108], tweet coordinates [90], or the posts of some handpicked trustworthy users [109]. An event can be detected by clustering tweets according to a similarity measure, or by finding bursts in the selected

tweet features, as discussed in Section 3.1. Whether it is a clustering or a burst detection method to detect events, event-pivot location estimation methods expect a set of event-related tweets, and aim to determine an aggregate geographical focus for these tweets using their spatial features. We categorize event-pivot techniques into the following four groups according to the adopted method for estimation; 1) basic statistics, 2) spatial clustering, 3) probabilistic approaches, and 4) Bayesian filters. These methods can vary in their ease of implementation, computational complexity, capability to execute online, and assumptions about the tweet and event characteristics. Table 6.4 presents the list of studies that applied these techniques to event localization in Twitter, together with their strengths and limitations addressed in each study.

6.3.1.1 Basic Statistics

In its simplest form, the location of an event can be estimated by executing basic arithmetic operations on the available data. For example, given a set of event-related tweets, after identifying location names mentioned in these tweets, a straightforward approach can be *maximum voting* or *majority voting*, which first counts the number of references to each location and then selects the mostly mentioned location as the event location [48, 49, 123]. This method has been extended in [109] using a weighing scheme based on spatial proximity. In order to assign a geographical focus to a set of clustered tweets, the authors first performed a tf-idf analysis on the user profiles and tweet texts, and looked up distinctive phrases extracted from these attributes in the GeoNames database. The geographical focus to the cluster is then assigned by calculating a score for each location mentioned in tweets and user profiles of the cluster, and selecting the location with the highest score. In these calculations, scores are calculated based on the frequencies of locations. Moreover, if a Geo/Geo ambiguity is observed, the frequency of the ambiguous location is divided into each of its possible interpretations, and those that are close to other locations in the cluster are re-weighed to increase their scores.

Table6.4: Event-Pivot Techniques

	Study/Technique		Strengths & Limitations
Basic Statistics	[109] Majority Voting *online*	+	describes an unsupervised toponym recognition
		+	uses containment and proximity for ambiguities
		−	relies on posts from specific seeders
	[123] Majority Voting *sliding windows*	+	estimations at multiple spatial granularity
		−	uses language-specific NER and POS tools
		−	requires gazetteer for the region of interest
	[48] Mean of Coordinates *sliding windows*	+	combines coarse- and fine-grained information
		−	requires training and grammar rules for addresses
		−	prevalence of profile location affects accuracy
Spatial Clustering	[1] Clustering of Signatures *sliding windows*	+	custom grid bandwidth for targeted event type
		+	adaptable thresholds for event locality/globality
		−	multiple locations can be found for the same event
	[14] Hierarchical Clustering *retrospective*	+	uses three spatial features primarily
		−	lower accuracy for sparsely populated regions
		−	expects at least one location in tweet text
	[58] Fast Spatial Scan *retrospective*	+	analyzes using keyword, event type, country, time
		−	requires city data and their neighborhood
		−	cluster significance threshold affects accuracy
Probabilistic App.	[49] Expectation Maximization *sliding windows*	+	jointly estimates user and event locations
		+	estimations at multiple spatial granularity
		−	assumes that users have location affinity
		−	requires approximate localization at the beginning
	[28] Maximum Likelihood Est. *not specified*	+	no external data source (purely tweet content)
		−	needs probabilistic distribution of words over cities
		−	few event tweets can result in low accuracy
		−	assumes independence of words in event tweets
Bayesian Filters	[107, 108] Kalman Filters *online*	+	updated estimations based variance of observations
		−	assumes independent and identical distribution
		−	assumes a single instance of a target event
		−	requires for low uncertainty and linear dynamics
	[107, 108] Particle Filters *online*	+	uses sampled weighted particles updated iteratively
		+	weighs particles to handle uneven user distribution
		−	assumes independent and identical distribution
		−	assumes a single instance of a target event
		−	more particles increases execution time

Finding the mean and median values of GPS coordinates is used as baseline methods in [107, 108] to estimate earthquake epicenters. The mean of latitude-longitude values for localizing building fires and traffic incidents is calculated in [48]. In that work, the authors argued that the location of an event can be different from the location of the tweet source, and hence, rather than us-

ing the GPS coordinates, they used spatial references in tweet texts and user profiles in a combined manner to extract geographical coordinates. For a set of event-related tweets, the authors identified fine-grained information such as landmarks or partial addresses in tweet texts by applying POS tagging. At the same time, the coarse-grained location for the event was determined by majority voting of the profile locations. The combination of these two types of information is expected to yield address-like definitions, which are further resolved into latitude-longitude pairs via Google Maps APIs. After filtering out the outliers, the retrieved coordinates are averaged to pin-point the event location.

These methods can usually be executed in real-time since they do not require complex calculations. In addition, they can also yield useful estimations under certain circumstances. For example, if we assume that tweets are posted only around the correct event location without any remarkable noise, calculating the mean of coordinates can result in accurate estimations. Similarly, majority voting based on tweet content expects users to frequently post tweets containing the name of the event location. Such assumptions can reasonably be made depending on the event type.

6.3.1.2 Spatial Clustering

Spatial clustering methods estimate event locations by performing a clustering analysis on the geographical coordinates extracted from event-related tweets. The event detection framework presented in [58] expects a query text and a country name to search for events in a given tweet corpus. In order to find events and event-related tweets in the corpus, distinctive labels in the form of named entities and action words for events are identified in tweets by leveraging news items in public media. For the localization of a detected event, each tweet is assigned to a city for the given country according to the GPS coordinates. For those that lack GPS information, the coordinates are estimated by analyzing the commonalities in the content of geotagged and non-geotagged tweets. Based on these tweet-city assignments, tweet counts are calculated for each city and then

used in a spatial scan method, namely the *fast subset scan* [85]. The algorithm scans clusters of neighboring cities, calculates a score for each cluster, and selects those with the highest scores as candidate locations for the event. Finally, the significance of clusters is calculated via random permutation testing, and clusters with significance higher than the threshold are selected as the location of the event.

In [1], the region of interest is modeled as a grid with a predetermined bandwidth. The authors collected the tweets posted in that region from the tweet stream, and processed them in time-based sliding windows, assigning geotagged tweets to the grid cells according to their coordinates. The proposed event localization method aims to identify remarkably mentioned words in tweets in a grid cell over a specific time period. To this end, keywords exhibiting bursty behavior are detected by keeping a history of word frequencies. Based on the frequencies in grid cells, *spatial signatures* are generated and then used to calculate entropy values for keywords. Keywords with small entropy are selected as words that occur at a few locations (rather than those that are widely spread over the region). The claim is that keywords related to the same event should have similar spatial signatures. Hence, a single-pass clustering algorithm is applied to keywords according to their spatial signatures to obtain spatially similar keywords in clusters. Event-related clusters are distinguished from those generated due to spurious words based on cluster scoring, and the location for an event cluster is found by averaging the spatial signatures of the keywords in that cluster.

The set of event-related weibos is obtained by performing a keyword search in [14]. The authors extracted geographical coordinates in terms of latitude and longitude from the locations mentioned in weibos, GPS geotags, and locations specified in user profiles. In order to detect the location of earthquakes in their experiment, the authors applied *hierarchical clustering* to the coordinates based on their Euclidean distance to each other.

Determining thresholds is usually a prerequisite for location estimation techniques using spatial clustering and can affect the accuracy of the estimation

results. Additionally, the execution time of clustering can be a concern depending on the size of the dataset and expected response time. According to our observation, most spatial clustering approaches are retrospectively applied. Estimated locations are usually at the granularity of regions, grids, or named locations, which can be plotted on maps using heatmap-like figures. Further processing may be required to obtain more precise estimations (e.g., calculating the cluster centroid for point-level granularity).

6.3.1.3 Probabilistic Approaches

Probabilistic approaches aim to determine the most likely location for a given event based on probabilities calculated for each location in the region of interest. A probabilistic location estimation method that jointly locates both events and users is proposed in [49]. The authors periodically collected and processed tweets in time windows, identifying pairs of keywords with high information gain, and grouping tweets containing these pairs into event clusters. The location estimation technique they proposed is an *Expectation Maximization* (EM) method, which models the events, users, and location references in tweet texts as nodes in a graph, and connects them with each other based on their relationships. Each event in the graph is associated with a latent variable. The method aims to estimate the unknown parameters in the graph that maximize the probability of observations. The location with the largest value of latent variable for an event is selected as the location of that event. This method requires some of the events to be approximately localized at the beginning. For this reason, event locations are initialized by applying the maximum voting method to the location names mentioned in event-related tweets and user profiles. Estimation is separately performed for each level in the location hierarchy, namely the country, state, city, and street.

Another probabilistic method, *maximum likelihood estimation*, is presented in [28] in order to estimate user locations, rather than the locations of events. This was experimented as a baseline method to perform event localization in [49].

The proposed method is based on learning a probability distribution of words in tweet texts over the cities in a country. In order to estimate the location of an event associated with a set of tweets, the posterior probability of the event belonging to a city is found for each city based on the probability distribution of words in tweets. The city with the maximum posterior probability is then selected as the event location.

Considering the two probabilistic methods discussed above, probabilistic approaches generally estimate event locations in terms of discrete named locations, rather than points or regions. These methods require a set of choices that represent possible event locations to determine probabilities.

6.3.1.4 Bayesian Filters

Bayesian filters estimate the state of a system in a probabilistic manner according to the observations of sensors received at discrete times [44]. These filters are generally adopted for estimating object locations in ubiquitous computing. Two implementations of Bayesian filters, namely *Kalman filters* and *particle filters*, are applied in [107, 108] to estimate the epicenter of earthquakes and trajectory of typhoons detected in Twitter. The authors detected such events by monitoring tweets containing specific keywords, such as "earthquake" and "shaking". They regard each Twitter user posting an event-related tweet as a sensor, and used the GPS geotag and user profile in tweets as primary and secondary spatial features, respectively in order to extract latitude-longitude information. Kalman filters estimate an event's location by applying an update rule based on its previous estimate and using the coordinates of the most recent tweet. It represents uncertainty by mean and covariance in the distribution of coordinate values. However, as stated in [107], Kalman filters work better in a linear Gaussian environment, and their use in event localization problem may yield poor performance since tweet dynamics are not necessarily linear and tweets posted from distant locations can have a negative effect on the variance.

For non-linear tracking problems, particle filters are considered more suitable than Kalman filters. An implementation of particle filters using a Sequential Importance Sampling (SIS) algorithm is described in [108]. In their implementation, particles distributed randomly around the region of interest were generated at an initial generation stage. For each new tweet providing latitude-longitude values, the algorithm iteratively samples new particles, updates particle locations, and assigns weights to the particles based on their distance to the most recent observation. The estimated location for the corresponding event is finally determined as the mean of particle coordinates.

In the above-mentioned studies, the authors argue that for the proposed location estimation methods to work accurately, the users posting event-related tweets should be independent and identically distributed, i.e., the information about an event should not diffuse much in Twitter. The authors present a diffusion analysis for several event types, and claim that the sensors can be assumed to be independent and identically distributed for earthquakes and typhoons. They also address the bias introduced by differences in the geographic distribution of Twitter users. Hence, they introduce an enhancement for particle filters assigning weights to the particles taking the sensor distribution into account based on a randomly sampled dataset. On the other hand, although the experiments demonstrate the superiority of particle filters over Kalman filters in terms of estimation accuracy in event localization, the execution time for particle filters may not be suitable for real-time alarm generation depending on the number of particles used and the settings in the proposed enhancement.

6.3.2 Location-Pivot Methods

Location-pivot methods prioritize the spatial analysis of tweets over event detection in the event localization process. More specifically, after the identification of regions with abnormal tweet activities, it can then be analyzed whether these activities are due to an event in that region at a posterior stage. As a result, location-pivot methods do not explicitly perform location estimation for a

detected event since the location aspects of an event would have already been analyzed. We categorize location-pivot techniques into the following four groups; 1) methods that collect tweets in a location-oriented manner, 2) activity analysis in partitioned regions, 3) burst detection, and 4) spatial clustering. Table 6.5 presents an overview of studies that have applied these techniques.

6.3.2.1 Location-Oriented Tweet Collection

The simplest and the most straightforward location-pivot method is probably collecting and processing tweets that are known to have been originated from a possible event location. Since tweets related to a specific region can be collected either through geographical filters or providing location names as keywords, the problem can be related to deciding whether there is any event in that region. For example, Stefanidis *et al.* collected tweets labeled with the Tahrir Square hashtag that had originated from Cairo (within a 10 km radius from Tahrir Square) [117]. Monitoring the hourly tweet counts collected this way, the authors could identify events related to that location, which they referred to as a "geographical hotspot".

In another study [128], the authors aimed to increase situation-awareness by following tweets posted about two natural hazard events that occurred in 2009, namely the Red River Floods and the Oklahoma Grassfires. The focus of their work was to identify the information that might be extractable from Twitter at the different stages of hazard events. The authors performed a retrospective analysis on these two events by searching for tweets with location and event-related keywords (e.g., "oklahoma", "okfire", "red river"), and selecting the tweets of users that were geographically close to the event regions according to user profiles. Although the objective of the work was not exactly event detection, the authors' tweet collection approach can also be used to detect predicted events in a specific location.

Despite their simplicity, the methods explained above require prior knowledge

Table6.5: Location-Pivot Techniques

	Study/Technique		Strengths & Limitations
Loc.-Or. Coll.	[117] Coordinates-based collection *sliding windows*	+	helps monitoring location for a predicted event
		−	location to monitor should be decided beforehand
	[128] Content-based collection *retrospective*	+	helps monitoring location for a predicted event
		−	location to monitor should be decided beforehand
		−	manual selection of on-topic tweets and users
Activity in Partitioned Regions	[132] Frequency Threshold *sliding windows*	+	custom grid bandwidth for targeted event type
		−	accuracy depends on the thresholds and grid size
		−	requires generation of place name database
	[13] Frequency Threshold *sliding windows*	+	custom grid bandwidth for targeted event type
		+	analyzes spatial, temporal, thematic coherence
		−	each grid is assumed to have a single event
		−	accuracy depends on the thresholds and grid size
	[71] Box Plot Statistics *sliding windows*	+	combines movement patterns and frequencies
		+	partitions areas to handle uneven distributions
		−	requires historical tweet set for training
		−	tuning of thresholds for partitioning and boxplots
	[82] Location Name Frequency *sliding windows*	+	geoparses tweets in real-time using VGI resources
		−	training with historical data and locations names
		−	thresholds need to be tuned
Burst	[121] Location-Based Burst Det. *retrospective*	+	can detect events with respect to a point
		−	must be executed for each possible event location
Spatial Clustering	[27] Space-Time Scan Statistics *retrospective*	+	all possible space-time windows are analyzed
		+	performs an efficient significance test for clusters
		−	missed events with no geographical focus
		−	memory usage for large tweet sets
	[42] Spatiotemporal Hashtag Cl. *online*	+	can efficiently handle evolving hashtags online
		−	expects user queries for a specific space-time scale
		−	memory usage for large tweet sets
	[19] DBSCAN *sliding windows*	+	uses spatial proximity and content similarity
		+	identifies patterns in location and word statistics
		−	training for the local event classification
		−	tuning of constants and time intervals
	[120] (ϵ, τ)-Density-Based Cl. *retrospective*	+	finds spatially and temporally separated clusters
		+	can find areas of arbitrary shape
		−	tuning of thresholds for targeted event type

or prediction about approximate event locations. Having decided on the region to monitor, the outcome of event detection would mostly be determining the event time and obtaining event-related tweets. Once an activity is detected,

more resources can be allocated in a timely manner to better monitor the event.

6.3.2.2 Activity Analysis in Partitioned Regions

Another method to detect regions with abnormal tweet activity is to count the number of tweets related to each region and identify those with remarkable tweet frequencies. The techniques can vary in how regions are represented, how tweets are associated with the regions, and how an abnormal activity in a region is determined. For example, aiming to detect local events in a city, Watanabe *et al.* modeled the region as a grid with cells of 20x30m^2 and applied a geohash function to the GPS coordinates in order to map each tweet to a grid cell [132]. In that work, the authors extracted event-related key terms from the tweets and counted the frequency of each key term in grids. If a key term appeared three or more times in the tweets associated with a grid cell, that grid cell and key term were marked as indicators of a local event. This method can be adapted to detect major/minor events in large/small regions by changing the grid size and expected number of tweets in grid cells. Additionally, the authors proposed a method to infer geographical coordinates for non-geotagged tweets by searching for patterns of specific place names in the content of tweets.

In a similar study [13], the authors assigned tweets to grid cells in a city by geohashing their GPS coordinates. Given a set of tweets posted for a duration of time, the authors defined feature vectors for each tweet using their geohashed location, event and location terms in tweet contents, and their posting times. These feature vectors were then grouped according to their spatial, temporal, and thematic coherence, and the groups with tweet counts above a certain threshold were marked as events.

As an alternative to using static thresholds to identify an unusual tweet activity, past tweet statistics can be utilized in a training phase to obtain information about the normal distribution of tweets in a region. Remarkable deviation from the expected patterns can then be interpreted as the indicators of events in that

region. [71] presents a geo-social event detection method by monitoring crowd activities in terms of three indicators in a region; the number of tweets, the number of users, and the number of movements into or out of the region. Using a historical set of tweets posted in a country, regularities in these indicators were retrieved over 6-hour time windows in the form of boxplots, describing five descriptive statistics (minimum, maximum, lower quartile, upper quartile, and median). In the execution of event localization, values observed in three activity indicators are compared with the historical statistics, and when a certain combination of irregularities is identified, the region is marked as the location of an event. For example, an abnormal tweet count in a region alone may not necessarily be the result of an event; but if it is also accompanied by a remarkably high number of movement activities into and out of the region, this can indicate an event in that region. It is noteworthy that in order to represent the regions of interest in the country, the authors applied a space partitioning method using K-means clustering on the geographical coordinates, and formed a Voronoi diagram. In order to handle heterogeneously distributed population in a country, this can be considered a more appropriate approach than defining the regions in terms of equal-sized grid cells or administrative regions.

In [82], tweets mentioning natural disasters in risky areas are collected by searching for specific event-related keywords (e.g., "flood"). In the offline stage, the authors prepared a geospatial database aggregated from multiple VGI sources. They devised a geoparsing technique to extract location references from the tweet texts, and applied it to a historical tweet set in order to obtain baseline statistics for the number of location references over 5-minute tweet windows in which no disasters occurred. In the event detection system, the authors calculated the simple moving average and the triangular weighted moving average for the location names in the current time window, and identified locations that were mentioned significantly more than the baseline frequencies (higher than a threshold).

The methods described above mostly require training data and thresholds to be

adapted according to expectations from event localization. Although there are solutions that can easily be implemented, if the criteria for an abnormal activity are not very strict, a large number of events could be detected, most of which would probably be false alarms. These methods are usually applied for the retrospective and window-based analysis of tweets since thresholds are mostly based on an assumption of tweet volume and abnormal activities in a region are identified based on statistics in fixed time windows. Another point is related to the interpretation of events after the detection of active locations. For example, the reviewed studies did not thoroughly discuss whether two alerts detected in two neighboring regions or two consecutive time slots should be interpreted as referring to the same event. In this case, post-processing including temporal and semantic analyses may be necessary.

6.3.2.3 Burst Detection

As mentioned in Section 3.1, burst detection is a widely adopted method to detect events in a temporally ordered series of tweets. [121] introduces a *Location-Based Burst Detection* method that extends the event-oriented burst detection method proposed in [67] by incorporating a spatial proximity aspect to the frequency-based burst analysis. The algorithm in [67] finds discrete time periods in which the number of documents including a specific keyword is higher than expected. In that work, a burst is modeled as a state transition from low (non-bursty) to high (bursty) state, associated with a cost function. The burst detection problem is then solved as an optimization problem to find the minimum-cost state sequence. The cost function is extended in [121] by introducing a new component called *influence rate*, which enables location-based burst detection. The algorithm is executed to decide whether a burst is observed at a specific user location. The influence rate component favors locations that are close to the analyzed tweets using the distances between the selected location and the GPS coordinates of tweets. The authors evaluated this method on snow events in five major cities in Japan.

The proposed location-based burst detection method can detect remarkable mentions of a keyword with respect to a point, i.e., a latitude-longitude pair. If there are multiple locations to analyze, the algorithm has to be executed for each point separately. Moreover, since this method analyzes state transitions in a time series, it also requires a list of recent tweets to model their distribution in discrete time periods.

6.3.2.4 Spatial Clustering

Spatial clustering methods aim to detect densely populated tweet regions according to their spatial proximity. In [27], events and their location are detected by clustering geotagged tweets according to their posting time and GPS coordinates. The authors used the *Space-Time Permutation Model* of the *Space-Time Scan Statistics* technique, which views tweets in a spatiotemporal data cube and analyzes them in cylindrical windows of varying space and times regardless of the tweet text. In that work, each space-time window of the data cube is considered a potential cluster. The process compares the observed number of tweets with the expected number of tweets in each window, testing whether there is a statistically significant increase. If a significant space-time cluster is found in the data cube, the content of tweets is analyzed to extract keywords describing the topic of the cluster. If a cluster containing keywords that are attributable to a real-world event is found, it is marked as an event cluster, and its location and time are calculated using the space-time properties of the cluster in the data cube.

An event detection framework called *STREAMCUBE* that aims to detect hashtag clusters from the tweet stream in real time is proposed in [42]. In that work, the clusters are organized in data cubes based on a spatial and temporal hierarchical structure. The highest level in the spatial hierarchy is the globe, which is divided into four smaller regions in a quad-tree like structure. A similar structure is designed for the temporal dimension, which may have scales from years to hours at different levels. As a new tweet is received from the tweet stream, it

is assigned to a cube in the lowest space-time hierarchy according to its location and posting time. The authors proposed a spatio-temporal hashtag clustering technique that incrementally clusters hashtags in the same cube according to similarity of words that co-occur in tweets. In order to detect remarkable events among the hashtag clusters in a cube, the authors devised a ranking method that measures the popularity, burstiness, and localness of clusters, and calculated event scores to obtain top-k events for the region and time of the cube. Events are maintained at the lowest level of the space-time hierarchy, but the system also contains an aggregation method to retrieve events at higher levels, i.e., those that occur in larger regions and over longer time periods. STREAMCUBE is presented as an online system since it updates the underlying data cube in real time and clusters hashtags incrementally. However, it cannot generate online alerts since it expects user-initiated queries to list the top-k highest ranked hashtag clusters.

Local events in a given tweet set are detected in [19] by finding the clusters of geographically collocated tweets containing the same subset of words. Given a list of recent tweets retrieved from the tweet stream, the authors analyzed tweet texts and determined if tweets containing the same words were spatially clustered. To that aim, they applied a density-based spatial clustering algorithm, *DBSCAN*, to cluster tweets according to their geographical distances and detect clusters of areas with high tweet densities [41]. The resulting tweet clusters, which contain a specific set of words and exhibit a spatially high density in a region, are considered potential event candidates in that region. Furthermore, the authors classified the event candidates based on the historical tweet data and selected the regions and words that had been active for a certain period of time. In this method, the size of the region and the minimum number of tweets to represent an event are determined by the thresholds in the DBSCAN algorithm.

An extension to the DBSCAN, called (ϵ, τ)-*density-based spatiotemporal clustering*, is proposed in [120] by analyzing the temporal proximity of tweets in

addition to their spatial distances. Specifically, two tweets are defined to be in the (ϵ, τ)-neighborhood if their geographical distance (obtained from the GPS coordinates) and the difference of their posting times are within the specified thresholds. Given a set of tweets containing a topic-specific keyword, the system generates tweet clusters to identify bursty areas of tweets using (ϵ, τ)-density-based spatiotemporal clustering. Clusters represent active regions for a specific time interval, and the location corresponding to a cluster is defined by the coordinates of all tweets in the cluster. The authors experimented with the proposed algorithm to locate "snow" and "rain" events in Japan.

Despite the proposed extensions to perform incrementally in an online fashion, spatial clustering methods are mostly executed on a given tweet corpus retrospectively. They usually require the set of tweets to be modeled in a data cube or graph in order to analyze spatial proximity and identify the clusters. Building this model and detecting clusters as soon as a new tweet is received from the online tweet stream might enforce performance limitations, considering the volume and velocity of tweets. For example, the analysis of a large tweet set over short periods and on a fine-grained spatial scale may require considerable amount of memory. Moreover, depending on the selected time and space granularity, multiple locations can be estimated for the same event, some of which may be false alarms.

6.4 Evaluation Metrics

A common practice to evaluate the performance of event localization is to apply the proposed solution to a specific data set, and compare the results with the locations of the selected real-world events. We note that in some of these studies reviewed in this chapter, analyses are primarily performed from an event detection perspective, in terms of whether the targeted events are detected in the expected locations. For this purpose, subjective user studies are widely undertaken, in which human annotators decide whether the identified events map

to the real-world events in the ground truth [38, 121, 132]. On the other hand, several of the reviewed studies conduct an evaluation specific to the location estimation method, trying to measure the accuracy of the estimations. In this section, we describe the evaluation metrics and discuss their applicability in different scenarios.

Methods that numerically evaluate the accuracy of a location estimation method can vary depending on the representation of locations and the dataset in experiments. We identified the following three metrics that are commonly used in the literature; 1) error distance, 2) match rate, and 3) precision-recall [28, 82, 138].

Error distance: If the estimated location and ground truth data are available as geographical coordinates, it is possible to verify the accuracy of estimation by measuring its distance to the expected ground truth location. The calculation of *error distance* for an event e is given in Equation 6.1, where $loc_{act}(e)$ represents the actual coordinates of the event, and $loc_{est}(e)$ is its estimated location. The Euclidean distance is a commonly used distance function for this purpose. Low error distance means that the estimation is close to the actual location of the event. If the experiments are performed on multiple events E, the overall accuracy can be measured using the *average error distance*, as given in Equation 6.2. An evaluation using the Euclidean distance between the estimated and actual locations for earthquakes is performed in [107, 108]. In the experiments, global seismic observations provided precise locations to be used as the ground truth.

$$ErrDist(e) = distance(loc_{act}(e), loc_{est}(e)) \tag{6.1}$$

$$AvgErrDist(E) = \frac{\sum_{e \in E} ErrDist(e)}{|E|} \tag{6.2}$$

Match rate: If the estimation results and ground truth locations are represented by named locations, an alternative evaluation metric can be counting the correct or incorrect estimations and calculating the ratio of the correctly localized events in the test dataset. We call this ratio *match rate*, which is also

referred to as *accuracy* in [28, 49, 138]. We chose the term "match rate" to describe this metric since "accuracy" has a more general meaning. It was used in [48] to evaluate the described event localization method on traffic accidents in California, comparing their estimations with the reports of the California Department of Transportation. We adopted a similar approach in [90] to evaluate the localization of two earthquakes in Turkey. Since the estimated location granularity was at the level of city, we could determine whether the estimated city matched the actual event location. Match rate can also be used in grid-based spatial models. For example, in [1], the region of interest is divided into grid cells to find the stadium that hosted a football match. If the center of the estimated grid cell coincides with the location of the stadium, the result is accepted as a correct estimation.

Precision-Recall: If the data model used for event localization is a grid, accuracy can also be measured by a precision-recall analysis, which is based on the number of true/false positives and negatives, and the calculation of precision, recall, and F_1 measures. In [82], Middleton *et al.* divided the region of interest into an 8x8 grid and manually labeled each grid cell to describe the actual location of an event. In the evaluation, the authors counted true/false positives and negatives, where a true positive occurred if the estimation result matched the expert label for a cell. Based on the number of correct and incorrect grid cells, the precision, recall, and F_1-scores were calculated to validate the results.

6.5 Human-Computer Interaction

The results of the spatial analysis for events detected in Twitter can be presented on a graphical user interface for visual interpretation. The visualization mostly consists of a map with event-related tweets given as pin-pointed objects or with active regions shaded in different colors. For example, TwitterStand[6] presents a list of topics detected on a global scale through a web application [109]. The application includes an interactive map that allows users to view

[6] http://twitterstand.umiacs.umd.edu/News [accessed 01 June 2016]

the details of the topic and related tweets according to the estimated location of the topic. In another example, SensePlace2[7] presented in [79] provides an interface enabling users to perform keyword-based queries. Tweets retrieved for a query are plotted on a map, which can also help improve situation awareness in crisis situations. Similarly, in [74], tweets mentioning crimes and disasters are collected via keyword search, and displayed on a map according to the location in the user profile. Another information system, LITMUS[8], displays feeds about landslides retrieved from social sensors throughout the world [84]. [80] presents a system to visualize and summarize events, displaying event-related tweets on a map colored in accordance with their sentiments. Jasmine[9], a local event detection system, expects event parameters from users, and lists the detected local events on an interactive graphical interface [132]. Once an event in the list is selected, focus moves to the location of that event on a map, displaying further details about the event. Tweets posted about a forest-fire in France are analyzed in [34], marking relevant locations in circles sized in proportion with the number of tweets mentioning a relevant place name. We adopted a similar presentation in [90] to display earthquake locations as circles centered at the city of the estimated location and with radius proportional to the strength of belief for that estimation. In [107], the estimated epicenter of an earthquake is shown as a point, and the trajectory of a typhoon is formed by connecting the estimated points obtained at discrete time intervals. The Emergency Awareness System[10] presented in [99] processes tweets over 5-minute windows to detect bursty keywords related to emergency events, such as fires, earthquakes, and terrorist attacks. This system also sends alert messages to local authorities to warn them about hazards. The authors classified the tweets to identify first-hand reports, and highlighted their geotag coordinates in the alert message.

The variety in the content of graphical user interfaces shows that it is possible to utilize spatial information in event-related tweets for different purposes. In

[7] http://www.geovista.psu.edu/SensePlace2 [accessed 01 June 2016]
[8] https://grait-dm.gatech.edu/demo-multi-source-integration [accessed 01 June 2016]
[9] https://sites.google.com/a/onailab.com/watanabe/jasmine [accessed 01 June 2016]
[10] https://esa.csiro.au [accessed 01 June 2016]

some cases, displaying only the locations of tweets on a map may be sufficient to evaluate the spatial aspects of events. For example, if the objective is to track the spread of an influenza-like illness, seeing the individual location of each tweet reporting about the illness may be more preferable than having a single location for the event. Despite the numerous efforts to develop visual presentations for event locations, only a handful of applications are available for public use on the Internet.

6.6 Discussion

The existing work regarding the applied cases of event-pivot and location-pivot methods reveals that location-pivot methods are preferable in situations where the location of an event is predicted beforehand (e.g., weather events based on forecasts or scheduled events) or where the objective is to track events in a specific location. In such cases, the question is mostly related to the time and extent of the event. Location-based burst detection or location-oriented tweet collection techniques may be considered useful candidates for situations where the location of an event is roughly predictable. However, using these techniques, it may not be possible to detect events that happen in a location outside the monitored region. For example, if the objective is to estimate the epicenter of an earthquake in a country, a location-based burst detection method may not be very useful since it detects bursts with respect to a specific point. Executing this method for all points in terms of latitude-longitude in a country would not be a practical solution. In these situations, other location-pivot methods that track tweets in regions larger than the expected location granularity are more flexible and thus appropriate. An abnormal tweet activity observed in a place usually indicates something happening in that place.

Event-pivot methods can be advantageous in estimating event locations for several reasons. Firstly, a significant portion of tweets consists of spam and irrelevant content. Detecting event-related tweets and filtering noisy data prior

to the spatial analysis considerably reduces the size of the problem. Secondly, event detection results can be directly affected by the performance of location-pivot methods in spatial analyses. For example, a location-pivot method using GPS coordinates can miss an event if the number of tweets in a region is not as high as expected. On the other hand, if the existence of an event is tested beforehand, even a single tweet with spatial information can be used to estimate the event's location. Once event-related tweets are retrieved, further effort can be dedicated to look for spatial evidence focusing on these tweets, probably by selecting secondary spatial features. Spatial attributes can even be selected after an analysis of the event type (e.g., GPS coordinates to locate earthquakes, street names in tweet texts for traffic-related events). Moreover, event-pivot methods can be considered more appropriate for tracking changes in events. If a new tweet received from the online tweet stream is associated with an existing event cluster, the location estimation algorithm can be re-executed for the event in order to update its location and improve estimation accuracy. For location-pivot methods, since the event detection task is performed after location analysis, it should be checked whether an event in location A is a new event or an update for a previously detected event in location B [1].

The location estimation method we propose in this thesis can be classified as an event-pivot method using evidential reasoning techniques, specifically DS theory. We presented an initial version of this solution in [90]. In this thesis, we improve the solution to cover locations at multiple granularities and various types of events detected by our event detection algorithm. We propose a way to define an association of evidence between coarse-grained and fine-grained data based on the mixed class hypothesis in DS theory. Additionally, we study the effect of heterogeneous population distribution on the results of location estimation. We implement and execute the proposed location estimation methods under various settings using three different combination rules in DS theory. We discuss the effect of each setting on the location detection performance. We examine the spatio-temporal characteristics of three location related attributes of the posts and their contribution in the location estimation problem. We use a graphical

presentation of the combined evidence which offers a visual representation of the geography of the event.

CHAPTER 7

EVIDENTIAL LOCATION ESTIMATION FOR EVENTS DETECTED IN TWITTER

In this chapter, the proposed solution using DS theory to estimate the locations for the detected events is explained by describing the representation of locations, defining location mapping functions for spatial features in tweets, assigning mass values for locations, and combining these assignments to estimate the location of an event.

7.1 Spatial Information for Location Estimation

The proposed solution based on DS theory requires the set of tweets about an event and the definition of the locations in the frame of discernment Θ. Tweets about the events are identified by the clustering algorithm ICVE-SO described in Chapter 4. However, our location estimation algorithm is not dependent on the event detection process. In other words, given a collection of tweets about an event, grouped either by a clustering or by a keyword-based search method, the proposed methods can be applied to estimate a location for that group of tweets. Moreover, since our clustering algorithm does not take into account tweet features other than the tweet content and time, the location estimation method does not rely on any special cues from event detection. That is to say, the subsequent task of location estimation is not favorably affected by event detection. In the remainder of this chapter, we use the symbol e for an event, x

for a tweet, and X_e for the set of tweets in e.

The locations in the proposed solution are defined as cities and towns in a country organized in a geographical settlement hierarchy in which the country is divided into cities, and a city is divided into towns. Types of spatial information required for the location estimation are the name, center and boundary coordinates, and their hierarchical relationship (i.e., city-town relationship). VGI sources such as OpenStreetMap, Wikimapia, and GeoNames are considered useful resources to provide this information. Regarding the evaluation scenarios described in Chapter 8 covering the events detected in Turkey, we collect city and town information for Turkey from these open sources. Specifically, OpenStreetMap and Wikimapia are used in a complementary way such that cities are retrieved from the OpenStreetMap since it provides more reliable city coordinates than WikiMapia for Turkey, and the towns for cities are taken from Wikimapia since it contains a more complete list of towns. Spatial information retrieved from these resources are stored in a gazetteer, namely in a PostgreSQL database with PostGIS[1] extention that supports spatial queries. While the proposed location estimation method is executed separately at the level of cities and towns, the city-town hierarchical relationship is used to enhance the accuracy of the estimation by applying an association of evidence between the city and the town levels. In the following sections, the proposed method using DS theory is explained for the estimation at the city level, which is the same method applied to the towns.

7.2 Location Mapping Functions

Location mapping functions are used to map a tweet to zero or more cities according to a spatial feature in the tweet for the location estimation at city level. We define mapping functions f_g, f_c and f_u for each of the spatial feature that is used in the proposed solution, namely GPS coordinates, content, and profile location for the tweet, respectively. These f functions map a given tweet

[1] http://postgis.net [accessed 01 June 2016]

x to a subset of Θ using the corresponding tweet attribute, hence $f : x \rightarrow 2^{\Theta}$. Missing information in an attribute is handled by the mapping that tweet to Θ for that attribute to represent indifference. These mapping functions are described as follows:

Mapping with coordinates: The first mapping function, f_g, compares the geographic coordinates obtained from the GPS metadata of a given tweet x against the boundary coordinates of the locations in the gazetteer. If x is geotagged, $f_g(x)$ finds the location that contains x's GPS position and returns that location in a set. Thus, the set of locations returned by $f_g(x)$ for a geotagged tweet consists of a set with a single element in practice. If the tweet is not geotagged, the function returns Θ, which can be interpreted as being indifferent to a specific location.

Mapping with tweet text: The second mapping function, f_c, is the function that maps a tweet x to the locations mentioned in the content of x. If the tweet x contains the names of multiple locations in Θ, $f_c(x)$ returns all these locations as a set with multiple elements. In case the tweet does not refer to a location in its content, the mapping function returns Θ for that tweet to represent indifference.

Recapping that the tweet content is a free-text field, named entity recognition methods can be applied to detect location names in tweet text. However, using conventional NER tools on tweets introduces certain challenges, as discussed in [76]. Moreover, hashtag phrases that do not appear in dictionaries or formal text documents may contain location references (e.g., "#direnankara", "#occupywallstreet"). As a result, since a dictionary of location names is available as a gazetteer, we geoparse a tweet by searching for the location names in its text [12, 54, 109]. For a given tweet text, it is first tokenized into n-gram tokens and the terms that match a location name in the gazetteer are identified in a case-insensitive manner. Since city and town names in Turkey are single term names (unigrams), tokenizing the text into unigrams is sufficient for our experiments [82]. In case the token is a hashtag phrase, the location names contained in the phrase are also identified.

This process results in a set of location names identified in the text. There are two types of ambiguity to be resolved after this step. The first is the Geo/Non-Geo ambiguity for the names that may also refer to entities other than locations [96]. We resolve this ambiguity using a heuristic method that looks for location names in all tweets about the event [12, 109]. Specifically, given a clustered set of tweets about an event, if a tweet in the cluster contains a term t_i with Geo/Non-Geo ambiguity, the heuristic searches for the name of a location t_j that is geographically related to t_i and mentioned at least once in the tweets of the same cluster. If such a t_j exists, t_i is marked as a Geo reference. Otherwise, it is accepted as Non-Geo. In our implementation, two locations are assumed to be geographically-related if they are neighbors or if one of them is the city/town of the other according to the city-town hierarchy (containment). That means, an ambiguous city name is resolved as Geo if there is a tweet in the cluster containing either the name of a town of the city or the name of a neighboring city. The same rule also applies for resolving ambiguous town names, with the neighborhood definition for towns extended as being in the same city. For example, the term "kartal", the name of a town in Istanbul, also means "eagle" in Turkish and it is the nickname of a major sports club in Turkey. When resolving this ambiguity in a tweet, the heuristic looks for the name "Istanbul" or the name of another town of Istanbul in all tweets of the corresponding cluster to resolve "kartal" as a Geo reference in that tweet.

Once the geographical references are identified, the second type of ambiguity to handle in tweet texts is the Geo/Geo ambiguity that may exist between the locations with the same name. This ambiguity is resolved by another heuristic similar to the one described above with additional calculations. The heuristic is based on a scoring of candidate locations and selecting the one with the highest score as the resolution of the ambiguity [109]. In other words, for each possible resolution r_i for an ambiguous location name r, the heuristic counts the number of tweets in the cluster that contains the name of another location that is geographically-related to r_i. The location r_i that attains the largest count is determined as the resolution of the ambiguity in r. If the ambiguity cannot

be resolved because of a tie, all possible choices for the ambiguous name are returned by $f_c(x)$ in the result set, according to the mixed-class hypothesis in DS theory. The resolution of the Geo/Geo ambiguity can be exemplified for an ambiguous town name "Eregli". In Turkey, there exists two towns with name "Eregli", one of which is a town of Konya, and the other is in Zonguldak. If a tweet x in a cluster contains "Eregli" in its content, the heuristic first calculates scores for these two choices, by counting the mentions of Konya and Zonguldak (as well as their towns) in other tweets of the cluster, and selects the one with the higher score. If this Geo/Geo ambiguity in x cannot be resolved due to a tie in the scores, $f_c(x)$ includes all possible choices in its results set, i.e., {Eregli (*in Konya*), Eregli (*in Zonguldak*)}. The idea is that during the fusion of all tweet features, another feature may provide evidence for one of these locations and finally resolve this ambiguity.

Mapping with profile location: The third location mapping function, f_u, utilizes the location names specified by users as their home locations in their Twitter profiles. Given a tweet x, $f_u(x)$ returns the set of locations stated in the Twitter profile of the user who posted the tweet x. If this information is not publicly available, or if no location name is found in the user profile, $f_u(x)$ returns Θ.

Since the location in the user profile is a free-text field, extraction of useful location names from this textual data requires text processing similar to the analyses required for tweet text. Therefore, while resolving the location references in the user profile, similar to the method applied on tweet content, we tokenize the text into unigrams and search the tokens in the gazetteer. As a result, fake locations and unclear location references (e.g., earth, home, wonderland) that are not referring to a specific location in the gazetteer are filtered out by the keyword-based search. Different from the analysis of tweet text, once a token that matches a location name in the gazetteer is identified in the user profile, we do not employ any special handling of Geo/Non-Geo ambiguity for it. Since the attribute is primarily designated to specify location, it can be assumed that

a location name really refers to a location. Additionally, if a location is found with Geo/Geo ambiguity in the profile, the same heuristic to resolve Geo/Geo ambiguities in tweet content is applied. For the unresolved cases, $f_u(x)$ returns all possible choices for the ambiguous name in the result set.

7.3 Basic Probability Assignments for Locations

The results of the three location mapping functions are utilized to make the basic probability assignments to the subsets of Θ, as defined by DS theory in Section 2.3. The BPAs for GPS, tweet content, and user profile are represented by the mass functions m_g, m_c, and m_u, respectively. Given a set of tweets X_e in a cluster, m functions find the basic probability numbers for each subset C of Θ, which can be represented as $m : (X_e, C) \rightarrow [0, 1]$. Their values for a location (or a set of locations) are expected to be directly proportional to the number of tweets mapped to that location by the corresponding location mapping function f.

Accordingly, m_g is the mass function that finds the basic probability numbers for a set of tweets using f_g, i.e., the evidence found in their GPS coordinates. The definition of m_g for a given set of tweets X_e in cluster e is given in Equation 7.1. The mass functions $m_c(X_e, C)$ and $m_u(X_e, C)$ are similarly defined by replacing $f_g(x)$ in Equation 7.1 with $f_c(x)$ and $f_u(x)$, respectively.

$$m_g(X_e, C) = \frac{|\{x \in X_e : f_g(x) = C\}|}{|X_e|} \tag{7.1}$$

In this equation, C represents a subset of locations in Θ. The proper subsets C of Θ where $m_g(X_e, C) > 0$ are called the focal elements of m_g. As explained in Section 7.2, there may be cases that a mapping function does not find evidence for a specific location in a tweet. In this case, mass assignments are made for Θ to represent ignorance. The corresponding basic probability number $m_g(X_e, \Theta)$ that represents total ignorance is equal to $1 - \sum_{C \subset \Theta} m_g(X_e, C)$.

7.4 Combination of BPAs

Combination rules in DS theory explained in Section 2.3 can be applied to combine the three mass functions described above. Fusing basic probability numbers assigned to the subsets of Θ yields combined mass values for subsets of Θ with respect to a cluster. We claim that combining bodies of evidence using location-related tweet features is a useful operation in the event localization problem since many tweets can provide incomplete location information or none at all. There can also be conflicting evidence in each tweet feature, which is resolved in different ways by the combination rules. In this thesis, the three combination rules described in Section 2.3 are implemented and evaluated for a tweet set.

By applying Dempster's rule of combination to the three mass functions m_g, m_c, and m_u pairwise as given in Equation 2.4, the combined mass function $m_{drc} = (m_g \oplus m_c) \oplus m_u$ is obtained. The combination operation yields a list of $m_{drc}(X_e, C)$ values assigned to the subsets C of Θ for the set of tweets X_e. In this notation, *drc* is used as the abbreviation for the Dempster's Rule of Combination. Applying combination rules in Equation 2.6 and Equation 2.7 yields the mass functions $m_{yr} = (m_g \oplus' m_c) \oplus' m_u$ and $m_{dp} = (m_g \oplus'' m_c) \oplus'' m_u$, respectively. That is to say, m_{yr} represents the combined mass function using Yager's Rule, and m_{dp} represents the combined mass function using the combination rule of Dubois and Prade.

7.5 Location Selection

Once the combined mass function is obtained for an event, there are several metrics that can be used to select the best estimation for its location. An interval comparison method based on belief-plausibility intervals has been applied in [40]. We experimented with using maximum total belief for each location in [90].

In this thesis, we use the commonality function Q given in Equation 2.3. Our

intuition is that, the belief value $Bel(c)$ for a city $c \in \Theta$ can be very low, but the total belief for the sets containing c may be high. This would be the case if c rarely appeared by itself in tweet contents, but frequently together with some other cities. A frequently referred to city should be favored and this can be achieved by the commonality function. Thus, the commonality values for all elements in Θ are calculated, and the location with the highest commonality is marked as the estimated location for the event, as given in Equation 7.2. In case of ties, all locations that maximize the commonality are reported. At this stage, the estimations at city and town level granularities are carried out separately, using different frames of discernment accordingly.

$$\arg\max_c Q(\{c : c \in \Theta\}) \tag{7.2}$$

7.6 Association Of Evidence

Although the presented location estimation method is carried out separately for towns and cities, a town name in the tweet content or user profile can also be used as implicit evidence in the estimation at city level (and vice versa). Therefore, we introduce an improvement on the two text-based location mapping functions f_c and f_u, such that they utilize the hierarchical relationship between cities and towns in order to associate the evidence between coarse-grained and fine-grained data. Specifically, let C′ and T′ be the set of cities and towns derived from the set of towns T and cities C, respectively, in the content of a tweet x. For location estimation at the city level using city-town association, $f_c(x)$ returns C ∪ C′. For town level estimation, $f_c(x)$ returns T if it is not empty, T′ otherwise. The rationale for the conditional association of evidence at the town level is to keep the precise information provided by the town name in the result set, without crowding it out with the list of towns in a city. In either level of estimation, if no evidence for a location is found, $f_c(x)$ returns Θ to represent indifference. The same association of evidence is also implemented for f_u. The resulting mass

functions obtained by applying this association on f_c and f_u are denoted by m'_c and m'_u, respectively. The effect of this extension is discussed by performing the proposed location estimation with and without the association of evidence in the evaluations.

7.7 Normalization of GPS

Spatial analyses that use geotagged tweets as a source of information can be affected by differences in population in different regions of the area of interest. In other words, if the population is heterogeneously distributed among the locations of concern, the evidence obtained from the GPS coordinates in geotagged tweets is expected to be inherently biased in favor of the highly populated locations, since the total number of tweets posted from a metropolis is usually higher than the number of tweets posted from a smaller city, independent of any event. This issue has also been discussed in [107, 108]. In order to eliminate such factors that are not related to an event, we devise a normalization method that redistributes the basic probability numbers assigned by m_g to the focal elements. The population distribution could be derived from the official census results in the country, or alternatively, from a set of random tweets sampled from the tweet corpus. Regarding the factors that may affect Twitter usage in a city or town other than their populations (e.g., Internet infrastructure, cultural and economical differences), we prefer to infer this distribution through a selection of non-event clusters, i.e., clusters that are not generated as a result of a specific real-world event at some city or town. For example, tweets about a TV show or about a national day in the country can be assumed to provide an objective unbiased geographic distribution of Twitter usage. In [90], we selected such clusters manually among the presence clusters (e.g., "good morning" clusters). In this thesis, we devise a heuristic to select a subset of such clusters automatically.

Our intuition is that no location in these clusters should be referred to remarkably more often than its average in all the clusters. Among such clusters, we

select the ones with a cluster size and number of distinct locations above certain thresholds. Although this heuristic may not identify all non-event clusters, since our objective is to have an idea of the population distribution, we expect it to provide us with an unbiased tweet sample. Once samples from non-events clusters are identified, we assemble their tweets in a single set, denoted by X_n, and calculate basic probability numbers $m_g(X_n, C)$ for each subset C of Θ by applying Equation 7.1 to X_n. These basic probability numbers calculated for the collection of non-event clusters are utilized to normalize the mass assignments for event clusters. The normalization process for a given event e with tweets X_e is expected to reassign the probability masses $m_g(X_e, C)$ in a way that is inversely proportional to $m_g(X_n, C)$. Therefore, when normalizing the probability masses assigned by $m_g(X_e, C)$ for a cluster e, the total mass on focal elements is redistributed as given in Equation 7.3.

$$m_g^*(X_e, C) = \frac{\frac{m_g(X_e,C)}{m_g(X_n,C)}}{\sum_{K \subset \Theta} \frac{m_g(X_e,K)}{m_g(X_n,K)}} \times (1 - m_g(X_e, \Theta)) \tag{7.3}$$

The resulting normalized mass function for GPS, which returns the normalized values for m_g, is denoted by m_g^*. Using this normalization, tweets posted from a rarely populated city are expected to be supported stronger than the equal number of tweets posted from a metropolis.

We noticed in our experiments that the normalization process in Equation 7.3 might be vulnerable to noise, especially for a frame of discernment with huge differences in the basic probability numbers for non-event clusters. For example, according to the 2013 census in Turkey, Istanbul is a metropolis with more than 14 million inhabitants. On the other hand, there are much smaller cities with a population of around 100 thousand, such as the northeastern city of Ardahan. In such a case, a single tweet in a cluster posted from Ardahan, might be overemphasized after the normalization process. Even if there are many tweets from Istanbul in the cluster, that single tweet from Ardahan could result in a higher normalized probability mass for that small city. For that reason,

if there are few tweets in X_e posted from a location c, we exclude c from the normalization process, i.e., m_g^* has the same value as m_g for c. The redistribution of the total mass is applied to the locations that are referenced more than three times in a cluster. The normalized mass function m_g^* is similarly used in the combination rules, substituting m_g with m_g^*.

7.8 Graphical Presentation for the Combined Evidence

The results of the location estimation process can be visualized in several ways on a graphical user interface for end users. One approach is to mark each tweet about the event on their geographic locations on a map [31, 94, 137]. This approach gives a relatively raw picture of tweet locations, leaving most of the interpretation to the end user. As the number of tweets about the event increases, their marked place on the map may amalgamate so that it becomes increasingly difficult to distinguish the relevant locations.

Once a specific location is estimated for the event, another approach to the process of visualization is to mark the estimation result on the map. This approach was adopted in [108] and [109]. If the estimation is accurately achieved, the end users are presented with a brief and clear view of an event's exact location.

Both of these visualizations mentioned above can easily be implemented using the results of the proposed location estimation method. However, in addition to the estimated location for a given event, DS theory yields valuable information that can also be useful for end users, namely the belief-plausibility intervals for all locations. Therefore, we implement a graphical presentation that shows not only the estimated location for a given event, but also the locations that may be relevant to the event according to their commonality values. These commonality values are calculated in the course of the location selection as explained in Section 7.5. For the visualization, these values are normalized to the range [0,1] and the locations are displayed on a map with colors in accordance with the normalized

commonalities. A web-based user interface is developed using the Google Maps component of the PrimeFaces[2] library, and the estimated locations are shown on the map, in a way that a darker color indicates stronger evidence. Such a map offers users a more comprehensive view of the geographical aspects of the event. For example, after an earthquake, in addition to the estimated epicenter, it would be useful for rescue teams to view a picture of the affected locations.

Since an event can be related with multiple discrete locations, the map provides a way in which all these locations can be displayed to the user. For example, a major earthquake or extraordinary weather conditions in a city might also affect neighboring cities. In this case, displaying all relevant locations with a darker color than the rest of the map would give the end user an idea about the affected region and the certainty of the estimations. This user interface is exemplified for two sample events in Section 8.9.

[2] http://www.primefaces.org [accessed 01 June 2016]

CHAPTER 8

EVALUATION OF THE PROPOSED LOCATION ESTIMATION SOLUTION

This chapter presents the results of the proposed evidential location estimation method applied to a group of tweets corresponding to various events. We describe the set of tweets and events to localize in Section 8.1 and 8.2, respectively. In Section 8.3, we give the metrics used in our evaluation. We compare the proposed DS method with other combination methods in Section 8.4, and discuss the results of using city-town association in Section 8.5. We describe baseline location estimation methods in the literature and compare them with the results of the proposed DS methods in Section 8.6. We discuss the results of using GPS normalization in Section 8.7, present an analysis for localizing earthquakes in Section 8.8, exemplify the graphical presentation of the combined evidence in Section 8.9, and discuss the limitations of the proposed method in Section 8.10.

8.1 Evaluation Setting

We confine our problem domain to include the tweets, events, and locations in Turkey. Therefore, we aggregated the definitions of 81 cities and 964 towns in Turkey from open data sources on the web, namely the OpenStreetMap and Wikimapia, into our gazetteer stored in a local PostgreSQL database with spatial extension. The tweet dataset for evaluation is composed of 10,163,159 public tweets posted in Turkey, retrieved from Twitter Streaming API from 01 May

2013 to 07 June 2013. Our first analysis of these tweets reveals that more than 63% of them are geotagged, and almost 59% have non-empty text content (not necessarily a location name) in the location attribute of the user profile. They were posted by 382,668 distinct Twitter users. The relatively high ratio of geotagged tweets is due to the geographic filter we use when collecting tweets through the Twitter Stream API.

8.2 Ground Truth Annotation

In order to evaluate the performance of the proposed location estimation method, first we detect events and event-related tweets in our tweet dataset using the incremental clustering algorithm with vector expansion presented in Chapter 4. The clusters with a tweet count of 50 or higher are stored in a database, since they may be mentioning an event that had attracted people's attention. Other clusters, i.e., those having less than 50 tweets at the end of their lifetime, are discarded. Among these clusters, some of them are expected to be related to a real-world event (which we call "event cluster"), and others may be concerning the ordinary daily activities of people and pointless babbles, which are not related to an actual story. We described a burst detection method to distinguish these two types of clusters in Chapter 4. However, since our focus is on the evaluation of location estimation, we chose to select event clusters manually in order not to omit any test case.

The clustering is executed on the collected tweets in the order of retrieval, resulting in 2777 clusters, each with a tweet count of at least 50. Among the detected clusters, we manually search for those that can be matched to a newsworthy real-world event published in newspapers, blogs, or earthquake and weather reports. For this purpose, we utilized cluster features, such as the first tweet time, best tweet content, and top terms in the cluster centroid vector. Finally, we identified 157 event clusters for which we could determine the event location at city level as our ground truth. This set includes events over a wide range of

Table8.1: Categories of Detected Events in Ground Truth

Event Type	Count	Example Event
Sports	91	Handball league final in Ankara
Concert/Show	4	Rihanna concert in Istanbul
Accident/Terrorism	5	Deadly traffic accident in Giresun
Demonstration/Protest	49	Gezi Park protests in Istanbul
Earthquake	4	Earthquakes in Izmir and Mugla
Weather Conditions	4	Storm that disrupts air traffic in Izmir

topics divided into six categories according to their location-related characteristics, as presented in Table 8.1 together with the number of clusters in each category. For example, earthquakes are unexpected events first reported by the people who feel the tremors. Events such as concerts and shows are scheduled events that may be mentioned at disparate locations before, during and after the event. Deadly traffic accidents and acts of terrorism are mostly reported via television or radio; therefore, the tweet content plays an important role for the estimation of their location. It may not always be possible to assign exactly one geographical focus to an event [12]. We identified 6 such events, and marked multiple locations as the ground truth. Furthermore, we could not assign any specific town for 12 events. Therefore, the number of event clusters with a town-level location in our ground truth is 145, slightly lower than the number of all event clusters.

8.3 Evaluation Metrics

The performance of the proposed location estimation is measured using two metrics, namely the *match rate* and *error distance*, explained in Section 6.4. Using the match rate metric, we aim to measure the ratio of the correctly localized events to the total number of events in the ground truth. Since estimations are made in terms of discrete locations (city or town), we can assess the correctness of an estimation essentially by checking the exact match between the estimated location and the actual event location. For events associated with multiple locations in the ground truth, an estimation is marked "correct" if the estimated

location is one of the locations in the ground truth.

The error distance for an event e measures the geographical distance between the estimated location $loc_{est}(e)$ and the actual location $loc_{act}(e)$ for that event using $ErrDist(e)$ in Equation 6.1 in which loc represents the location in terms of latitude-longitude coordinates of the city/town centers as defined in the gazetteer. If the estimated location is correct, the error distance becomes zero. Otherwise, it is the distance between $loc_{est}(e)$ and $loc_{act}(e)$ in terms of kilometers calculated using the open source Java library of Openmap[1] for easier human interpretation. In our evaluation, since we have more than one event in the test dataset, the overall performance on all events E is measured in terms of *average error distance*, i.e., by taking the arithmetic mean of error distances as given in Equation 6.2. These evaluations are conducted for town and city level granularity separately.

8.4 Evaluation of Combination Methods

In this section, we examine the results obtained by applying the combination rules in DS theory to the three evidence sources, and compare them with other combination methods in the literature. The accuracy of the estimations in terms of average error distance is given in Table 8.2, grouped in two sections separately for city and town level estimations. Each row in the table corresponds to an evaluation using a different portion of ground truth data for the test, as explained in detail below. The group of the first three columns of the table shows the results of using each BPA separately, i.e., using a mass function alone to make an estimation rather than applying a combination rule to all of them. The estimation for a BPA is made by selecting the location with the highest mass value. A mass assignment made for a multi-class set is handled by dividing the mass equally between the members in the set, as dictated by the principle of insufficient reason [114]. These results show that GPS and user profile provide more accurate evidence than tweet content at the city level. At town level, results

[1] http://openmap.bbn.com [accessed 01 June 2016]

Table8.2: Average error distances (in kilometers) for different sized data sets

	BPAs			**DS theory rules**			**Baselines**	
	m_g	m_c	m_u	DRC	YR	DP	MAJ	NB
Test[a]	**City Level Estimations**							
20%	148.1	490.8	148.1	110.1	110.1	110.1	148.1	**54.6**
30%	130.9	486.6	129.2	112.0	112.0	97.3	130.9	**56.2**
40%	151.5	477.8	137.4	105.7	118.2	108.7	137.4	**61.9**
50%	170.9	398.7	158.6	137.3	147.2	130.8	159.6	**67.5**
60%	140.9	470.9	130.5	127.7	127.7	120.2	131.4	**64.5**
70%	156.1	429.4	139.8	123.8	130.9	125.5	139.8	**84.3**
80%	144.4	469.2	129.4	121.1	127.4	**117.0**	130.1	118.0
90%	147.5	457.3	134.2	122.2	127.8	**118.5**	134.7	135.5
100%*	151.2	445.8	133.6	128.5	133.6	**125.2**	139.9	N/A
Test[b]	**Town Level Estimations**							
20%	192.4	479.0	304.1	121.2	121.2	120.4	359.6	**72.7**
30%	119.6	492.4	422.4	107.4	107.0	121.7	420.8	**99.3**
40%	220.9	446.2	385.8	125.8	125.8	**125.2**	368.1	125.6
50%	183.4	461.9	366.9	**122.6**	**122.6**	125.9	384.4	150.7
60%	141.2	488.0	373.3	112.8	112.6	**107.6**	405.7	114.5
70%	185.4	416.5	371.9	111.9	111.9	**105.5**	352.4	155.5
80%	168.2	469.2	397.3	117.2	117.1	**113.2**	398.6	132.6
90%	167.6	453.8	384.0	110.7	110.6	**107.1**	379.4	148.5
100%*	166.2	438.2	393.1	114.6	114.4	**109.0**	372.7	N/A

[a] Test set size: Percentage of 157 event clusters at city level

[b] Test set size: Percentage of 145 event clusters at town level

* 100% test set size means "no training"

of m_g are close to its city level estimations. The results of m_c are also similar at city and town levels. However, the accuracy of m_u degrades considerably at town level, which can be caused by town names being specified less frequently than the city names in profiles.

The second group of three columns in Table 8.2 shows the results obtained by applying the proposed estimation method using various combination rules in DS theory. Specifically, we combine the mass functions m_g, m_c, and m_u by each of the combination rules given in Section 7.4, and determine the location of a given event by using the commonality value described in Section 7.5. Results of this setting are denoted as *DRC*, *YR*, *DP*, as abbreviations for Dempster's Rule of Combination, Yager's Rule, and Dubois and Prade, respectively. These

results indicate that the use of a disjunctive consensus in *DP* for conflicting cases yields the most accurate estimations compared to the results of *DRC* and *YR*. *DRC* performing slightly better than *YR* can be interpreted as a sign that the handling of conflict by partial ignorance in Yager's rule does not make any contribution to the estimation results. Thus, we dedicate our attention on *DRC* and *DP* in the remainder of our evaluations.

The rightmost two columns in Table 8.2 contains the results of baseline methods. In this group, for the column labeled *MAJ* represents majority voting. This method treats the three mass functions, namely the m_g, m_c, and m_u, as three separate classifiers, and combines their estimations into a single solution by finding the maximum voted location [68]. That means, if at least two BPAs assign the highest mass value to the same location, that location is selected as the event location. If all three BPAs disagree with each other, the majority voting cannot select a specific location.

The last column in Table 8.2 present the accuracy of a Naïve Bayes (NB) classifier that we implemented and trained on a subset of our annotated ground truth events [83]. Each row in the table corresponds to an evaluation using a portion of ground truth data for test, leaving the rest for the training of *NB*. For example, the first row uses 33 events (20% of 157) for test, leaving 124 events for training. Although the experimented methods other than Naïve Bayes do not require any training, they were all executed on the same test set for fairness in comparison.

The Naïve Bayes classifier we implemented works as follows. Given the tweets of an event X_e, it finds the location that maximizes the posterior probability $P(a|X_e)$ for each location $a \in \Theta$ using Equation 8.1. In this equation, $P(a)$ represents the ratio of events in the training set that are assigned to location a, and $P_g(X_e|a)$, $P_c(X_e|a)$, and $P_u(X_e|a)$ represent the prior probabilities found according to the GPS coordinates, content, and the user profile statistics for the tweets in the training set, respectively. The combination taking the product of these probabilities is based on the assumption of independence between

the evidence sources. In order to find probabilities for each evidence source, we use the prior probabilities assigned by the corresponding location mapping function presented in Section 7.2 for the tweets of events in the training set. The calculation of prior probabilities is exemplified for $P_g(X_e|a)$ in Equation 8.2. The idea here is that if tweets of the event X_e are mapped to m distinct locations according to their GPS coordinates, we can measure how likely it is to observe this distribution for each $a \in \Theta$ based on the past observations in the training set. In this equation, n_{a,c_i} represents the number of tweets in class a in the training set that are mapped to location c_i using GPS coordinates, and n_a is the total number of mappings to all locations for class a using GPS. The number of tweets in X_e that are mapped to location c_i is denoted by $z_{e,i}$. The $z_{e,i}$ value helps to differentiate the effect of frequently observed locations in X_e from that of rarely observed locations. In order to avoid zero probability when a location is never observed in the tweets of a class in a training set, we apply smoothing to the equation by adding the constants 1 and $|\Theta|$ in the numerator and denominator, respectively.

$$P(a|X_e) \cong P(a) \times P_g(X_e|a) \times P_c(X_e|a) \times P_u(X_e|a) \tag{8.1}$$

$$P_g(X_e|a) = \prod_{i=1}^{m} \left(\frac{1 + n_{a,c_i}}{|\Theta| + n_a}\right)^{z_{e,i}} \tag{8.2}$$

A major difficulty with the Bayesian methods is related to obtaining accurate *a priori* distributions for classes. In our case, obtaining an event set that includes training data for each distinct city and town in Turkey also presented a challenge. In other words, considering that there are 81 cities and 964 towns in Turkey, it requires huge effort to detect sample events from each of these locations. Since we have 157 events in our ground truth, using a subset of these events for training did not yield prior statistics for all locations. Thus we devised a heuristic to obtain approximate distribution for the locations that do not exist in the training set. It builds on the similarity in the scales (populations) of the

cities/towns. Specifically, we assume that cities in similar scales should have similar distribution patterns for GPS, content, and user profiles in tweets. For example, the number of tweets posted in Ankara (5 million inhabitants) can be expected to be closer to the number of tweets posted in Izmir (4 million inhabitants) rather than in Isparta (400 thousand inhabitants). Thus, if we do not find Ankara but Izmir in the training set, we use the statistics of Izmir to infer the prior probabilities for Ankara (substituting the references to Izmir as references to Ankara properly). This heuristic applies to Equation 8.2 as follows: in order to harvest approximate prior probabilities for a location $a_i \in \Theta$ that does not exist in the training set, we find the location a_j that exists in the training set having the most similar population distribution as a_i. Then, the location mapping statistics available in the training set for a_j are copied and adapted for a_i. Using this heuristic, we can obtain n_{a,c_i} and n_a values for all locations in Θ, i.e., populate prior probabilities for all cities and towns in Turkey. As a result, we could execute Naïve Bayes classifier using all of the three tweet features.

The results in Table 8.2 reveal that, combining evidence in tweet features using a combination method results in an improvement in location estimation accuracy. DS theory yields better results than the other unsupervised combination method, *MAJ*. Among the combination rules experimented in this thesis, *DP* performs slightly better than *DRC* and *YR* in most of the evaluation settings. On the other hand, the supervised method based on Bayesian theory yields comparable results with our unsupervised methods. The Bayesian and Dempster-Shafer techniques have been compared in numerous studies [18, 55] identifying that incomplete probabilistic knowledge is a major impediment to the effective use of Bayesian methods. As stated in [55], Bayesian theory requires accurate and the explicit formulation of prior distributions, whereas DS theory embeds conditioning into belief functions and does not rely on prior information. This argument is also valid for the event localization problem addressed in this thesis. The results given in Table 8.2 illustrate that the accuracy of *NB* is proportional to the size of the training set. Moreover, the size of the training set had to be

at least 30% (leaving 70% for test) to observe the superiority of *NB* over *DP* at city level. At town level, *NB* becomes more advantageous than *DP* only if a much larger training set is used (at least 70%).

8.5 Evaluation of City-Town Association

In this section, we analyze the effect of applying the association of evidence between city and town presented in Section 7.6 on the location estimation results. The results obtained by using city-town association to the methods described in previous section are presented in Table 8.3. In this table, for the columns labeled m'_c and m'_u represent estimations using tweet content and user profile, respectively. Comparison with the results of m_c and m_u in Table 8.2 suggests that the use of association results in a remarkable improvement, particularly for the content. It can be seen that estimations made by m'_c are closer to the ground truth than estimations of m_c, both at city and town levels. The city-town association does not yield much change for m_u at the city level. However, at town level, it noticeably improves the estimation accuracy. Thus, comparing the results for m_u and m'_u, we can conclude that people mostly give a city name as their home locations in their profiles, rather than a town name.

Combination rules in DS theory are applied on mass functions that use city-town associations, as well. That means, we combine m_g, m'_c and m'_u by the combination rules given in Section 7.4 and estimate a location as described in Section 7.5. The results obtained by using city-town association on *DRC*, *YR* and *DP* are denoted as *ADRC*, *AYR* and *ADP* in Table 8.3, respectively. The results indicate improvement when the association of evidence is applied, particularly for the town-level estimations. Users stating their hometowns at the city level in their profiles, and tweets mentioning the city names of the events make a remarkable contribution in estimating the towns for events. These results show that among the methods based on DS theory the most accurate estimations are made by *ADP*.

Table8.3: Average error distances (in kilometers) using city-town association

	BPAs		**DS theory rules**			**Baselines**	
	m'_c	m'_u	ADRC	AYR	ADP	AMAJ	ANB
Test[a]	**City Level Estimations**						
20%	307.4	148.1	110.1	110.1	86.0	148.1	**19.8**
30%	219.7	129.2	88.6	88.6	73.9	130.9	**40.4**
40%	267.7	137.4	105.7	105.7	96.3	137.4	**41.2**
50%	256.1	158.6	137.3	137.3	120.9	159.6	**42.2**
60%	270.1	140.0	115.8	115.8	100.0	140.9	**49.3**
70%	262.5	148.0	123.8	123.8	118.3	148.0	**70.9**
80%	259.1	136.6	112.2	112.2	101.8	137.2	**92.2**
90%	268.5	140.5	114.3	114.3	**105.0**	141.1	113.1
100%*	255.5	137.1	121.4	121.4	**107.9**	143.4	N/A
Test[b]	**Town Level Estimations**						
20%	328.7	155.0	29.8	29.8	**29.0**	326.1	63.8
30%	423.9	210.1	47.5	64.9	**46.4**	325.4	74.7
40%	334.7	217.0	78.0	78.0	**64.5**	297.6	91.0
50%	347.9	218.5	58.7	58.7	**54.2**	297.8	99.0
60%	362.3	183.4	43.3	52.2	**42.8**	315.6	63.7
70%	296.2	203.9	68.0	68.0	**57.3**	285.7	119.7
80%	357.0	207.7	64.3	71.0	**57.3**	303.9	101.5
90%	345.1	207.8	63.0	69.0	**54.5**	296.4	130.3
100%*	338.4	205.1	66.9	72.3	**59.2**	292.4	N/A

[a] Test set size: Percentage of 157 event clusters at city level

[b] Test set size: Percentage of 145 event clusters at town level

* 100% test set size means "no training"

The column labeled *AMAJ* presents the results of applying the majority voting on m_g, m'_c and m'_u. Town level results in Table 8.3 show that using city-town association yields an improvement for *AMAJ* over *MAJ*. However, at the city level, despite the differences in m_c and m'_c, the two settings of majority voting are very similar in accuracy. In our analysis, we observe that except for a minor disruption due to a distant town name in user profiles for one event, using city-town association did not change the results much at city level for *AMAJ*. In most of the cases, a more accurate estimation of m'_c was not sufficient to change the majority of the votes.

ANB represents the extended version of *NB* using the city-town association for content and profile. That means, for location estimation at the city level, while

NB uses only the city names identified in content and profile, *ANB* additionally uses the cities for the towns occurring in these features. For town level estimation, *ANB* handles a city name in a feature as a reference to the towns of that city, if no town name is found in that feature. This rule is applied for the training of the classifier and for calculating the posterior probabilities using Equation 8.1 and Equation 8.2.

Comparing the results in Table 8.3 with the results in Table 8.2, we can conclude that applying city-town association in tweet content and user profiles mostly improves the estimation accuracy. Among the combination methods, DS theory yields better results than the other unsupervised methods, with *ADP* apparently outperforming *AMAJ* and yielding slightly better estimations than *ADRC* and *AYR*. For city level estimations, the size of the training set had to be at least 20% (leaving 80% for test) to observe the superiority of *ANB* over *ADP*. However, at town level, all combination rules in DS theory, namely *ADRC*, *AYR* and *ADP* become more advantageous than the Bayesian classifiers for any ratio of the training set size in our experiments. We, thus, conclude that if sufficient amount of quality training data is not available, then combination rules in DS methods, particularly *ADP*, are stronger candidates for combining bodies of evidence in tweets for location estimation.

8.6 Comparison with Baseline Methods in the Literature

In this section, we compare the results obtained using the proposed method based on DS theory with the results of other methods in the literature that were developed for location estimation in microblogs.

8.6.1 Baseline Methods

Among the event-pivot location estimation methods in the literature, we implemented the following eight as baselines to use in our evaluations.

Maximum Content Frequency (ConFreq): In [123], authors argue that the mostly referred location names in event-related tweets can be assigned as the location of an event. Therefore, they apply majority voting based on the number of location names in tweets. Accordingly, in this method, which we refer as *ConFreq*, we count the number of tweets mentioning each location in an event cluster, and assign the mostly mentioned location in tweets as the location of that event. This approach is also referred to as "majority voting" or "maximum voting" in the literature. Given a set of tweets about an event, it basically finds the mostly referred location name in tweet contents.

Maximum Profile Frequency (ProFreq): As stated in [48], users tend to provide coarse-grained location information in their user profiles. Hence, the authors apply majority voting on the identified location names for localizing events. Based on this idea, we find the frequencies of locations specified in the profiles of the users who posted a tweet about an event. The location mostly referred to in profiles is assigned as the event location. We call this method as *ProFreq* in our evaluations.

Mean of GPS coordinates (MeanGPS): In this method, we find the mean of the coordinates in terms of latitude-longitude pairs obtained from the GPS coordinates of the geotagged tweets in each cluster. The location mapped to the mean of the geographical coordinates is assigned as the location of the event. This method has also been used as a baseline in [107, 108].

Kalman Filter (KF): Kalman filter is a variant of Bayesian filters widely used for location tracking problems using sensor observations. It is based on updating an estimate at each new measurement by applying an update rule using the previous estimate and the new sensor value [44]. It represents uncertainty in estimation by finding the distribution's mean and covariance. Applying Kalman filters for the estimation of earthquake locations using tweets has been proposed in [107]. In that work, the authors use the GPS coordinates of the geotagged tweets as measurements. For tweets that lack GPS coordinates, they use the location in the user profiles. We implement and evaluate the Kalman filter as a

baseline according to the description given in [107]. We process the coordinates in the form of latitude-longitude pairs in the tweets of an event cluster according to their received order, and update the latitude and longitude estimations iteratively until all the tweets in the cluster are processed.

Particle Filter (PF): The particle filter is a probabilistic approximation algorithm that represents belief by sets of particles [44]. The method we use as a baseline is an implementation of the "normal particle filter" following the description of the SIS algorithm presented in [107, 108] to estimate the locations of earthquakes in Twitter. According to this algorithm, particles distributed around the region of interest are created at an initial generation stage. Then for each new observation, i.e., a new tweet providing location information, it executes iteratively by sampling new particles, updating particle locations, and assigning weights based on their distances to the recently received tweet. At each iteration, the estimated location for the event is updated as the mean of particle coordinates. Following the description given in [108], we use the GPS coordinates and the location in the user profile of event-related tweets. The location that is mapped to the coordinates estimated by the algorithm is assigned as the location of the event. During our evaluation process, since the algorithm includes random sampling, we executed it 10 times on the same dataset and used the average of their accuracies.

Sampled Weighted Particle Filter (SWPF): This enhancement to the "normal particle filter" proposed in [108] weighs the particles by considering the population distribution. It is introduced as a way to handle cases where users are not placed evenly in a region. In our experiments, in order to obtain a population distribution sample for the country, we use the tweets in selected non-event clusters that we already found for a similar purpose as given in Section 7.7.

Maximum Content and Profile Frequency (ConProFreq): This method is a combination of *ConFreq* and *ProFreq*, such that it finds the total number of locations referred in tweet texts as well as in user profiles, and selects the location that maximizes this number as the location of an event. This method

has also been used to initialize estimates for the method presented in [49].

Expectation Maximization (EM): A probabilistic location estimation method that jointly locates events and users is proposed in [49], by extending the method presented in [130]. The estimation method is designed to operate for a set of events, users, and locations. We implemented an estimation method following the steps described in that work. The localization estimation process starts by making initial estimates for the locations of events using the locations names in tweet content and user profiles in the corresponding tweets. These estimates are represented in a matrix Z. In our implementation, if majority of location references in tweet content and user profile indicate a location k for an event e_j, we initialize $Z[j][k]$ to 1 and other rows in $Z[j]$ to 0. If there is a tie between t locations, we set the value $1/t$ for each of the t elements in the j^{th} row of Z corresponding to these t locations. If no evidence for a location is found in tweets, we set all elements in $Z[j]$ to 0 at initialization. Matrices X and Y are defined to represent observations. Specifically, $X_{i,j}$ is set to 1 if the user i posted a tweet about the event j, otherwise it is 0. Similarly, $Y_{i,j}$ is set to 1 if the location i is mentioned in a tweet of event j. We also defined SE_i to represent the events that user i actually posted about. Similarly, LE_i is defined as the events that include a mention to a location i. Once the initialization of these variables, we executed two steps iteratively, until convergence. The first step calculates a and f values as given in Equation 8.3, where N represents the number of events. The second step updates Z matrix using Equation 8.4.

$$a_{i,k} = \frac{\sum_{j \in SE_i} Z(j,k)}{\sum_{j=1...N} Z(j,k)} \quad f_{i,k} = \frac{\sum_{j \in LE_i} Z(j,k)}{\sum_{j=1...N} Z(j,k)} \tag{8.3}$$

$$Z(j,k) = \frac{A(j,k)}{\sum_{k=1...K} A(j,k)} \tag{8.4}$$

Calculation of $A(j,k)$ is given in Equation 8.5, where $a_{i,k}$ and $f_{i,k}$ refer to the affinity between "user-event location" and "location name-event location" pairs, respectively. M is the number of users, and P is the number of referred locations.

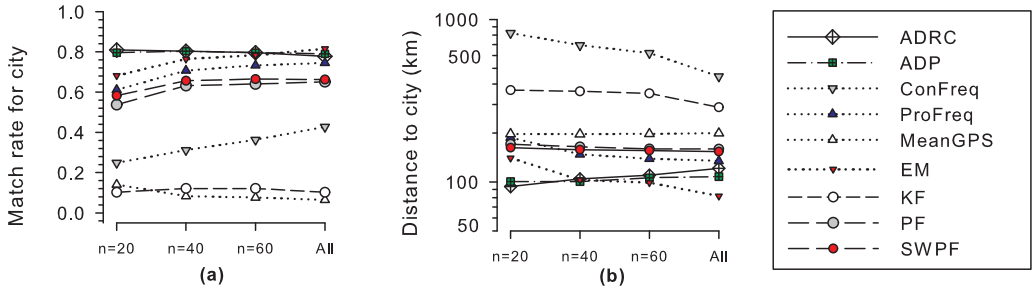

Figure 8.1: Match rates and average error distances at city level

$$A(j,k) = \left(\prod_{i=1}^{M} a_{i,k}^{X_{i,j}} (1 - a_{i,k})^{(1-X_{i,j})} \right) \times \left(\prod_{i=M+1}^{M+P} f_{i,k}^{Y_{i,j}} (1 - f_{i,k})^{(1-Y_{i,j})} \right) \quad (8.5)$$

Finally, values in Z matrix are utilized to estimate the location of the events. Specifically, location k that maximizes $Z[j][k]$ is selected as the location of the event e_j.

8.6.2 Evaluation Results for Baselines

We executed *ADRC*, *ADP*, and the aforementioned eight baseline implementations to estimate the locations of all events in our test dataset. In this set of experiments, we also observed the effect of information diffusion over time, as recognized in [108] and [31], in comparison with the baseline methods. Therefore, the location estimations are generated using the first n tweets in each cluster in the order of retrieval, where n is set to 20, 40, 60, and finally, the number of tweets in the cluster. Detailed evaluation results obtained by DS methods and the baseline methods for different settings are given in Appendix B.

Our findings in terms of match rate and average error distance at city level are displayed in Figure 8.1(a) and Figure 8.1(b), respectively. The results in Figure 8.1 show that the most accurate estimations at city level are generally achieved by DS theory methods, with a slight decrease in performance as n increases. We explain the reason for this using the example of a plane crash on

13 May 2013 in a small city close to the Turkish-Syrian border. For n=20, that means soon after the incident, we observed tweets posted from various cities in Turkey, mentioning the name of the city where the event occurred. Hence, at that time, m_c commits remarkable mass to the correct location, whereas m_g and m_u mostly remain indifferent (assigning the majority of the mass to Θ). As a result, the large amount of mass assigned by m_c to the correct city yields correct estimation after combination. However, as n increases over time, tweets posted from distant large cities, such as Istanbul, dominate the tweet traffic. Therefore, after n=40, large mass values assigned to Istanbul by m_g and m_u result in combined mass values supporting Istanbul, which results in a decline in the estimation performance.

According to the error distance and match rates in Figure 8.1, *EM* yields slightly better results than *ADP* when all event-related tweets are used for location estimation. This can be related to the initialization procedure of *EM*, i.e., using the total number of city names in tweet content and user profile, as suggested by similar results obtained by *ConProFreq* and *EM* given in Table B.1. At city level, combining these two information sources via *ConProFreq* performs better than using tweet content (*ConFreq*) and user profile (*ProFreq*) alone.

These figures illustrate that as n is increased, the tweet content and user profile become more useful and the GPS-based methods start to degrade. The most important reason for this might be the information diffusion. In other words, as more people hear about an event, probably after it is announced through media channels, more people from distant places start posting tweets about that event. Moreover, the name of the event location starts to be mentioned more frequently. We can thus deduce that the first tweets about an event are usually close to the actual location, whereas the tweet content becomes more reliable over time. Differing from GPS-based methods, estimations made by *ProFreq* yield more accurate results as n increases. That means, in terms of information diffusion, the evidence in user profile and GPS coordinates exhibit different patterns. This can be interpreted as people tend to be more sensitive

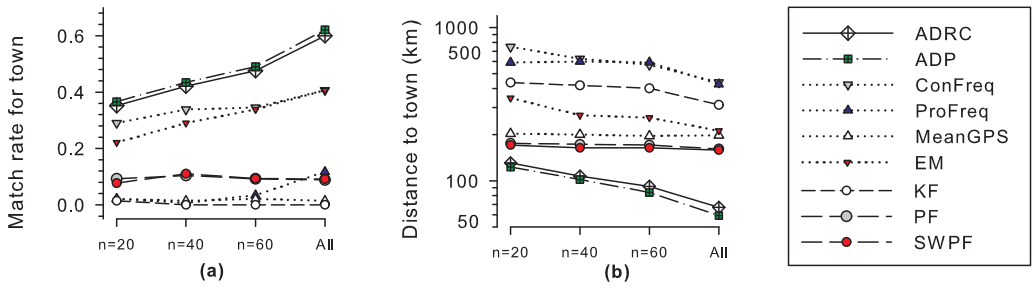

Figure 8.2: Match rates and average error distances at town level

to events in the location stated in their profiles, even if they do not reside in that location at the time of the event. By fusing these three features into a single model through a combination rule in DS theory, we can benefit from all features meanwhile compensating for the variances in their spatio-temporal characteristics.

The results at town level are presented in Figure 8.2(a) and Figure 8.2(b). At town level, the highest accuracy in terms of match rate and average error distance is obtained by DS methods, namely *ADRC* and *ADP*, for all settings of n.

At city level, *ProFreq* achieves remarkably successful estimations. However, its performance is not so good at town level, as shown in Figure 8.2. That means, people mostly specify their home location at city level rather than town level in their profiles. Results of *ConFreq* at city and town levels exhibit similarities in accuracy with increasing performance over time. This can be interpreted as a sign that as more tweets are posted about an event, the number of tweets mentioning the correct event location also increases. Figure 8.2(a) shows that GPS-based methods yield low match rates at town level. Difference in their match rates at city level and town level estimations can be reasonable since it is more difficult to precisely name the correct town, as a smaller area, than to name the correct city using GPS coordinates. On the other hand, the results of GPS-based methods in terms of the average error distance are similar at city level and town level. That means, even if these methods cannot precisely identify the town, their accuracy in terms of average error distance is still comparable to

those of other estimation methods, as given in Figure 8.2(b). Combining these three features using *ADP*, we can obtain the highest accuracy at town level. Noticeably, tweets posted from distant locations over time do not cause any decay in accuracy at town level. We explain the reason for this in relation to the same plane crash event. We observed that if there is a town name mentioned in the tweets of this event, it is usually the correct town name. Meanwhile, although there are tweets with GPS coordinates of distant locations, since the number of towns is much larger than the number of cities (larger $|\Theta|$), evidence from GPS and the user profile cannot focus on a specific incorrect town to mislead the combined evidence. Hence, the accuracy of *ADRC* and *ADP* are improved over time.

The three implementations of Bayesian filters mostly perform better than the *MeanGPS*. Among these filters, the performance of particle filters is obviously better than the Kalman filter, which is consistent with the findings in [107]. It has been argued that Kalman filters are suitable if the uncertainty is not too high and the system has linear state dynamics [44]. In this problem, where tweet dynamics are not necessarily linear and tweets posted from distant locations can have a negative effect on the variance, the use of Kalman filters gives poor performance. Particle filters, being more suitable for nonlinear tracking problems, yield better results; however, they do not surpass *ADRC* and *ADP* for any n value that we used in our experiments.

8.6.3 Evaluations on Event Categories

As discussed in Section 8.2, different types of events can exhibit varying spatio-temporal characteristics. Therefore, we also investigated the estimation performance of our method for each event category. Table 8.4 shows the accuracy of estimations in terms of the average error distance, where estimations are made using all the tweets in event clusters.

The results in Table 8.4 indicate that for most of the cases combining tweet

Table8.4: Average error distances per event type using different estimation methods (in kilometers)

	DS theory				**Baselines**							
	DRC	**ADRC**	**DP**	**ADP**	**Con Freq**	**Pro Freq**	**Mean GPS**	**KF**	**PF**	**SW PF**	**ConPro Freq**	**EM**
Event Type	**City Level Estimations**											
Sports	112.3	108.7	114.4	110.8	655.2	106.4	195.4	314.7	145.9	135.2	**72.8**	**72.8**
Concert/Show	87.6	87.6	87.6	87.6	**0.0**	87.6	96.7	98.0	81.0	87.6	87.6	87.6
Acc./Terrorism	630.3	471.0	630.3	311.8	**22.6**	789.5	622.4	838.1	709.8	787.6	**22.6**	**22.6**
Demo./Protest	131.8	131.8	117.3	**101.1**	209.2	145.6	193.0	242.4	159.4	153.1	117.3	117.3
Earthquake	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	15.1	**0.0**	**0.0**	**0.0**	**0.0**
Weather	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	125.4	20.1	4.6	5.4	**0.0**	**0.0**
All Events*	128.5	121.4	125.2	107.9	445.8	134.5	199.0	288.2	159.1	153.6	**81.8**	**81.8**
Event Type	**Town Level Estimations**											
Sports	141.9	65.2	143.0	**61.6**	531.5	425.6	200.4	329.5	152.9	146.5	263.9	226.5
Concert/Show	**0.0**	**0.0**	**0.0**	**0.0**	165.7	193.2	86.2	133.5	100.0	92.8	247.1	247.1
Acc./Terrorism	12.3	164.0	**0.0**	12.3	**0.0**	683.5	629.9	867.0	770.4	836.9	**0.0**	**0.0**
Demo./Protest	89.3	71.9	**68.6**	71.7	349.2	449.7	169.5	257.2	129.0	125.6	250.8	224.6
Earthquake	14.3	14.3	9.9	9.9	19.3	158.6	14.3	37.8	11.6	11.1	**6.5**	**6.5**
Weather	9.9	9.9	9.9	9.9	462.3	318.6	46.3	51.0	**6.5**	7.1	116.2	116.2
All Events*	114.6	66.9	109.0	**59.2**	438.2	426.6	197.4	312.7	161.2	158.3	242.6	211.9

* All Events means average over all events

features in a single model by a combination rule in DS theory yields better results, especially when the association of evidence between city and town is used. For sports events, we observe that GPS-based methods cannot make accurate estimations. The reason might be that the number of people providing on-site GPS data concerning these events is usually much smaller than the number of people following the event on TV or through other media channels. For sports events, *ProFreq* performs slightly better than *ADP* at city level. This can be interpreted as a sign that a sports event in a city is followed by users having the same city name as their home location in their profiles. Using tweet content in addition to the profile in *ConProFreq* and *EM* improves the accuracy for sports events at city level. But at town level, *ADP* locates the sports events most accurately.

Tweets about concert/show can have frequent references to the city name, such as the Justin Bieber concert on 2 May 2013 in Istanbul, which is usually mentioned as "Istanbul concert" in tweets. Such tweets can help make accurate estimations at the city level. However, these events can receive less accurate evidence from GPS and profile since they can also be followed by people from distant cities. Thus, the combined estimations using DS methods can be less accurate than *ConFreq* at the city level. We observe that town names are rarely mentioned in tweets related to concert/show. Similarly, user profiles tend to be indifferent about town, which is probably the reason why using city-town association did not make any difference between *DP* and *ADP* for these events at the city level. Hence, even very few references to a town can disturb the accuracy of *ProFreq* and *ConFreq*. However, when combined with GPS-provided evidence that supports the correct town more strongly than the other towns in the country, DS methods can make accurate estimations.

Considering the events of type accident/terrorism, the results in Table 8.4 suggest that the evidence from GPS and user profiles is not very promising for such events. However, tweet contents provide remarkably more precise evidence. When the three bodies of evidence are combined, accuracy of content can mit-

igate the poor performance of GPS and the user profile, particularly at town level. For accident/terrorism, there are noticeable differences in the accuracy of city level estimations obtained by different settings in DS theory methods. For example, our test data contained a terrorist act known as the "Reyhanli" attack, named after the town where the incident occurred. After the event has been announced on TV channels, those who began to post tweets about it mostly included the name of the town. This also reflects on the high accuracy of *ConFreq* at town level estimations. This valuable town level information was utilized at the city level through the association of evidence in *ADRC* and *ADP*. On the other hand, at town level, information diffusion and applying city-town association can cause a frequent city name in profiles to support a distant town promoted by GPS coordinates. This has been observed for an accident/terrorism event, and resulted in different error distances for *DRC* and *ADRC*.

Table 8.4 shows that with our methods demonstration/protest, weather, and earthquake events are correctly localized at city level, and at town level, particle filters make slightly better estimations for weather. We observe that tweets from populous towns close to the actual event locations caused minor disruption on the results of DS methods for weather events. At town level, *ConProFreq* and *EM* can localize earthquakes a few kilometers closer than *ADP*. In fact, *ConProFreq* yields better results than using tweet content and user profile alone for localizing earthquakes. We study the estimations obtained by *ConFreq*, *ProFreq*, and *ConProFreq* for the four earthquake events in our test cases, and we observe that for these events, tweet content and user profile rarely contain mentions of town names; and thus, even few references can affect the accuracy of the results. For example, in one of the earthquake events, the correct town was mentioned only in two tweets, but this was sufficient to result in an accurate estimation by *ConFreq*. The correct town was not found in any of the user profiles for that event, but since the number of references to other towns in profile was also very low, the combination of tweet content and user profile in *ConProFreq* could still estimate the town accurately, due to the references in content. In another example for earthquake, *ProFreq* makes an accurate estimation whereas *ConFreq* can

not find the correct town of the earthquake. Their combination in *ConProFreq* was again in favor of the correct town. Hence, *ConProFreq* could perform better than *ConFreq* and *ProFreq* for earthquakes in our experiments.

At city level, methods that combine only tweet content and user profile can be more advantageous than *ADP* for specific event types. However, the superiority of DS methods is more noticable at a finer level of location granularity, namely at town level. The average accuracies of *ADRC* and *ADP* over all events are remarkably better than the baseline methods. There are few specific cases where another method might perform slightly better than *ADP*, but no specific baseline method appears to perform consistently better than our methods at town level.

8.7 Evaluation of GPS Normalization

In this section, we discuss the results obtained by normalizing m_g as described in Section 7.7, and using its normalized form m_g^* in the combination rules in DS theory. Applying Dempster's combination rule to combine $(m_g^* \oplus m_c) \oplus m_u$ yields *NormDRC*. If city-town association is also used for tweet content and user profile, the corresponding combination operation $(m_g^* \oplus m_c') \oplus m_u'$ results in *NormADRC*. Applying this setting for *DP* in a similar way, we obtain *NormDP* and *NormADP*. The results in terms of average error distance with and without GPS-normalization procedures using all the tweets in event clusters are presented in Table 8.5 for comparison.

These results show that normalization can improve the accuracy, but the improvement is not consistent. For example, the results of m_g and m_g^* in Table 8.5 suggest that normalization of evidence in GPS can sometimes yield estimations closer to the actual event location; but sometimes normalization can be misleading. We analyze the estimations made by m_g and m_g^* for the test cases in our experiments to find out possible reasons for different estimations. We observe that if the actual event location is a major city, m_g makes more accurate estimations than m_g^*. On the other hand, m_g^* can achieve to locate events in rural

Table8.5: Average error distances per event type using GPS-normalization (in kilometers)

	Without GPS-Normalization					**With GPS-Normalization**				
	m_g	**DRC**	**DP**	**ADRC**	**ADP**	m_g^*	**Norm DRC**	**Norm DP**	**Norm ADRC**	**Norm ADP**
Event Type	**City Level Estimations**									
Sports	135.5	112.3	114.4	108.7	110.8	271.7	180.9	110.0	146.5	**106.1**
Concert/Show	87.6	87.6	87.6	87.6	87.6	96.0	**35.7**	**35.7**	**35.7**	**35.7**
Accident/Terrorism	789.5	630.3	630.3	471.0	311.8	368.9	335.3	335.3	**176.0**	**176.0**
Demonstration/Protest	145.2	131.8	117.3	131.8	101.1	126.3	93.5	94.5	**78.2**	**78.2**
Earthquake	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Weather	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	38.7	**0.0**	**0.0**	**0.0**	**0.0**
All Events*	151.2	128.5	125.2	121.4	107.9	212.1	145.7	104.8	115.8	**92.4**
Event Type	**Town Level Estimations**									
Sports	171.2	141.9	143.0	65.2	**61.6**	235.5	192.0	193.0	67.2	63.2
Concert/Show	0.0	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
Accident/Terrorism	608.0	12.3	**0.0**	164.0	12.3	698.6	12.3	**0.0**	164.0	12.3
Demonstration/Protest	135.0	89.3	**68.6**	71.9	71.7	121.3	90.1	69.4	71.7	71.6
Earthquake	14.3	14.3	9.9	14.3	9.9	**1.1**	7.6	7.6	7.6	7.6
Weather	19.2	9.9	9.9	9.9	9.9	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
All Events*	166.2	114.6	109.0	66.9	**59.2**	205.4	146.0	140.5	67.8	60.1

* All Events means average over all events

areas more precisely than m_g, since our normalization method favors smaller locations rather than the large ones. In order to prevent noise, we simply used a threshold while selecting for which cities to normalize the mass assignments (Section 7.7). Specifically, we did not normalize the mass for a location unless the location was supported by more than three tweets in a cluster. In fact, this helped reducing the vulnerability to noise; but the threshold could also be determined in a more dynamic way. Rather than using a fixed number for all locations, each location could have a different threshold since few tweets can be considered as noise for small cities and towns, whereas a higher threshold could be more useful for populous locations. Moreover, these location-specific thresholds can also be changed in time. For example, one or two tweets posted within the first few minutes after an event can be interpreted as noise; but as more tweets are added to the event cluster over time, the noise threshold to apply normalization can also be raised accordingly.

This normalization process requires prior knowledge about the population distribution for the locations in Θ, which can also be inferred from the non-event clusters as we described in Section 7.7. Apart from this, GPS normalization does not require any training and it does not introduce significant computational complexity. We regard the first results obtained by the presented normalization heuristic as promising. Aforementioned improvements are considered as a future work.

8.8 Analysis of Earthquakes

There are numerous studies on situation awareness targeting the detection of earthquakes and their locations using social networks [31, 107, 108]. Therefore, we present a detailed analysis of the location estimation results for the detected earthquakes in our test dataset and examine the performance of our event-independent location estimation method for these events. A comparison of the location estimation methods applied to the earthquake clusters is shown

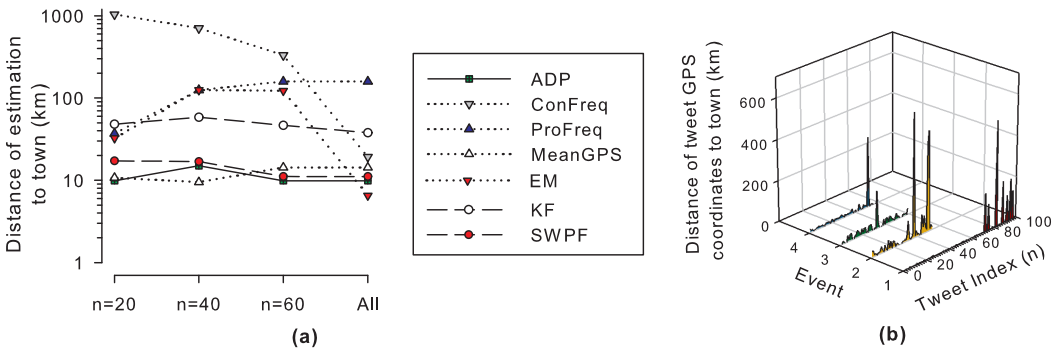

Figure 8.3: (a) Average error distances for earthquakes using different location estimation methods, (b) Distances between the epicenters of four sample earthquakes and the GPS coordinates of the corresponding tweets

in Figure 8.3(a). In order not to clutter the figure, we did not include all evaluation results. The results for all experimented methods are given in Appendix B.

In Figure 8.3(a), it can be seen that although *MeanGPS* attains the most accurate estimations for n=40, its average error distance increases as n increases. We analyze the reason for this by examining the distances of tweets to the epicenter of the four earthquakes referred to in our test dataset according to the GPS coordinates, as shown in Figure 8.3(b). We plot the distance for non-geotagged tweets as -1km in the graph, in order to maintain consistency for different n values. The Figure 8.3(b) shows that although GPS provides useful evidence for earthquakes, it seems to be less reliable over time. On the other hand, the estimation obtained by *ConFreq* gets closer to the epicenter as n increases. A probable reason is that as time passes, after people hear about an event, they might post tweets to report or comment about it and its location, even if they live in places distant from the event.

Figure 8.3(a) illustrates that our event-independent location estimation method yields comparable results to those found by other methods applied for earthquake localization. When all tweets are utilized in estimations, combining tweet content and user profile can result in slightly better estimations than *ADP*. However, for n=20, *ADP* yields the highest accuracy in terms of average error

Figure 8.4: City level estimations for an earthquake in the town Gaziemir,Izmir on 26 May 2013, at 8:31am (3.5 M_L in magnitude).

distance. According to our observation, although *ConFreq*, *ProFreq* and *ConProFreq* cannot make precise estimations for n=20, when combined with GPS coordinates, evidence provided by these features can be useful to obtain better estimations. The increase in error distance for *ADP* at n=40 is mostly due to very large mass assigned by m_g to other populous nearby towns. These results are consistent with the findings of [108]. In that work, authors used n=20 observations for location estimation, and n=40 to alert people about an earthquake.

8.9 Graphical Presentation for the Combined Evidence

The visualization of the combined evidence described in Section 7.8 is given for two example events. The first event is an earthquake in Gaziemir, Izmir. Figure 8.4 shows city level estimations obtained by *DP*. The results of *NormDP*, *ADP* and *NormADP* are the same. In this figure, Izmir is shaded in the darkest color as expected, since it was found to have a much higher commonality score than the other cities.

Figure 8.5(a), Figure 8.5(b) and Figure 8.5(c) show the town level estimations for the same earthquake event by *DP*, *NormDP*, and *NormADP*, respectively. In order to show the actual location of the event, we also mark Izmir and Gaziemir

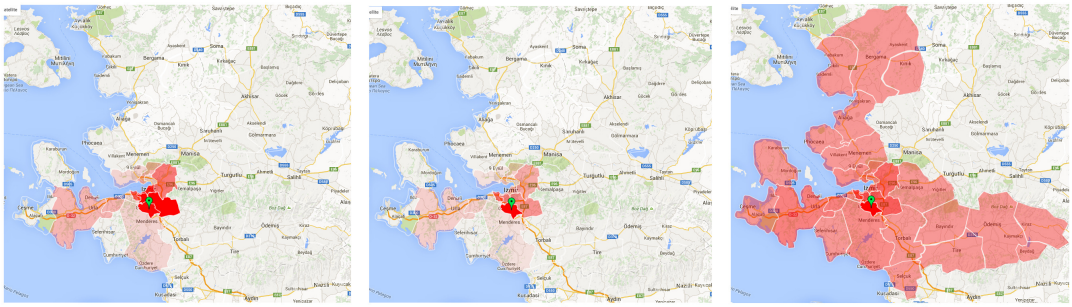

Figure 8.5: Town level estimations for the earthquake in Gaziemir,Izmir. (a): DP, (b): NormDP, (c): NormADP

with green pins as the ground truths. In these figures, Gaziemir and its neighboring towns are shaded, in accordance with the epicenter and extent of the impact of the earthquake.

The difference in shadings between the map in Figure 8.5(a) and Figure 8.5(b) shows the effect of GPS-normalization on the results of combined evidence. In Figure 8.5(a), the correct location of the earthquake, Gaziemir, and several neighboring towns are shaded in a darker color than the rest of the towns in the map. These neighboring towns (namely, Buca, Konak, Karabaglar) were more "populous" regions in terms of the number of tweets measured by our method using non-event clusters (see Section 7.7). Therefore, the normalization of probability assignments using GPS in *NormDP* increased the commonality value of Gaziemir and resulted in the map in Figure 8.5(b), which paints Gaziemir with the darkest color as expected. The map in Figure 8.5(c) has a different shading than the other two. The reason for the coloring of all towns in Izmir is that the city level evidence in the tweets equally affects all of the towns in Izmir when the association is applied.

Our second example is a demonstration/protest type of event. The best tweet found for the corresponding event cluster reports about the fans of two local sports clubs in Izmir traveling towards Istanbul to join the Gezi Park protests. In our ground truth, we assigned both Izmir and Istanbul as the locations of this event at city level. Figure 8.6(a) and Figure 8.6(b) show the city level

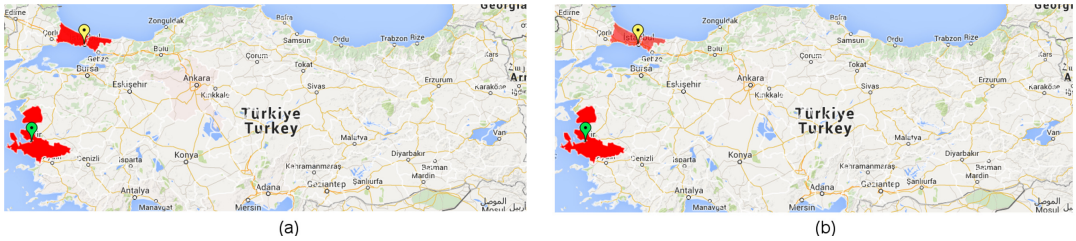

Figure 8.6: People traveling from Izmir to Istanbul to join the Gezi Park protests on 31 May 2013.

estimations made by *DP* and *ADP*, respectively. We marked Izmir (green pin) and Istanbul (yellow pin) on the maps as the ground truths. Both algorithms shade Izmir and Istanbul remarkably darker than the other cities as expected. However, there are differences particularly in the shadings for Istanbul. The map in Figure 8.6(a) shows that *DP* finds very close commonality scores for Izmir and Istanbul, which is acceptable for this event. In Figure 8.6(b), Istanbul is shaded in a lighter color. That means, *ADP* assigns a higher commonality score to Izmir than to Istanbul for the reason that for this event the town names of Izmir were mentioned more frequently in tweets than the towns of Istanbul. Thus, the city-town association in *ADP* resulted in a higher commonality score for Izmir.

These examples demonstrate how shadings based on commonality scores can give the end user an idea about the certainty of the estimations. Whereas the shading in Figure 8.4 points to Izmir as the location of the earthquake with a strong certainty, according to the shading in Figure 8.6, we may not claim that the event happened only and definitely in Izmir. Such shadings can be interpreted either as an event affecting multiple regions, or two distinct events with very similar content happening at different locations.

8.10 Limitations of the Proposed Methods

As reported in [48], by clustering tweets about the same event, spatial indicators from several tweets can be utilized in order to obtain a high confidence in the

location estimation. However, it should be noted that location estimation methods using clustered groups of tweets can suffer from a deficiency in clustering. If tweets about an event are not accurately identified, the following location estimation process may fail to accurately localize that event [123]. The elimination of outliers has been proposed as a solution to this problem in [48]. However, if tweets mentioning two similar events (but in distinct locations) are grouped in the same cluster, tweets of one event could be handled as outliers, which may result in an inconsistency between the description of the detected event and its estimated location. Therefore, when applying an event-pivot location estimation method, it is crucial to minimize false positives and negatives during the selection of tweets. A similar situation can arise for location-pivot methods. If two distinct events occur in close locations, spatial clustering of tweets can yield a single cluster for these two events. For example, if a tweet "goal" were received in the same minute that two goals in two separate games are scored, it would be difficult for the clustering algorithm to determine for which goal the tweet was posted without further information. This sensitivity is a problem for the other baseline methods we examined, as well. However, we can manage this problem by presenting the results to the end user on a map, with locations colored in accordance with their commonalities. In such cases, the map would display multiple locations for the same event, which can be interpreted as two separate events of the same type.

In the case in which the number of subsets of Θ is very large, the execution of combination rules for all subsets of Θ may require considerable amount of memory. In our implementation, we handled this problem by considering only the focal elements, i.e., the subsets of Θ with non-zero probability mass, which constitute a very small number compared to the number of all subsets of Θ. The conjunctive combination operator in Dempster's rule does not cause additional complications related to the size of the focal elements. However, the disjunctive nature of the combination operator in DP (as well as in ADP) may cause the number of focal elements to increase considerably at each combination in the presence of frequent conflict. Therefore, attention should be paid when applying

DP to problems with a large frame of discernment and frequently conflicting bodies of evidence.

One limitation with the applicability of Dempster's rule of combination to two mass functions is that, they should not flatly contradict each other [110]. In other words, their cores should not be disjoint (also see Equation 2.4). All the test cases in our experiments conformed to this assumption, and therefore we were able to apply our methods on all of the clusters in our dataset. In fact this is reasonable since if at least one tweet in a cluster does not provide any evidence for a feature, which is very common, then the combination rule would basically be applicable to the corresponding mass function m. Recalling that Θ represents the space of all possible solutions in the domain, Θ being a focal element for m prevents m from having a disjoint core with the other mass functions. Moreover, even if all of the tweets provided evidence for all three features, it would be very unusual for them to have no common support for any of the locations.

CHAPTER 9

CONCLUSION AND FUTURE RESEARCH

Detecting and localizing real-world events online using the messages posted in microblogs, particularly in Twitter, in order to improve situation awareness has been the primary motivation of this thesis. We studied event detection and location estimation methods in two parts. The contributions of the proposed solutions can be summarized as follows:

- We presented an enhancement for online tweet clustering algorithms by automatically extracting and scoring term similarities in a temporal locality to be used in a vector expansion process, which we call *Incremental Clustering with Vector Expansion (ICVE)*.

- Our experiments indicate an improvement in tweet clustering and event detection accuracy compared to the baseline incremental clustering method.

- We present a comprehensive anaylsis about the solutions that aim to estimate the locations of events detected in microblogs. The analysis includes the types of spatial information that can be used for event localization, their advantages and disadvantages, types of locations that can be generated as a result of location estimation, and specific location estimation techniques in the literature.

- We propose a location estimation method for events detected in microblogs using Dempster-Shafer theory, which allows us to use multiple spatial fea-

tures in tweets in a complementary way and define an association of evidence between coarse-grained and fine-grained data.

- The proposed method is not specific to the event type and it does not require training.

- We experimentally evaluated the proposed location estimation method under different settings on a set of tweets posted in Turkey about events of different types, including concerts, sports, street protests, accidents, and earthquakes. The results show that the proposed method can estimate the location of events with higher accuracy in comparison to the state of the art methods.

In the first part of the thesis, we studied event detection techniques in microblogs, particularly the ones that employ incremental clustering to group tweets around topics in a timely manner for online processing. We observed that the performance of the state of the art clustering techniques can further be improved by finding term-level similarities automatically in a temporal context in tweets and using these similarities in a vector expansion process. Based on this idea, we proposed an enhancement to the standard incremental clustering, which we call Incremental Clustering with Vector Expansion (ICVE), in order to improve the clustering and event detection accuracy in microblogs.

In our evaluation for the proposed enhancement on incremental clustering, we showed that the vector expansion based on term similarities in tweets facilitates collecting relevant tweets in the same cluster more accurately even if they do not share common terms. The evaluations performed for different scenarios suggest that ICVE yields statistically significant improvement in the clustering accuracy, and reduces the false alarm counts for events compared to the results obtained by the baseline incremental clustering algorithm. We demonstrate that the proposed enhancement is efficient, and it does not incur any remarkable cost that would hinder online processing of the stream. As a result, by accurately grouping similar tweets in coherent clusters, we obtain more reliable tweet sets

that are related to a real-world event in order to estimate the location of the corresponding events.

The method we propose to estimate the locations of the detected events uses Dempster-Shafer theory. In this method, we use three location-related features in the tweets as evidence sources and define three separate mass functions to represent the corresponding evidence. Considering this evidence in a complementary way, we combine them all into a single solution using combination rules in Dempster-Shafer theory. In this thesis, we experimented with three combination rules, namely Dempster's rule (DRC), Yager's rule (YR), and the rule of Dubois and Prade (DP). As a result of fusing these sources of evidence in a single model, we can leverage them all and effectively handle the missing and conflicting data. The proposed method is not specific to the type of the event, and it does not require training. Applying DS theory to this problem enables us to represent indifference in case of uncertainties, assign probability values for sets of locations, and make estimations for locations in terms of upper and lower probabilities. Additionally, we introduce a method to apply an association of evidence between coarse-grained and fine-grained data based on the mixed class hypothesis in DS theory. We also discuss the effect of heterogeneous population distribution on the results of location estimation, and experiment with a normalization heuristic to minimize the bias that acts in favor of populous regions. Accordingly, we implemented and executed the proposed location estimation methods under various settings. In addition to the estimated location for an event, we display all locations on a map and use colors and shades in accordance with their commonality values. From the end user perspective, this picture presents all viable candidates for the solution as well as the strongest candidate. Being able to present multiple locations is considered to be useful especially for events that may cover a larger region than a specific town or a city.

In order to evaluate our location estimation algorithm, we compared it with different baselines. We executed these location estimation methods at two levels of location granularity, namely city and town levels. Evaluations on diverse

types of events have shown that combining evidence in tweet features using disjunctive consensus in *DP* mostly yields more accurate estimations than the baselines. Moreover, the proposed enhancement to the combination operation based on association of evidence between city and town level references indicates consistent improvement in accuracy. Hence, the enhanced version of *DP*, which we call *ADP*, appears as a recommendable method to perform event localization.

We regard several issues as further research areas related to event localization in Twitter. Although real-world activities at some place on Earth are expected to trigger new tweets, which can be detected and localized as events, a popular TV show or a meme that has been introduced and gained popularity in a virtual world like Twitter can become trending topics. According to this viewpoint, not all the topics discussed in Twitter can be associated with a physical location. Alternatively, in the context of Twitter, virtual locations such as TV channels or even Twitter itself can be perceived as locations. We believe that, studies in this direction, particularly those that aim to detect events in an open domain, need to take this difference into account and distinguish the concrete physical locations from the virtual ones. For this purpose, a preprocessing stage should be included to analyze whether a detected event is eligible for physical location estimation.

Evaluation is another issue that may be worth further researching. One problem with evaluation is the possible subjectivity in the interpretation of events and their locations. For events whose occurrence and location are not questionable, such as earthquakes or sports events, building a ground truth is relatively straightforward. Seismic reports announced by state authorities can be used as precise point-level ground truths for earthquakes. Similarly, the location of a sports event, e.g., a stadium, can be compared with the estimation result to evaluate the accuracy of an algorithm. However, not all events have such well-defined locations. For example, traffic congestions or weather conditions usually affect fuzzy regions, rather than a single point or strictly bounded areas. Such types of events can also initiate further questions, such as how to describe a traf-

fic congestion, or whether to count a drizzle at a specific location as a rain event. Furthermore, the evaluation of location of these events can be more complicated than for earthquakes. For example, even if the ground truth and estimation results are described in a precise way, judging an estimation as right or wrong can be dubious. If the locations are defined as regions, should we expect an exact match or a high degree of intersection between the ground truth and the estimated regions? Therefore, it can be argued that for some event types, evaluation can be more accountable if the ground truth and estimated locations are described as named locations to facilitate decisions about matches. This issue can be studied in detail to analyze event types and their appropriate location types.

Future work for the study in this thesis will focus on enhancing the location granularity to include more detailed location types, such as streets and topic specific points of interests (e.g., stadiums, concert halls). The normalization that we apply to handle the effect of heterogeneous population distribution has yielded promising results; however, we believe that these results can further be improved by extending it with additional logic using time and the scale of locations. Accordingly, developing more accudate estimators will be among our future research directions.

We plan to investigate the adaptive selection of the parameter values in our clustering algorithm, ICVE. For even better performance, admitting a language-specific perspective and applying natural language processing techniques for similarity analysis seem to have potential. In addition, exploring the relationship between the detected events and building a hierarchical view to present them at various abstraction levels would be quite beneficial to end users.

REFERENCES

[1] H. Abdelhaq, C. Sengstock, and M. Gertz. EvenTweet: Online localized event detection from Twitter. *Proc. VLDB Endow.*, 6(12):1326–1329, Aug. 2013.

[2] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu. Online social networks flu trend tracker: A novel sensory approach to predict flu trends. In J. Gabriel, J. Schier, et al., editors, *Biomedical Engineering Systems and Technologies*, volume 357 of *Communications in Computer and Information Science*, pages 353–368. Springer Berlin Heidelberg, 2013.

[3] C. Aggarwal and C. Zhai. A survey of text clustering algorithms. In C. C. Aggarwal and C. Zhai, editors, *Mining Text Data*, pages 77–128. Springer US, 2012.

[4] C. C. Aggarwal. A survey of stream clustering algorithms. In *Data Clustering: Algorithms and Applications*, pages 231–258. 2013.

[5] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*, VLDB '03, pages 81–92. VLDB Endowment, 2003.

[6] C. C. Aggarwal and K. Subbian. Event detection in social streams. In *SDM*, pages 624–635. SIAM / Omnipress, 2012.

[7] C. C. Aggarwal and P. S. Yu. A framework for clustering massive text and categorical data streams. In J. Ghosh, D. Lambert, D. B. Skillicorn, and J. Srivastava, editors, *SDM*, pages 479–483. SIAM, 2006.

[8] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 19–27, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[9] O. Ajao, J. Hong, and W. Liu. A survey of location inference techniques on Twitter. *J. Inf. Sci.*, 41(6):855–864, Dec. 2015.

[10] A. Al-Ani and M. Deriche. A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence. *Journal of Artificial Intelligence Research*, 17(1):333–361, July 2002.

[11] J. Allan, editor. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, 2002.

[12] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: Geotagging web content. In *Proceedings of ACM SIGIR 2004*, pages 273–280, New York, NY, USA, 2004. ACM.

[13] P. Anantharam, P. Barnaghi, K. Thirunarayan, and A. Sheth. Extracting city traffic events from social streams. *ACM Trans. Intell. Syst. Technol.*, 6(4):43:1–43:27, July 2015.

[14] J. Ao, P. Zhang, and Y. Cao. Estimating the locations of emergency events from Twitter streams. In *Proceedings of ITQM 2014*, pages 731–739, 2014.

[15] F. Atefeh and W. Khreich. A survey of techniques for event detection in Twitter. *Computational Intelligence*, 31(1):132–164, Feb. 2015.

[16] N. Bansal and N. Koudas. Blogscope: A system for online analysis of high volume text streams. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07, pages 1410–1413. VLDB Endowment, 2007.

[17] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Rev.*, 37(4):573–595, Dec. 1995.

[18] S. S. Blackman and R. Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House radar library. Artech House, Boston, London, 1999.

[19] A. Boettcher and D. Lee. EventRadar: A real-time local event detection scheme using Twitter stream. In *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications*, GREENCOM '12, pages 358–367. IEEE Computer Society, 2012.

[20] F. Can. Incremental clustering for dynamic information processing. *ACM Trans. Inf. Syst.*, 11(2):143–164, Apr. 1993.

[21] F. Can, S. Kocberber, O. Baglioglu, S. Kardas, H. C. Ocalan, and E. Uyar. New event detection and topic tracking in Turkish. *J. Am. Soc. Inf. Sci. Technol.*, 61(4):802–819, Apr. 2010.

[22] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st Annual*

International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pages 243–250, New York, NY, USA, 2008. ACM.

[23] Ç. Çöltekin. A freely available morphological analyzer for Turkish. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, and D. Tapias, editors, *LREC*. European Language Resources Association, 2010.

[24] S. Chakraborti, N. Wiratunga, R. Lothian, and S. Watt. Acquiring word similarities with higher order association mining. In *Proceedings of the 7th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, ICCBR '07, pages 61–76, Berlin, Heidelberg, 2007. Springer-Verlag.

[25] H.-w. Chang, D. Lee, M. Eltaher, and J. Lee. @Phillies tweeting from Philly? Predicting Twitter user locations with spatial word usage. In *Proceedings of ASONAM 2012*, pages 111–118. IEEE Computer Society, 2012.

[26] L. Chen and A. Roy. Event detection from Flickr data through wavelet-based spatial analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 523–532. ACM, 2009.

[27] T. Cheng and T. Wicks. Event detection using Twitter: A spatio-temporal approach. *PLoS ONE*, 9(6):1–10, 06 2014.

[28] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based approach to geo-locating Twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 759–768. ACM, 2010.

[29] K. Coombs, D. Freel, D. Lampert, and S. J. Brahm. Using Dempster-Shafer methods for object classification in the theater ballistic missile environment. In *Proceedings of SPIE, Sensor Fusion: Architectures, Algorithms, and Applications III*, volume 3719, pages 103–113, 1999.

[30] D. Corney, C. Martin, and A. Göker. Spot the ball: Detecting sports events on Twitter. In M. de Rijke, T. Kenter, A. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, editors, *Advances in Information Retrieval*, volume 8416 of *Lecture Notes in Computer Science*, pages 449–454. Springer International Publishing, 2014.

[31] A. Crooks, A. Croitoru, A. Stefanidis, and J. Radzikowski. #Earthquake: Twitter as a distributed sensor system. *Transactions in GIS*, 17(1):124–147, 2013.

[32] S. P. Curley. The application of Dempster-Shafer theory demonstrated with justification provided by legal evidence. *Judgment and Decision Making*, 2:257–276, 2007.

[33] C. De Boom, S. Van Canneyt, and B. Dhoedt. Semantics-driven event clustering in Twitter feeds. In M. Rowe, M. Stankovic, and A.-S. Dadzie, editors, *Proceedings of the 5th Workshop on Making Sense of Microposts*, volume 1395, pages 2–9. CEUR, 2015.

[34] B. De Longueville, R. S. Smith, and G. Luraschi. "OMG, from here, I can see the flames!": A use case of mining location based social networks to acquire spatio-temporal data on forest fires. In *Proceedings of International Workshop on Location Based Social Networks*, LBSN '09, pages 73–80. ACM, 2009.

[35] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[36] A. P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339, 04 1967.

[37] A. P. Dempster. A generalization of Bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(2):205–247, 1968.

[38] A. D. P. Dos Santos, L. K. Wives, and L. O. Alvares. Location-based events detection on micro-blogs. *CoRR*, abs/1210.4008, 2012.

[39] D. Dubois and H. Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4(3):244–264, 1988.

[40] L. Dymova, P. Sevastianov, and P. Bartosiewicz. A new approach to the rule-base evidential reasoning: Stock trading expert system application. *Expert Systems with Applications*, 37(8):5564–5576, Aug. 2010.

[41] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[42] W. Feng, C. Zhang, W. Zhang, J. Han, J. Wang, C. Aggarwal, and J. Huang. STREAMCUBE: Hierarchical spatio-temporal hashtag clustering for event exploration over the Twitter stream. In *International Conference on Data Engineering (ICDE)*, pages 1561–1572, April 2015.

[43] J. G. Fiscus and G. R. Doddington. Topic detection and tracking evaluation overview. In J. Allan, editor, *Topic Detection and Tracking Event-based Information Organization*. Kluwer Academic Publishers, 2002.

[44] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *Pervasive Computing, IEEE*, 2(3):24–33, July 2003.

[45] C. Fu and S.-L. Yang. The group consensus based evidential reasoning approach for multiple attributive group decision analysis. *European Journal of Operational Research*, 206(3):601–608, 2010.

[46] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pages 181–192. VLDB Endowment, 2005.

[47] J. Gelernter and N. Mushegian. Geo-parsing messages from microtext. *Transactions in GIS*, 15(6):753–773, 2011.

[48] P. Giridhar, T. Abdelzaher, J. George, and L. Kaplan. On quality of event localization from social network feeds. In *Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 75–80, March 2015.

[49] P. Giridhar, S. Wang, T. Abdelzaher, J. George, L. Kaplan, and R. Ganti. Joint localization of events and sources in social networks. In *Distributed Computing in Sensor Systems (DCOSS)*, pages 179–188, June 2015.

[50] M. F. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221, 2007.

[51] D. He, A. Göker, and D. J. Harper. Combining evidence for automatic web session identification. *Information Processing and Management*, 38(5):727–742, Sept. 2002.

[52] B. Hecht, L. Hong, B. Suh, and E. H. Chi. Tweets from Justin Bieber's heart: The dynamics of the location field in user profiles. In *Proceedings of CHI Conference on Human Factors in Computing Systems*, pages 237–246. ACM, 2011.

[53] B. R. Heravi, D. Morrison, P. Khare, and S. Marchand-Maillet. Where is the news breaking? Towards a location-based event detection framework for journalists. In *Proceedings of the 20th International Conference on MultiMedia Modeling - Volume 8326*, pages 192–204. Springer-Verlag, 2014.

[54] L. L. Hill. *Georeferencing: The Geographic Associations of Information*. The MIT Press, September 2006.

[55] J. C. Hoffman and R. R. Murphy. Comparison of Bayesian and Dempster-Shafer theory for sensing: A practitioner's approach. In *Proceedings of SPIE, Neural and Stochastic Methods in Image and Signal Processing II*, volume 2032, pages 266–279, 1993.

[56] D. Hou, H. He, P. Huang, G. Zhang, and H. Loaiciga. Detection of water-quality contamination events based on multi-sensor fusion using an extented Dempster-Shafer method. *Measurement Science and Technology*, 24(5):1–24, 2013.

[57] L. Hou, J. Li, Z. Wang, J. Tang, P. Zhang, R. Yang, and Q. Zheng. NewsMiner: Multifaceted news analysis for event search. *Knowledge-Based Systems*, 76:17–29, 2015.

[58] T. Hua, F. Chen, L. Zhao, C.-T. Lu, and N. Ramakrishnan. STED: Semi-supervised targeted-interest event detection in Twitter. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 1466–1469. ACM, 2013.

[59] S. Huang, X. Wu, and A. Bolivar. The effect of title term suggestion on e-commerce sites. In *Proceedings of the 10th ACM Workshop on Web Information and Data Management*, WIDM '08, pages 31–38, New York, NY, USA, 2008. ACM.

[60] M. Imran, C. Castillo, F. Diaz, and S. Vieweg. Processing social media messages in mass emergency: A survey. *ACM Computing Surveys*, 47(4):67:1–67:38, Jun 2015.

[61] A. Java, X. Song, T. Finin, and B. Tseng. Why we Twitter: Understanding microblogging usage and communities. In *Proceedings of WebKDD/SNA-KDD 2007*, pages 56–65, New York, NY, USA, 2007. ACM.

[62] P. Jin, S. Lin, and Q. Zhang. *Encyclopedia of Social Network Analysis and Mining*, chapter Spatiotemporal Information for the Web, pages 1997–2010. Springer, 2014.

[63] B. Jongman, J. Wagemaker, B. R. Romero, and E. C. de Perez. Early flood detection for rapid humanitarian response: Harnessing near real-time satellite and Twitter signals. *ISPRS International Journal of Geo-Information*, 4(4):2246, 2015.

[64] S. Jun, S.-S. Park, and D.-S. Jang. Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications*, 41(7):3204–3212, June 2014.

[65] A. M. Kaplan and M. Haenlein. Users of the world, unite! The challenges and opportunities of social media. *Business Horizons*, 53(1):59–68, 2010.

[66] D. Kim, S. Rho, and E. Hwang. Detecting trend and bursty keywords using characteristics of Twitter stream data. *International Journal of Smart Home*, 7(1):209–220, Jan 2013.

[67] J. Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 91–101, New York, NY, USA, 2002. ACM.

[68] L. I. Kuncheva, J. C. Bezdek, and R. P. W. Duin. Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, 34:299–314, 2001.

[69] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600. ACM, 2010.

[70] K. Lee, R. K. Ganti, M. Srivatsa, and L. Liu. When Twitter meets Foursquare: Tweet location prediction using Foursquare. In *Proceedings of MOBIQUITOUS 2014*, pages 198–207, Brussels, Belgium, 2014. ICST.

[71] 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*, LBSN '10, pages 1–10. ACM, 2010.

[72] 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*, CIKM '12, pages 155–164, New York, NY, USA, 2012. ACM.

[73] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee. TwiNER: Named entity recognition in targeted Twitter stream. In *Proceedings of SIGIR 2012*, pages 721–730, New York, NY, USA, 2012. ACM.

[74] R. Li, K. H. Lei, R. Khadiwala, and K.-C. Chang. TEDAS: A Twitter-based event detection and analysis system. In *IEEE International Conference on Data Engineering (ICDE)*, pages 1273–1276, April 2012.

[75] D. Lin, S. Zhao, L. Qin, and M. Zhou. Identifying synonyms among distributionally similar words. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, IJCAI'03, pages 1492–1493, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc.

[76] J. Lingad, S. Karimi, and J. Yin. Location extraction from disaster-related microblogs. In *Proceedings of WWW Companion 2013*, pages 1017–1020, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.

[77] L. Liu and R. Yager. Classic works of the Dempster-Shafer theory of belief functions: An introduction. In R. Yager and L. Liu, editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 1–34. Springer Berlin Heidelberg, 2008.

[78] J. D. Lowrance, T. D. Garvey, and T. M. Strat. A framework for evidential-reasoning systems. In R. Yager and L. Liu, editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 419–434. Springer Berlin Heidelberg, 2008.

[79] A. MacEachren, A. Jaiswal, A. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford. SensePlace2: GeoTwitter analytics support for situational awareness. In *IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 181–190, Oct 2011.

[80] 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*, CHI '11, pages 227–236, New York, NY, USA, 2011. ACM.

[81] 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*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010. ACM.

[82] S. Middleton, L. Middleton, and S. Modafferi. Real-time crisis mapping of natural disasters using social media. *Intelligent Systems, IEEE*, 29(2):9–17, Mar 2014.

[83] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1997.

[84] A. Musaev, D. Wang, and C. Pu. LITMUS: A multi-service composition system for landslide detection. *IEEE Transactions on Services Computing*, 8(5):715–726, Sept 2015.

[85] D. B. Neill. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):337–360, 2012.

[86] D. Nguyen and J. Jung. Real-time event detection on social data stream. *Mobile Networks and Applications*, 20(4):475–486, 2015.

[87] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng. A survey on data stream clustering and classification. *Knowledge and Information Systems*, 45(3):535–569, 2015.

[88] M. Okazaki and Y. Matsuo. Semantic Twitter: Analyzing tweets for real-time event notification. In J. Breslin, T. Burg, H.-G. Kim, T. Raftery, and J.-H. Schmidt, editors, *Recent Trends and Developments in Social Software*, volume 6045 of *Lecture Notes in Computer Science*, pages 63–74. Springer Berlin Heidelberg, 2010.

[89] M. Oussalah, F. Bhat, K. Challis, and T. Schnier. A software architecture for Twitter collection, search and geolocation services. *Knowledge-Based Systems*, 37:105–120, 2013.

[90] 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*, GIR '13, pages 9–16, New York, NY, USA, 2013. ACM.

[91] O. Ozdikis, P. Senkul, and H. Oguztuzun. Semantic expansion of hashtags for enhanced event detection in Twitter. In *Proceedings of VLDB 2012 Workshop on Online Social Systems (WOSS)*, 2012.

[92] O. Ozdikis, P. Senkul, and H. Oguztuzun. Semantic expansion of tweet contents for enhanced event detection in Twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 20–24, Aug 2012.

[93] O. Ozdikis, P. Senkul, and H. Oguztuzun. Context based semantic relations in tweets. In F. Can, T. Özyer, and F. Polat, editors, *State of the Art Applications of Social Network Analysis*, Lecture Notes in Social Networks, pages 35–52. Springer International Publishing, 2014.

[94] A. Padmanabhan, S. Wang, G. Cao, M. Hwang, Z. Zhang, Y. Gao, K. Soltani, and Y. Liu. FluMapper: A cyberGIS application for interactive analysis of massive location-based social media. *Concurrency and Computation: Practice and Experience*, 26(13):2253–2265, 2014.

[95] G. Panteras, S. Wise, X. Lu, A. Croitoru, A. Crooks, and A. Stefanidis. Triangulating social multimedia content for event localization using Flickr and Twitter. *Transactions in GIS*, 19(5):694–715, 2015.

[96] S. M. Paradesi. Geotagging tweets using their content. In *Proceedings of FLAIRS*. AAAI Press, 2011.

[97] 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*, HLT '10, pages 181–189, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[98] S. Phuvipadawat and T. Murata. Breaking news detection and tracking in Twitter. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 3, pages 120–123, Aug 2010.

[99] R. Power, B. Robinson, J. Colton, and M. Cameron. Emergency situation awareness: Twitter case studies. In C. Hanachi, F. Bénaben, and F. Charoy, editors, *Information Systems for Crisis Response and Management in Mediterranean Countries*, volume 196 of *Lecture Notes in Business Information Processing*, pages 218–231. Springer International Publishing, 2014.

[100] Y. Qiu and H.-P. Frei. Concept based query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, pages 160–169, New York, NY, USA, 1993. ACM.

[101] R. Rapp. The computation of word associations: Comparing syntagmatic and paradigmatic approaches. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[102] R. Rapp. A freely available automatically generated thesaurus of related words. In *LREC*. European Language Resources Association, 2004.

[103] A. I. J. T. Ribeiro, T. H. Silva, F. de L. P. Duarte-Figueiredo, and A. A. F. Loureiro. Studying traffic conditions by analyzing Foursquare and Instagram data. In *Proceedings of the 11th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, pages 17–24. ACM, 2014.

[104] S. Rill, D. Reinel, J. Scheidt, and R. V. Zicari. PoliTwi: Early detection of emerging political topics on Twitter and the impact on concept-level sentiment analysis. *Knowledge-Based Systems*, 69:24–33, 2014.

[105] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[106] O. Roick and S. Heuser. Location based social networks - definition, current state of the art and research agenda. *Transactions in GIS*, 17(5):763–784, 2013.

[107] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes Twitter users: Real-time event detection by social sensors. In *Proceedings of WWW 2010*, pages 851–860, New York, NY, USA, 2010. ACM.

[108] T. Sakaki, M. Okazaki, and Y. Matsuo. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):919–931, April 2013.

[109] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. TwitterStand: News in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 42–51, New York, NY, USA, 2009. ACM.

[110] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[111] B. P. Sharifi, D. I. Inouye, and J. K. Kalita. Summarization of Twitter microblogs. *The Computer Journal*, 57(3):378–402, 2014.

[112] L. Shou, Z. Wang, K. Chen, and G. Chen. Sumblr: Continuous summarization of evolving tweet streams. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 533–542, New York, NY, USA, 2013. ACM.

[113] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. d. Carvalho, and J. a. Gama. Data stream clustering: A survey. *ACM Computing Surveys*, 46(1):13:1–13:31, Jul 2013.

[114] P. Smets. The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):447–458, May 1990.

[115] W. Song and S. C. Park. A novel document clustering model based on latent semantic analysis. In *Semantics, Knowledge and Grid, Third International Conference on*, pages 539–542, Oct 2007.

[116] R. P. Srivastava, T. J. Mock, and L. Gao. The Dempster-Shafer theory: An introduction and fraud risk assessment illustration. *Australian Accounting Review*, 21(3):282–291, 2011.

[117] A. Stefanidis, A. Crooks, and J. Radzikowski. Harvesting ambient geospatial information from social media feeds. *GeoJournal*, 78(2):319–338, 2013.

[118] E. Steiger, J. P. de Albuquerque, and A. Zipf. An advanced systematic literature review on spatiotemporal analyses of Twitter data. *Transactions in GIS*, 19(6):809–834, 2015.

[119] D. Sui and M. Goodchild. The convergence of GIS and social media: Challenges for GIScience. *International Journal of Geographical Information Science*, 25(11):1737–1748, 2011.

[120] K. Tamura and T. Ichimura. Density-based spatiotemporal clustering algorithm for extracting bursty areas from georeferenced documents. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2079–2084, Oct 2013.

[121] K. Tamura and H. Kitakami. Detecting location-based enumerating bursts in georeferenced micro-posts. In *International Conference on Advanced Applied Informatics*, pages 389–394, Aug 2013.

[122] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling. NewsStand: A new view on news. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 1–10. ACM, 2008.

[123] S. Unankard, X. Li, and M. Sharaf. Emerging event detection in social networks with location sensitivity. *World Wide Web*, 18(5):1393–1417, 2015.

[124] J. van Dijck. *The Culture of Connectivity: A Critical History of Social Media*. Oxford University Press, 2013.

[125] O. Van Laere, S. Schockaert, and B. Dhoedt. Georeferencing Flickr photos using language models at different levels of granularity: An evidence based approach. *Web Semantics*, 16:17–31, Nov. 2012.

[126] O. Van Laere, S. Schockaert, V. Tanasescu, B. Dhoedt, and C. B. Jones. Georeferencing Wikipedia documents using data from social media sources. *ACM Transactions on Information Systems*, 32(3):12:1–12:32, July 2014.

[127] A. Varga, A. E. C. Basave, M. Rowe, F. Ciravegna, and Y. He. Linked knowledge sources for topic classification of microposts: a semantic graph-based approach. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 26:36–57, May 2014.

[128] 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*, CHI '10, pages 1079–1088. ACM, 2010.

[129] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[130] D. Wang, M. T. A. Amin, T. Abdelzaher, D. Roth, C. R. Voss, L. M. Kaplan, S. Tratz, J. Laoudi, and D. Briesch. Provenance-assisted classification in social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):624–637, Aug 2014.

[131] F. Wanner, A. Stoffel, D. Jäckle, B. C. Kwon, A. Weiler, and D. A. Keim. State-of-the-art report of visual analysis for event detection in text data streams. In R. Borgo, R. Maciejewski, and I. Viola, editors, *EuroVis - STARs*. The Eurographics Association, 2014.

[132] K. Watanabe, M. Ochi, M. Okabe, and R. Onai. Jasmine: A real-time local-event detection system based on geolocation information propagated to microblogs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2541–2544. ACM, 2011.

[133] J. Weng and B. Lee. Event detection in Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*, 2011.

[134] W. Xie, F. Zhu, J. Jiang, E.-P. Lim, and K. Wang. TopicSketch: Real-time bursty topic detection from Twitter. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 837–846, Dec 2013.

[135] R. R. Yager. On the Dempster-Shafer framework and new combination rules. *Information Sciences*, 41(2):93–137, Mar. 1987.

[136] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and online event detection. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 28–36, New York, NY, USA, 1998. ACM.

[137] J. Yin, A. Lampert, M. Cameron, B. Robinson, and R. Power. Using social media to enhance emergency situation awareness. *Intelligent Systems, IEEE*, 27(6):52–59, Nov 2012.

[138] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: Discover spatio-temporal topics for Twitter users. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 605–613. ACM, 2013.

[139] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, pages 103–114, New York, NY, USA, 1996. ACM.

[140] W. X. Zhao, B. Shu, J. Jiang, Y. Song, H. Yan, and X. Li. Identifying event-related bursts via social media activities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1466–1477, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

APPENDICES

Appendix A

CLUSTERING EVALUATION RESULTS

This chapter presents the results of different online tweet clustering methods in our experiments. The results are given in terms of Precision, Recall and F_1-score in Table A.1 for each of the 31 events in our test set, described in Section 5.2. A part of these results were given in Table 5.4 as a comparison between the baseline clustering, namely IC, and the proposed enhancement, ICVE-SO. Table A.1 also presents the results obtained by ICVE-SSO and ICVE-SVD.

TableA.1: Precision, Recall and F_1-scores using IC and the three settings of ICVE

Target Event	**Target Tweet Count**	Precision				Recall				F_1-score			
		IC	ICVE SO	ICVE SSO	ICVE SVD	IC	ICVE SO	ICVE SSO	ICVE SVD	IC	ICVE SO	ICVE SSO	ICVE SVD
EQ#1	34	0.900	0.900	0.900	0.903	0.794	0.794	0.794	0.824	0.844	0.844	0.844	0.862
EQ#2	40	0.569	0.576	0.559	0.523	0.825	0.850	0.825	0.850	0.674	0.687	0.667	0.648
EQ#3	21	0.708	0.783	0.708	0.704	0.810	0.857	0.810	0.905	0.756	0.818	0.756	0.792
GOAL#1	42	0.095	0.922	0.867	0.876	0.254	0.836	0.832	0.845	0.138	0.877	0.849	0.860
GOAL#2	499	0.963	0.922	0.867	0.876	0.577	0.836	0.832	0.845	0.722	0.877	0.849	0.860
GOAL#3	29	0.095	1.000	1.000	0.846	0.254	0.379	0.138	0.379	0.138	0.550	0.242	0.524
GOAL#4	134	1.000	0.669	0.624	0.649	0.634	0.828	0.769	0.843	0.776	0.740	0.689	0.734
GOAL#5	40	0.958	0.767	0.767	0.719	0.575	0.575	0.575	0.575	0.719	0.657	0.657	0.639
GOAL#6	31	1.000	1.000	1.000	0.923	0.323	0.323	0.323	0.387	0.488	0.488	0.488	0.546
GOAL#7	28	1.000	1.000	0.895	1.000	0.607	0.607	0.607	0.750	0.756	0.756	0.723	0.857
GOAL#8	20	1.000	1.000	1.000	0.833	0.350	0.400	0.350	0.500	0.519	0.571	0.519	0.625
GOAL#9	36	1.000	0.895	1.000	0.895	0.333	0.472	0.444	0.472	0.500	0.618	0.615	0.618
GOAL#10	20	0.750	0.857	0.750	0.750	0.300	0.300	0.300	0.300	0.429	0.444	0.429	0.429
GOAL#11	51	1.000	1.000	1.000	0.833	0.745	0.686	0.745	0.588	0.854	0.814	0.854	0.690
GOAL#12	297	0.857	0.802	0.740	0.804	0.444	0.872	0.441	0.896	0.585	0.836	0.553	0.847
GOAL#13	248	0.672	0.853	0.836	0.801	0.315	0.867	0.843	0.895	0.429	0.860	0.839	0.846
GOAL#14	19	0.929	0.938	0.929	0.824	0.684	0.790	0.684	0.737	0.788	0.857	0.788	0.778
GOAL#15	14	0.875	0.875	0.875	0.875	0.500	0.500	0.500	0.500	0.636	0.636	0.636	0.636
GOAL#16	17	0.001	0.111	0.001	0.015	0.059	0.059	0.059	0.305	0.003	0.077	0.003	0.029
GOAL#17	10	1.000	1.000	1.000	1.000	0.700	0.900	0.700	0.900	0.824	0.947	0.824	0.947
GOAL#18	14	0.240	0.308	0.599	0.361	0.403	0.286	0.434	0.500	0.301	0.296	0.503	0.419
GOAL#19	12	0.240	0.758	0.599	0.361	0.403	0.431	0.434	0.500	0.301	0.550	0.503	0.419
GOAL#20	46	0.240	0.758	0.599	1.000	0.403	0.431	0.434	0.283	0.301	0.550	0.503	0.441
GOAL#21	33	0.717	0.709	0.599	0.355	0.328	0.595	0.434	0.333	0.450	0.647	0.503	0.344
GOAL#22	98	0.717	0.709	0.599	0.719	0.328	0.595	0.434	0.653	0.450	0.647	0.503	0.685
GOAL#23	42	1.000	0.906	1.000	0.015	0.405	0.691	0.405	0.305	0.576	0.784	0.576	0.029
GOAL#24	2679	0.740	0.688	0.721	0.682	0.459	0.497	0.500	0.463	0.566	0.577	0.590	0.552
GOAL#25	55	0.844	0.778	0.844	0.810	0.614	0.764	0.614	0.727	0.711	0.771	0.711	0.767
GOAL#26	33	0.844	1.000	0.844	0.810	0.614	0.152	0.614	0.727	0.711	0.263	0.711	0.767
GOAL#27	61	1.000	0.966	1.000	1.000	0.426	0.459	0.426	0.426	0.598	0.622	0.598	0.598
GOAL#28	80	0.414	0.624	0.653	0.380	0.363	0.663	0.400	0.613	0.387	0.642	0.496	0.469
Mean for all events		0.722	0.809	0.786	0.714	0.478	0.590	0.539	0.607	0.546	0.655	0.614	0.621
Unpaired t-test result			0.206	0.352	0.923		0.041	0.237	0.014		0.046	0.203	0.193

Appendix B

LOCATION ESTIMATION EVALUATION RESULTS

This chapter presents the results in terms of match rate and average error distance obtained by different location estimation methods experimented on the test dataset. Table B.1 and Table B.2 give the city-level average error distances and match rates obtained by the baseline methods described in Chapter 8, respectively. Similarly, Table B.3 and Table B.4 contain the town-level results of these baselines.

The following four tables illustrate the accuracy of the proposed location estimation methods under different settings. Among these tables, Table B.5 and Table B.6 present the city-level average error distances and match rates, respectively, whereas Table B.7 and Table B.8 give the accuracy of estimations at town-level.

TableB.1: City-level average error distances using baseline estimation methods (in kilometers)

	m_g	m_c	m_u	m_g^*	m_c'	m_u'	**Con Freq**	**Pro Freq**	**Mean GPS**	**KF**	**PF**	**SW PF**	**EM**	**ConPro Freq**
Event Type	**City Level Estimations (n=20)**													
Sports	135.8	1028.8	184.3	134.0	746.3	181.5	1028.8	187.5	198.4	372.8	172.7	153.4	164.5	171.2
Concert/Show	0.0	0.0	207.9	0.0	0.0	207.9	0.0	207.9	46.2	350.6	52.4	74.1	120.3	120.3
Acc./Terrorism	809.5	0.0	778.9	809.5	0.0	778.9	96.4	799.1	607.4	857.3	665.8	739.9	158.4	317.6
Demo./Protest	151.9	611.0	154.1	138.6	293.8	154.1	618.3	154.1	195.3	354.7	153.4	153.9	118.5	118.5
Earthquake	0.0	800.0	0.0	0.0	800.0	0.0	800.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weather	0.0	436.0	0.0	0.0	436.0	0.0	436.0	0.0	28.3	142.2	1.2	0.0	0.0	0.0
All Events	151.9	818.5	185.0	146.7	555.7	183.4	823.8	187.5	197.2	366.6	170.6	162.4	140.4	149.4
Event Type	**City Level Estimations (n=40)**													
Sports	138.3	900.8	125.1	188.8	595.8	112.7	900.8	125.1	200.4	371.7	160.2	147	94.9	107.8
Concert/Show	0.0	0.0	87.6	0.0	0.0	87.6	0.0	87.6	96.2	324.5	86.3	97.3	0.0	0.0
Acc./Terrorism	789.5	22.6	789.5	794.9	0.0	789.5	22.6	789.5	593.0	925.0	702.8	775.4	341.0	341.0
Demo./Protest	151.9	465.2	152.9	132.1	221.2	152.9	492.3	152.9	186.1	335.7	149.7	145.7	117.9	117.9
Earthquake	0.0	327.1	0.0	0.0	0.0	0.0	327.1	0.0	0.0	56.5	0.0	0.0	0.0	0.0
Weather	0.0	353.9	0.0	0.0	353.9	0.0	353.9	0.0	36.6	44.3	0.0	0.0	0.0	0.0
All Events	152.7	685.4	147.6	176.0	423.4	140.4	693.8	147.6	196.5	360.5	164.2	157.8	102.7	110.1
Event Type	**City Level Estimations (n=60)**													
Sports	138.3	854.7	112.1	227.3	560.5	104.6	854.7	113.6	200.6	373.4	150.1	139.6	96.9	96.9
Concert/Show	0.0	0.0	87.6	0.0	0.0	87.6	0.0	87.6	96.2	293.7	87.6	87.6	0.0	0.0
Acc./Terrorism	789.5	22.6	789.5	674.1	22.6	789.5	22.6	789.5	599.1	881.2	723.4	780.1	181.8	181.8
Demo./Protest	152.3	371.1	145.6	143.9	186.7	145.6	371.1	146.2	188.2	305.8	151.7	153.3	117.9	117.9
Earthquake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.1	0.0	0.0	0.0	0.0
Weather	0.0	331.7	0.0	0.0	331.7	0.0	331.7	0.0	36.6	95.0	0.0	0.0	0.0	0.0
All Events	152.8	620.4	137.8	198.1	392.3	133.5	620.4	138.8	197.5	350.3	159.7	155.8	98.8	98.8
Event Type	**City Level Estimations (n=all)**													
Sports	135.5	655.2	104.9	271.7	396.7	114.8	655.2	106.4	195.4	314.7	145.9	135.2	72.8	72.8
Concert/Show	87.6	0.0	87.6	96.0	0.0	87.6	0.0	87.6	96.7	98.0	81.0	87.6	87.6	87.6
Acc./Terrorism	789.5	22.6	789.5	368.9	22.6	789.5	22.6	789.5	622.4	838.1	709.8	787.6	22.6	22.6
Demo./Protest	145.2	209.2	145.6	126.3	79.5	138.5	209.2	145.6	193.0	242.4	159.4	153.1	117.3	117.3
Earthquake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.1	0.0	0.0	0.0	0.0
Weather	0.0	0.0	0.0	38.7	0.0	0.0	0.0	0.0	125.4	20.1	4.6	5.4	0.0	0.0
All Events	151.2	445.8	133.6	212.1	255.5	137.1	445.8	134.5	199.0	288.2	159.1	153.6	81.8	81.8

TableB.2: City-level match rates using baseline estimation methods

	m_g	m_c	m_u	m_g^*	m_c'	m_u'	**Con Freq**	**Pro Freq**	**Mean GPS**	**KF**	**PF**	**SW PF**	**EM**	**ConPro Freq**
Event Type	**City Level Estimations (n=20)**													
Sports	0.692	0.088	0.549	0.670	0.286	0.560	0.088	0.549	0.088	0.044	0.462	0.525	0.571	0.571
Concert/Show	1.000	1.000	0.500	1.000	1.000	0.500	1.000	0.500	0.500	0.000	0.775	0.750	0.750	0.750
Acc./Terrorism	0.000	1.000	0.000	0.000	1.000	0.000	0.800	0.000	0.000	0.000	0.000	0.000	0.800	0.600
Demo./Protest	0.755	0.429	0.735	0.776	0.694	0.735	0.429	0.735	0.143	0.122	0.639	0.663	0.816	0.816
Earthquake	1.000	0.250	1.000	1.000	0.250	1.000	0.250	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Weather	1.000	0.250	1.000	1.000	0.250	1.000	0.250	1.000	0.250	0.500	0.975	1.000	1.000	1.000
All Events	0.713	0.255	0.611	0.707	0.452	0.618	0.248	0.611	0.140	0.102	0.537	0.582	0.682	0.675
Event Type	**City Level Estimations (n=40)**													
Sports	0.703	0.132	0.692	0.495	0.330	0.725	0.132	0.692	0.055	0.066	0.579	0.610	0.714	0.703
Concert/Show	1.000	1.000	0.750	1.000	1.000	0.750	1.000	0.750	0.250	0.000	0.750	0.75	1.000	1.000
Acc./Terrorism	0.000	0.800	0.000	0.000	1.000	0.000	0.800	0.000	0.000	0.000	0.000	0.000	0.400	0.400
Demo./Protest	0.755	0.510	0.755	0.776	0.735	0.755	0.490	0.755	0.061	0.143	0.727	0.745	0.837	0.837
Earthquake	1.000	0.750	1.000	1.000	1.000	1.000	0.750	1.000	1.000	0.750	1.000	1.000	1.000	1.000
Weather	1.000	0.500	1.000	1.000	0.500	1.000	0.500	1.000	0.000	0.750	1.000	1.000	1.000	1.000
All Events	0.720	0.318	0.707	0.605	0.516	0.726	0.312	0.707	0.083	0.121	0.632	0.656	0.764	0.758
Event Type	**City Level Estimations (n=60)**													
Sports	0.703	0.165	0.725	0.407	0.363	0.747	0.165	0.725	0.055	0.066	0.600	0.633	0.736	0.736
Concert/Show	1.000	1.000	0.750	1.000	1.000	0.750	1.000	0.750	0.250	0.000	0.750	0.750	1.000	1.000
Acc./Terrorism	0.000	0.800	0.000	0.000	0.800	0.000	0.800	0.000	0.000	0.000	0.000	0.000	0.600	0.600
Demo./Protest	0.755	0.551	0.776	0.735	0.755	0.776	0.551	0.776	0.041	0.163	0.712	0.727	0.837	0.837
Earthquake	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.750	1.000	1.000	1.000	1.000
Weather	1.000	0.750	1.000	1.000	0.750	1.000	0.750	1.000	0.000	0.500	1.000	1.000	1.000	1.000
All Events	0.720	0.363	0.732	0.541	0.541	0.745	0.363	0.732	0.076	0.121	0.64	0.664	0.783	0.783
Event Type	**City Level Estimations (n=all)**													
Sports	0.714	0.198	0.747	0.341	0.440	0.736	0.198	0.747	0.044	0.044	0.618	0.632	0.791	0.791
Concert/Show	0.750	1.000	0.750	0.500	1.000	0.750	1.000	0.750	0.000	0.250	0.750	0.750	0.750	0.750
Acc./Terrorism	0.000	0.800	0.000	0.200	0.800	0.000	0.800	0.000	0.000	0.000	0.000	0.000	0.800	0.800
Demo./Protest	0.776	0.673	0.776	0.714	0.878	0.796	0.673	0.776	0.041	0.122	0.718	0.729	0.837	0.837
Earthquake	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.750	1.000	1.000	1.000	1.000
Weather	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000	0.000	0.500	0.950	0.925	1.000	1.000
All Events	0.726	0.427	0.745	0.478	0.631	0.745	0.427	0.745	0.064	0.102	0.651	0.662	0.815	0.815

TableB.3: Town-level average error distances using baseline estimation methods (in kilometers)

	m_g	m_c	m_u	m_g^*	m_c'	m_u'	**Con Freq**	**Pro Freq**	**Mean GPS**	**KF**	**PF**	**SW PF**	**EM**	**ConPro Freq**
Event Type	**Town Level Estimations (n=20)**													
Sports	384.6	878.2	540.0	384.6	781.0	328.0	878.2	580.3	210.6	444.3	179.1	167.3	362.8	468.7
Concert/Show	3.1	385.9	438.0	3.1	51.6	254.2	385.9	438.0	58.2	398.9	73.1	86.2	266.1	438.0
Acc./Terrorism	873.0	10.5	865.4	873.0	10.5	803.0	10.5	865.4	624.5	956.7	711.4	781.0	10.5	10.5
Demo./Protest	174.9	532.3	630.8	174.9	370.7	330.3	532.3	657.1	168.4	399.1	130.6	130.5	400.1	494.5
Earthquake	14.3	1034.0	37.0	14.3	751.7	37.0	1034.0	37.0	10.7	48.0	14.8	17.2	32.5	32.5
Weather	19.2	1414.8	349.5	19.2	1414.8	0.0	1414.8	349.5	51.8	455.9	12.9	12.7	0.0	349.5
All Events	320.3	747.2	558.3	320.3	624.7	332.7	747.2	590.8	202.4	437.4	175.4	170.9	346.7	446.3
Event Type	**Town Level Estimations (n=40)**													
Sports	244.8	748.4	600.1	244.7	621.6	223.6	748.4	641.5	203.5	433.0	171.5	155.2	307.1	447.4
Concert/Show	2.8	37.7	438.0	2.8	37.7	255.3	37.7	438.0	86.5	365.8	111.1	105.8	235.5	341.2
Acc./Terrorism	766.7	0.0	688.4	766.7	0.0	579.4	0.0	688.4	654.8	950.1	748.9	832.2	0.0	0.0
Demo./Protest	150.8	450.3	529.9	150.6	337.7	349.2	454.8	561.5	168.5	367.5	130.1	124.1	225.2	332.0
Earthquake	15.0	705.7	125.0	11.6	65.0	39.4	705.7	125.0	9.5	58.4	14.7	16.9	125.0	125.0
Weather	9.9	1414.8	349.5	9.9	1414.8	0.0	1414.8	349.5	45.5	216.3	2.2	1.7	344.3	349.5
All Events	222.2	624.1	564.5	222.0	495.9	264.7	625.4	599.2	199.7	419.1	172.8	163.7	267.2	387.6
Event Type	**Town Level Estimations (n=60)**													
Sports	216.7	689.4	603.6	213.0	588.7	197.6	689.4	611.4	201.5	421.2	165.7	154.0	300.6	415.5
Concert/Show	3.1	119.4	462.5	2.8	37.7	98.9	119.4	462.5	76.4	346.5	105.4	111.2	234.4	365.7
Acc./Terrorism	832.5	0.0	725.6	832.5	0.0	616.6	0.0	760.4	629.4	960.9	776.0	838.6	0.0	0.0
Demo./Protest	148.3	420.4	535.5	139.6	284.7	232.3	420.4	601.2	163.6	331.1	133.9	125.7	214.4	325.1
Earthquake	14.3	331.6	158.6	1.1	49.3	36.5	331.6	158.6	14.3	46.4	6.3	11.1	122.2	158.6
Weather	19.2	1414.8	0.0	8.3	1414.8	0.0	1414.8	0.0	21.7	126.0	3.9	3.6	0.0	0.0
All Events	206.3	570.8	568.7	201.1	460.2	213.1	570.8	592.9	195.9	400.5	170.8	163.7	257.7	364.9
Event Type	**Town Level Estimations (n=all)**													
Sports	171.2	531.5	389.5	235.5	425.4	188.1	531.5	425.6	200.4	329.5	152.9	146.5	226.5	263.9
Concert/Show	0.0	165.7	193.2	0.0	165.7	98.9	165.7	193.2	86.2	133.5	100.0	92.8	247.1	247.1
Acc./Terrorism	608.0	0.0	635.9	698.6	0.0	810.4	0.0	683.5	629.9	867.0	770.4	836.9	0.0	0.0
Demo./Protest	135.0	349.2	416.3	121.3	228.7	200.3	349.2	449.7	169.5	257.2	129.0	125.6	224.6	250.8
Earthquake	14.3	19.3	158.6	1.1	19.3	36.5	19.3	158.6	14.3	37.8	11.6	11.1	6.5	6.5
Weather	19.2	462.3	318.6	0.0	462.3	18.8	462.3	318.6	46.3	51.0	6.5	7.1	116.2	116.2
All Events	166.2	438.2	393.1	205.4	338.4	205.1	438.2	426.6	197.4	312.7	161.2	158.3	211.9	242.6

TableB.4: Town-level match rates using baseline estimation methods

	m_g	m_c	m_u	m_g^*	m_c'	m_u'	**Con Freq**	**Pro Freq**	**Mean GPS**	**KF**	**PF**	**SW PF**	**EM**	**ConPro Freq**
Event Type	**Town Level Estimations (n=20)**													
Sports	0.176	0.220	0.011	0.176	0.220	0.033	0.220	0.011	0.022	0.022	0.066	0.054	0.132	0.132
Concert/Show	0.750	0.500	0.000	0.750	0.500	0.000	0.500	0.000	0.000	0.000	0.050	0.050	0.000	0.000
Acc./Terrorism	0.000	0.800	0.000	0.000	0.800	0.000	0.800	0.000	0.000	0.000	0.000	0.000	0.800	0.800
Demo./Protest	0.300	0.400	0.050	0.300	0.400	0.075	0.400	0.050	0.000	0.000	0.162	0.145	0.375	0.375
Earthquake	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.000	0.125	0.075	0.000	0.000
Weather	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.100	0.000	1.000	0.000
All Events	0.214	0.290	0.021	0.214	0.290	0.048	0.290	0.021	0.021	0.014	0.092	0.077	0.221	0.214
Event Type	**Town Level Estimations (n=40)**													
Sports	0.231	0.275	0.000	0.231	0.275	0.055	0.275	0.000	0.011	0.000	0.088	0.098	0.220	0.220
Concert/Show	0.750	0.500	0.000	0.750	0.500	0.000	0.500	0.000	0.000	0.000	0.075	0.075	0.250	0.250
Acc./Terrorism	0.000	1.000	0.000	0.000	1.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000
Demo./Protest	0.400	0.425	0.025	0.400	0.425	0.050	0.425	0.025	0.025	0.000	0.143	0.143	0.400	0.400
Earthquake	0.000	0.000	0.000	0.250	0.000	0.000	0.000	0.000	0.000	0.000	0.125	0.075	0.000	0.000
Weather	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.600	0.700	0.000	0.000
All Events	0.276	0.338	0.007	0.283	0.338	0.055	0.338	0.007	0.014	0.000	0.104	0.110	0.290	0.290
Event Type	**Town Level Estimations (n=60)**													
Sports	0.275	0.297	0.022	0.275	0.297	0.055	0.297	0.022	0.011	0.000	0.071	0.076	0.264	0.264
Concert/Show	0.750	0.250	0.000	0.750	0.500	0.000	0.250	0.000	0.000	0.000	0.075	0.000	0.250	0.250
Acc./Terrorism	0.000	1.000	0.000	0.000	1.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000
Demo./Protest	0.500	0.425	0.075	0.525	0.425	0.125	0.425	0.025	0.025	0.000	0.108	0.120	0.425	0.425
Earthquake	0.000	0.000	0.250	0.750	0.000	0.250	0.000	0.250	0.250	0.000	0.550	0.425	0.250	0.250
Weather	0.000	0.000	1.000	0.000	0.000	1.000	0.000	1.000	0.000	0.000	0.000	0.100	1.000	1.000
All Events	0.331	0.345	0.048	0.359	0.352	0.083	0.345	0.034	0.021	0.000	0.092	0.093	0.338	0.338
Event Type	**Town Level Estimations (n=all)**													
Sports	0.352	0.374	0.121	0.264	0.374	0.132	0.374	0.121	0.000	0.000	0.077	0.071	0.374	0.374
Concert/Show	1.000	0.500	0.500	1.000	0.500	0.500	0.500	0.500	0.000	0.000	0.050	0.075	0.500	0.500
Acc./Terrorism	0.000	1.000	0.000	0.000	1.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000
Demo./Protest	0.550	0.425	0.125	0.600	0.450	0.150	0.425	0.075	0.025	0.000	0.120	0.140	0.400	0.400
Earthquake	0.000	0.250	0.250	0.750	0.250	0.250	0.250	0.250	0.250	0.000	0.200	0.200	0.500	0.500
Weather	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
All Events	0.400	0.407	0.131	0.386	0.414	0.145	0.407	0.117	0.014	0.000	0.088	0.091	0.407	0.407

TableB.5: City-level average error distances using the proposed DS methods (in kilometers)

	DRC	YR	DP	ADRC	AYR	ADP	Norm DRC	Norm YR	Norm DP	Norm ADRC	Norm AYR	Norm ADP
Event Type	**City Level Estimations (n=20)**											
Sports	102.7	100.6	105.6	93.1	93.1	105.1	110.8	108.7	103.8	101.3	101.3	103.4
Concert/Show	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Acc./Terrorism	476.8	476.8	317.6	317.6	317.6	317.6	476.8	476.8	317.6	317.6	317.6	317.6
Demo./Protest	110.7	110.7	110.7	94.5	94.5	94.5	110.7	110.7	110.7	94.5	94.5	94.5
Earthquake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weather	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
All Events	109.2	108.0	105.9	93.6	93.6	100.5	114.0	112.7	104.9	98.3	98.3	99.5
Event Type	**City Level Estimations (n=40)**											
Sports	105.1	103.0	105.1	99.4	109.3	101.5	158.7	156.6	155.1	142.3	150.1	142.3
Concert/Show	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Acc./Terrorism	630.3	630.3	630.3	471.0	471.0	471.0	635.6	635.6	635.6	476.4	471.0	311.8
Demo./Protest	117.3	117.3	117.3	101.1	101.1	84.8	104.2	104.2	110.7	94.5	94.5	78.2
Earthquake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weather	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
All Events	117.6	116.4	117.6	104.2	109.9	100.3	144.8	143.6	144.7	127.1	131.5	116.8
Event Type	**City Level Estimations (n=60)**											
Sports	112.9	122.9	115.0	109.3	119.3	111.4	170.2	180.2	158.9	146.3	156.3	135.0
Concert/Show	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Acc./Terrorism	630.3	630.3	630.3	471.0	471.0	471.0	499.9	499.9	499.9	340.6	340.6	176.0
Demo./Protest	117.3	117.3	117.3	101.1	115.5	84.8	115.4	115.4	115.4	101.2	115.6	84.9
Earthquake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weather	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
All Events	122.2	127.9	123.4	109.9	120.2	106.1	150.6	156.4	144.1	127.2	137.5	110.4
Event Type	**City Level Estimations (n=all)**											
Sports	112.3	112.3	114.4	108.7	108.7	110.8	180.9	178.0	110.0	146.5	133.2	106.1
Concert/Show	87.6	87.6	87.6	87.6	87.6	87.6	35.7	35.7	35.7	35.7	35.7	35.7
Acc./Terrorism	630.3	789.5	630.3	471.0	471.0	311.8	335.3	335.3	335.3	176.0	176.0	176.0
Demo./Protest	131.8	131.8	117.3	131.8	131.8	101.1	93.5	93.5	94.5	78.2	94.5	78.2
Earthquake	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Weather	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
All Events	128.5	133.6	125.2	121.4	121.4	107.9	145.7	143.9	104.8	115.8	113.2	92.4

TableB.6: City-level match rates using the proposed DS methods

	DRC	YR	DP	ADRC	AYR	ADP	Norm DRC	Norm YR	Norm DP	Norm ADRC	Norm AYR	Norm ADP
Event Type	**City Level Estimations (n=20)**											
Sports	0.736	0.747	0.725	0.769	0.769	0.747	0.703	0.714	0.703	0.736	0.736	0.725
Concert/Show	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Acc./Terrorism	0.400	0.400	0.600	0.600	0.600	0.600	0.400	0.400	0.600	0.600	0.600	0.600
Demo./Protest	0.837	0.837	0.837	0.857	0.857	0.857	0.837	0.837	0.837	0.857	0.857	0.857
Earthquake	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Weather	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
All Events	0.777	0.783	0.777	0.809	0.809	0.796	0.758	0.764	0.764	0.790	0.790	0.783
Event Type	**City Level Estimations (n=40)**											
Sports	0.747	0.758	0.747	0.769	0.758	0.758	0.560	0.571	0.571	0.604	0.604	0.604
Concert/Show	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Acc./Terrorism	0.200	0.200	0.200	0.400	0.400	0.400	0.200	0.200	0.200	0.400	0.400	0.600
Demo./Protest	0.837	0.837	0.837	0.857	0.857	0.878	0.837	0.837	0.837	0.857	0.857	0.878
Earthquake	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Weather	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
All Events	0.777	0.783	0.777	0.803	0.796	0.803	0.669	0.675	0.675	0.707	0.707	0.720
Event Type	**City Level Estimations (n=60)**											
Sports	0.747	0.736	0.736	0.758	0.747	0.747	0.560	0.549	0.593	0.626	0.615	0.659
Concert/Show	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Acc./Terrorism	0.200	0.200	0.200	0.400	0.400	0.400	0.200	0.200	0.200	0.400	0.400	0.600
Demo./Protest	0.837	0.837	0.837	0.857	0.837	0.878	0.816	0.816	0.816	0.837	0.816	0.857
Earthquake	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Weather	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
All Events	0.777	0.771	0.771	0.796	0.783	0.796	0.662	0.656	0.682	0.713	0.701	0.745
Event Type	**City Level Estimations (n=all)**											
Sports	0.747	0.747	0.736	0.758	0.758	0.747	0.582	0.593	0.725	0.670	0.703	0.747
Concert/Show	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
Acc./Terrorism	0.200	0.000	0.200	0.400	0.400	0.600	0.400	0.400	0.400	0.600	0.600	0.600
Demo./Protest	0.816	0.816	0.837	0.816	0.816	0.857	0.857	0.857	0.857	0.878	0.857	0.878
Earthquake	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Weather	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
All Events	0.764	0.758	0.764	0.777	0.777	0.790	0.688	0.694	0.771	0.752	0.764	0.796

TableB.7: Town-level average error distances using the proposed DS methods (in kilometers)

	DRC	YR	DP	ADRC	AYR	ADP	Norm DRC	Norm YR	Norm DP	Norm ADRC	Norm AYR	Norm ADP
Event Type	**Town Level Estimations (n=20)**											
Sports	242.8	242.8	221.0	155.8	155.7	152.4	242.8	242.8	221.0	155.8	155.7	152.4
Concert/Show	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1	3.1
Acc./Terrorism	339.6	339.6	172.4	339.6	339.6	172.4	339.6	339.6	172.4	339.6	339.6	172.4
Demo./Protest	76.8	76.8	81.7	74.9	95.9	75.2	76.8	76.8	81.7	74.9	95.9	75.2
Earthquake	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9
Weather	19.2	19.2	19.2	19.2	19.2	19.2	19.2	19.2	19.2	19.2	19.2	19.2
All Events	185.7	185.7	167.7	130.6	136.3	122.8	185.7	185.7	167.7	130.6	136.3	122.8
Event Type	**Town Level Estimations (n=40)**											
Sports	197.3	197.3	179.3	119.0	119.0	119.0	197.2	197.2	184.0	119.0	119.0	119.0
Concert/Show	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8
Acc./Terrorism	172.4	172.4	0.0	324.1	492.1	172.4	172.4	172.4	0.0	324.1	492.1	172.4
Demo./Protest	90.9	90.9	99.2	74.5	74.5	74.1	90.7	90.7	99.2	74.3	74.3	74.1
Earthquake	15.0	15.0	15.0	15.0	15.0	15.0	14.9	14.9	14.9	14.9	14.9	14.9
Weather	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9	9.9
All Events	155.4	155.4	140.4	107.0	112.8	101.6	155.3	155.3	143.4	106.9	112.7	101.6
Event Type	**Town Level Estimations (n=60)**											
Sports	171.5	171.5	140.0	104.2	111.0	100.2	171.5	171.5	139.6	104.2	111.0	100.1
Concert/Show	3.1	3.1	3.1	3.1	3.1	3.1	2.8	2.8	2.8	2.8	2.8	2.8
Acc./Terrorism	172.4	172.4	0.0	164.0	495.4	12.3	172.4	172.4	0.0	164.0	495.4	12.3
Demo./Protest	80.9	81.1	60.4	72.7	72.7	72.6	72.3	72.5	51.9	72.8	72.8	72.8
Earthquake	14.3	9.9	9.9	14.3	9.9	9.9	7.6	7.6	7.6	7.6	7.6	7.6
Weather	9.9	9.9	9.9	9.9	9.9	9.9	8.3	8.3	8.3	8.3	8.3	8.3
All Events	136.5	136.4	104.9	91.7	107.2	83.8	133.8	133.9	102.3	91.4	107.1	83.7
Event Type	**Town Level Estimations (n=all)**											
Sports	141.9	141.9	143.0	65.2	65.2	61.6	192.0	194.0	193.0	67.2	67.2	63.2
Concert/Show	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Acc./Terrorism	12.3	12.3	0.0	164.0	324.9	12.3	12.3	12.3	0.0	164.0	324.9	12.3
Demo./Protest	89.3	89.3	68.6	71.9	71.9	71.7	90.1	90.1	69.4	71.7	71.7	71.6
Earthquake	14.3	9.9	9.9	14.3	9.9	9.9	7.6	7.6	7.6	7.6	7.6	7.6
Weather	9.9	9.9	9.9	9.9	9.9	9.9	0.0	0.0	0.0	0.0	0.0	0.0
All Events	114.6	114.4	109.0	66.9	72.3	59.2	146.0	147.3	140.5	67.8	73.3	60.1

TableB.8: Town-level match rates using the proposed DS methods

	DRC	YR	DP	ADRC	AYR	ADP	Norm DRC	Norm YR	Norm DP	Norm ADRC	Norm AYR	Norm ADP
Event Type	**Town Level Estimations (n=20)**											
Sports	0.264	0.264	0.275	0.275	0.275	0.286	0.264	0.264	0.275	0.275	0.275	0.286
Concert/Show	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
Acc./Terrorism	0.600	0.600	0.800	0.600	0.600	0.800	0.600	0.600	0.800	0.600	0.600	0.800
Demo./Protest	0.500	0.500	0.500	0.500	0.475	0.500	0.500	0.500	0.500	0.500	0.475	0.500
Earthquake	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Weather	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
All Events	0.345	0.345	0.359	0.352	0.345	0.366	0.345	0.345	0.359	0.352	0.345	0.366
Event Type	**Town Level Estimations (n=40)**											
Sports	0.363	0.363	0.374	0.352	0.352	0.352	0.363	0.363	0.374	0.352	0.352	0.352
Concert/Show	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
Acc./Terrorism	0.800	0.800	1.000	0.600	0.400	0.800	0.800	0.800	1.000	0.600	0.400	0.800
Demo./Protest	0.575	0.575	0.600	0.575	0.575	0.600	0.575	0.575	0.600	0.575	0.575	0.600
Earthquake	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Weather	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
All Events	0.434	0.434	0.455	0.421	0.414	0.434	0.434	0.434	0.455	0.421	0.414	0.434
Event Type	**Town Level Estimations (n=60)**											
Sports	0.396	0.396	0.440	0.407	0.407	0.418	0.396	0.396	0.440	0.407	0.407	0.418
Concert/Show	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750	0.750
Acc./Terrorism	0.800	0.800	1.000	0.600	0.400	0.800	0.800	0.800	1.000	0.600	0.400	0.800
Demo./Protest	0.650	0.650	0.675	0.650	0.650	0.650	0.675	0.675	0.700	0.675	0.675	0.675
Earthquake	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.250	0.250	0.250	0.250	0.250
Weather	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
All Events	0.476	0.476	0.517	0.476	0.469	0.490	0.490	0.490	0.531	0.490	0.483	0.503
Event Type	**Town Level Estimations (n=all)**											
Sports	0.473	0.473	0.495	0.571	0.571	0.593	0.396	0.396	0.396	0.495	0.495	0.516
Concert/Show	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Acc./Terrorism	0.800	0.800	1.000	0.600	0.600	0.800	0.800	0.800	1.000	0.600	0.600	0.800
Demo./Protest	0.700	0.700	0.725	0.700	0.700	0.700	0.700	0.700	0.725	0.750	0.750	0.750
Earthquake	0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.250	0.250	0.250	0.250	0.250
Weather	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000
All Events	0.545	0.545	0.572	0.600	0.600	0.621	0.510	0.510	0.524	0.579	0.579	0.600

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Özdikiş, Özer
Nationality: Turkish (TC)
Date and Place of Birth: 15.09.1980, Bursa, Turkey
Email: oozdikis@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.S.	Computer Engineering Dept., METU	2007
B.S.	Computer Engineering Dept., METU	2003
High School	Bursa Tophane Anatolian Tech. High School	1998

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2003-2005	Aselsan	Software Engineer
2005-2009	Siemens	Software Engineer
2009-2010	Oyak Teknoloji	Software Engineer
2010-2012	Anel Arge	Software Engineer
2012-present	Garanti Teknoloji	Solution Architect

PUBLICATIONS

Journal Publications

O. Ozdikis, H. Oğuztüzün, P. Karagoz. Evidential Estimation of Event Locations in Microblogs Using the Dempster-Shafer Theory, Information Processing and Management, 2016. DOI: 10.1016/j.ipm.2016.06.001 (accepted for publication).

U. Durak, H. Oğuztüzün, C. Köksal-Algın, and O. Ozdikis. Towards composable and interoperable trajectory simulations: An ontology-based approach. Journal of Simulation, vol. 5, pp. 217-229, August 2011, DOI:10.1057/jos.2011.9.

International Conference Publications

O. Ozdikis, P. Senkul, and H. Oguztuzun. Context Based Semantic Relations in Tweets. State of the Art Applications of Social Network Analysis, Lecture Notes in Social Networks, pp. 35-52, 2014.

O. Ozdikis, H. Oguztuzun, and P. Karagoz. Evidential Location Estimation for Events Detected in Twitter. 7th Workshop on Geographic Information Retrieval (GIR '13), pp. 9-16, 2013.

O. Ozdikis, P. Senkul, and H. Oguztuzun. Semantic expansion of hashtags for enhanced event detection in Twitter. Proceedings of VLDB 2012 Workshop on Online Social Systems (WOSS), 2012.

O. Ozdikis, P. Senkul, and H. Oguztuzun. Semantic Expansion of Tweet Contents for Enhanced Event Detection in Twitter. Advances in Social Networks Analysis and Mining (ASONAM '12), pp. 20-24, 2012.

O. Ozdikis, P. Senkul, S. Sinir. Confidence-Based Incremental Classification for Objects with Limited Attributes in Vertical Search. Advanced Research in Applied Artificial Intelligence, pp. 10-19, 2012.

O. Ozdikis, F. Orhan, F. Danismaz. Ontology-based recommendation for points of interest retrieved from multiple data sources. International Workshop on Semantic Web Information Management (SWIM '11), pp. 1:1–1:6, 2011.

O. Ozdikis, U. Durak, and H. Oğuztüzün. Tool support for transformation from an OWL ontology to an HLA object model. In Proceedings of 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools), 15-19 March 2010, Torremolinos, Malaga, Spain, pp. 1-6, 2010.

O. Ozdikis, U. Durak, and H. Oğuztüzün. User-guided transformations for ontology-based simulation design. In Proceedings of Summer Computer Simulation Conference (SCSC), 13-16 July 2009, İstanbul, pp. 75-82, 2009.

G. Laleci, Y. Kabak, A. Dogac, I. Cingil, S. Kirbas, A. Yildiz, S. Sinir, O. Ozdikis, O. Ozturk, A Platform for Agent Behavior Design and Multi Agent Orchestration. Agent-Oriented Software Engineering Workshop (AOSE-2004), the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2004.