The Islamic University of Gaza
Deanship of Research and Graduate Studies
Faculty of Information Technology
Master of Information Technology

الجامعـــــــة الإســـلاميـة - بغـزة
عمادة البحث العلمـي والدراسـات العليـا
كليـــة تكنـولـوجيـا المعلـومـات
ماجستيـر تكنـولـوجيـا المعلـومـات

A New Approach Based on Soft Frequent Pattern Mining for Detecting Significant Events in Arabic Microblogs

تطوير منهجية تعتمد على تنقيب الأنماط المتكررة المرنة للكشف عن الأحداث الهامة في المدونات العربية المصغرة

Jehad H. Zendah

Supervised by

Dr. Ashraf Y. Maghari

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Information Technology

August/2018

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

A New Approach Based on Soft Frequent Pattern Mining for Detecting Significant Events in Arabic Microblogs

تطوير منهجية تعتمد على تنقيب الأنماط المتكررة المرنة للكشف عن الأحداث الهامة في المدونات العربية المصغرة

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل الاخرين لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

Declaration

I understand the nature of plagiarism, and I am aware of the University's policy on this.

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted by others elsewhere for any other degree or qualification.

Student's name:	جهاد حسام الدين صالح زنداح	اسم الطالب:
Signature:		التوقيع:
Date:	01/08/2018	التاريخ:

بسم الله الرحمن الرحيم

الجامعة الإسلامية غزة
The Islamic University of Gaza

عمادة البحث العلمي والدراسات العليا
هاتف داخلي: 1150

الرقم: Ref: ج س غ/35/
التاريخ: Date: 2018/10/02م

نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة البحث العلمي والدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ **جهاد حسام صالح زنداح** لنيل درجة الماجستير في كلية **تكنولوجيا المعلومات**/ برنامج **تكنولوجيا المعلومات** وموضوعها:

تطوير منهجية تعتمد على تنقيب الأنماط المتكررة المرنة للكشف عن الأحداث الهامة في المدونات العربية المصغرة

A new approach based on soft frequent Pattern Mining for detecting significant events in Arabic Microblogs

وبعد المناقشة التي تمت اليوم **الاربعاء 19 ذو القعدة 1439هـ** الموافق **2018/08/01م** الساعة الحادية عشر صباحاً، في قاعة مؤتمرات الكلية اجتمعت لجنة الحكم على الأطروحة والمكونة من:

د. أشرف يونس مغاري	**مشرفاً ورئيساً**	.....
د. باسم عمر العجلة	**مناقشاً داخلياً**	.....
د. محمد عوض عوض الله	**مناقشاً خارجياً**	.....

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في **كلية تكنولوجيا المعلومات/برنامج تكنولوجيا المعلومات.**

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله تعالى ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

عميد البحث العلمي والدراسات العليا

أ. د. مازن إسماعيل هنية

	اللغة	310683	الرقم العام للنسخة

التاريخ: 2018/10/17

الموضوع/ استلام النسخة الإلكترونية لرسالة علمية

قامت إدارة المكتبات بالجامعة الإسلامية باستلام **النسخة الإلكترونية** من رسالة

الطالب/ محمد حسام الدين صلاح زنون

رقم جامعي: 120120176 قسم: تكنولوجيا المعلومات كلية: تكنولوجيا المعلومات

وتم الاطلاع عليها، ومطابقتها بالنسخة الورقية للرسالة نفسها، ضمن المحددات المبينة أدناه:

- تم إجراء جميع التعديلات التي طلبتها لجنة المناقشة.
- تم توقيع المشرف/المشرفين على النسخة الورقية لاعتمادها كنسخة معدلة ونهائية.
- تم وضع ختم "عمادة الدراسات العليا" على النسخة الورقية لاعتماد توقيع المشرف/المشرفين.
- وجود جميع فصول الرسالة مجمَّعة في ملف (WORD) وآخر (PDF).
- وجود فهرس الرسالة، والملخصين باللغتين العربية والإنجليزية بملفات منفصلة (PDF +WORD)
- تطابق النص في كل صفحة ورقية مع النص في كل صفحة تقابلها في الصفحات الإلكترونية.
- تطابق التنسيق في جميع الصفحات (نوع وحجم الخط) بين النسخة الورقية والإلكترونية.

ملاحظة: ستقوم إدارة المكتبات بنشر هذه الرسالة كاملة بصيغة (PDF) على موقع المكتبة الإلكتروني.

والله ولي التوفيق،

توقيع الطالب

إدارة المكتبة المركزية

أ. محمود خليل الداعور

289

Abstract

Recently, Microblogs have become the new communication medium between users. It allows millions of users to post and share content of their own activities, opinions about different topics. Posting about occurring real-world events has attracted people to follow events through microblogs instead of mainstream media. As a result, there is an urgent need to detect events from microblogs so that users can identify events quickly, also and more importantly to aid higher authorities to respond faster to occurring events by taking proper actions.

While considerable researches have been conducted for event detection on the English language. Arabic context have not received much research even though there are millions of Arabic users. Also existing approaches rely on platform dependent features such as hashtags, mentions, retweets etc. which make their approaches fail when these features are not present in the process. In addition to that, approaches that depend on the presence of frequently used words only do not always detect real events because it cannot differentiate events and general viral topics.

In this thesis, we propose an approach for Arabic event detection from microblogs. We first collect the data, then a preprocessing step is applied to enhance the data quality and reduce noise. The sentence text is analyzed and the part-of-speech tags are identified. Then a set of rules are used to extract event indicator keywords called event triggers. The frequency of each event triggers is calculated, where event triggers that have frequencies higher than the average are kept, or removed otherwise. We detect events by clustering similar event triggers together. An Adapted soft frequent pattern mining is applied to the remaining event triggers for clustering.

We used a dataset called Evetar to evaluate the proposed approach. The dataset contains tweets that cover different types of Arabic events that occurred in a one month period. We split the dataset into different subsets using different time intervals, so that we can mimic the streaming behavior of microblogs. We used precision, recall and f-measure as evaluation metrics. The highest average f-measure value achieved was 0.717. Our results were acceptable compared to three popular approaches applied to the same dataset.

Keywords*: Event Detection, Frequent Pattern Mining*

الملخص

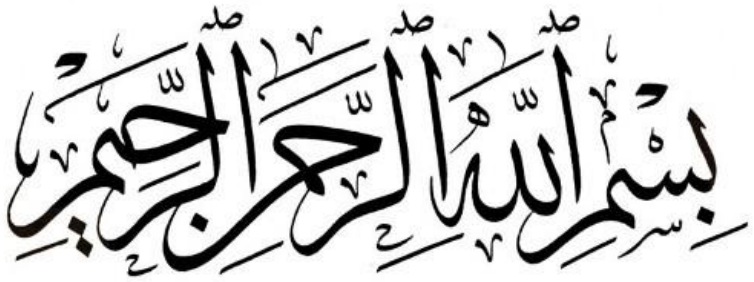
حديثاً، أصبحت المدونات الصغيرة وسيلة إتصال جديدة بين المستخدمين. فقد سمحت لملايين المستخدمين من نشر ومشاركة محتويات متعلقة بأنشطتهم وأرائهم عن مواضيع مختلفة. إن نشر المحتوى المتعلق بالأحداث الجارية في العالم الحقيقي قد جذب الناس لمتابعة الأحداث من خلال المدونات الصغيرة بدلاً من وسائل الإعلام الرئيسية. نتيجة لذلك، أصبحت هناك حاجة طارئة لكشف الأحداث من الدونات الصغيرة حتى يتمكن المستخدمون من تحديد الأحداث الجارية بشكل أسرع، أيضاً والأهم من ذلك، مساعدة السلطات العليا للإستجابة بشكل سريع في عمل اللازم عند حدوث حدثاً ما.

في حين أنه أجريت العديد من الأبحاث على كشف الأحداث باللغة الإنجليزية، إلا أن السياق العربي لم يأخذ نصيباً وفيراً في هذا المجال، على الرغم من وجود الملايين من المستخدمين العرب. ايضاً، العديد من المناهج الموجودة حالياً تعتمد على خصائص معتمدة على المنصة المستخدمة في البحث مثل وسم الهاشتاق، وتأشيرة المستخدم، وإعادة التغريد، إلخ. مما يجعل النهج المستخدم يتأثر سلباً في حال لم تكن هذه الخصائص موجودة أثناء عملية الكشف عن الأحداث. بالإضافة الي ذلك، المناهج التي تعتمد فقط على وجود الكلمات الأكثر استخداما لا تكشف الاحداث الحقيقية دائما لانها لا تستطيع التفرقة بين الحدث والمواضيع العامة الشائعة.

في هذه الأطروحة، نقترح نهج لكشف الأحداث العربية من المدونات الصغيرة. أولاً نقوم بجمع البيانات، ثم نقوم بتجهيزها من خلال تحسينها وتقليل الشوائب فيها. يتم تحليل نص الجملة لإستخراج الأوسمة الخاصة بأجزاء الكلام. بعدها نقوم بتطبيق مجموعة من القواعد لإستخراج الكلمات الدلالية التي تشير إلي الأحداث و تسمى مشغلات الأحداث. يتم حساب عدد تكرار كل مشغل حدث، بحيث يتم الإحتفاظ على المشغلات التي لها عدد تكراراكبر من المتوسط ويتم حذف عكس ذالك. يتم الكشف عن الحدث من خلال تجميع مشغلات الأحداث المتشابهة مع بعضها. حيث نقوم بتطبيق إصدار ملائم من خوارزمية "التنقيب الناعم عن الأنماط المتكررة" على مشغلات الأحداث التي تبقت لكي يتم تجميع المتشابه منها.

قمنا بإستخدام قاعدة بيانات تسمى "Evetar" لتقييم النهج المقترح. حيث تحتوي قاعدة البيانات على تغريدات تغطى عدة انواع من الأحداث العربية التي حدثت خلال فترة شهر. لكي نقوم بمحاكاة طريقة تدفق البيانات في المدونات الصغيرة، قمنا بتقسيم البيانات إلي عدة مجموعات بناءاً على فترات زمنية مختلفة. تم إستخدام كل من "Precision"، "Recall"، "F-Measure" كمقياس للتقييم، حيث كانت أعلى متوسط قيمة للـ "F-Measure" تم الحصول عليها هي 0.717. تعتبر النتائج التي حصلنا عليها مقبولة مقارنة مع ثلاث مناهج مشهورة تم تطبيقها على نفس قاعدة البيانات.

كلمات دلالية: الكشف عن الأحداث، التقيب عن الأنماط المتكررة.


بسم الله الرحمن الرحيم

﴿قَالُوا سُبْحَانَكَ لَا عِلْمَ لَنَا إِلَّا مَا عَلَّمْتَنَا إِنَّكَ أَنتَ الْعَلِيمُ الْحَكِيمُ﴾

صدق الله العظيم

Dedication

To my father & mother,

To my beloved wife,

To my beautiful and adorable son

& To all who supported me

I dedicate this work

Acknowledgement

First and foremost, thanks to Allah for giving me the power and help to accomplish this research. Without the grace of Allah, I was not able to accomplish this work.

Many thanks and sincere gratefulness goes to my supervisor **Dr. Ashraf Y. Maghari**, for his help, guidance, and continuous follow-up in this research.

Special thanks also to my great family for the endless support. Without my family, I would never have been able to achieve my goals.

Table of Content

List of Tables

List of Figures

List of Abbreviation

API	Application Programming Interface
ED	Event Detection
ET	Event Trigger
FPM	Frequent Pattern Mining
FSD	First Story Detection
LDA	Latent Dirichlet Allocation
NLP	Natural Language Processing
POS	Part of Speech
SFPM	Soft Frequent Pattern Mining
SNS	Social Networking Site
TDT	Topic Detection and Tracking
TF-IDF	Term Frequency-Inverse Document Frequency

Chapter 1
Introduction

Chapter 1

Introduction

This chapter is an introduction to the thesis, first it gives a brief description of event detection, Arabic event detection, and then a brief description about approaches of event detection. In addition, it states the thesis problem, the research objectives, the significance of the thesis, the scope and limitation of the thesis work, and the research methodology.

1.1 Overview

Nowadays, Microblogs have become the main virtual environment for connecting people and sharing digital content. It allows users to post and share short text, images, and/or short length videos of any type. This content is delivered to a network of followers or virtual friends of the content creator at relatively no time. Content type depends on the user interests and situation. At the occurrence of an event, users post details about the event with their friends. **Figure 1.1** shows Twitter posts (tweets) containing events details written in Arabic language. With the instant delivery, events news usually spread faster and reach wider audience in microblogs compared to mainstream media (Alkhamees & Fasli, 2016).

Events are real-world occurrences that take place in a certain geographical location over a certain time period (Allan, 2002). Capturing information about an event can help in many aspects. For example, it can help in accelerating the crisis response when the information about disastrous events are retrieved at the time of its occurrence. Also it can help people to easily track occurring events. Other applications of event detection is the Facebook Safety Check shown in **Figure 1.2** in which users confirm they are safe when a natural or a man-made disaster occurs near them. Capturing event details from microblogs is not an easy task, because events information are covered with a huge amount of unnecessary data such as random topics, users daily activities, spam or any kind of data that are not related to an event.

Figure 1.1: Tweets corresponding to real life events written in Arabic.

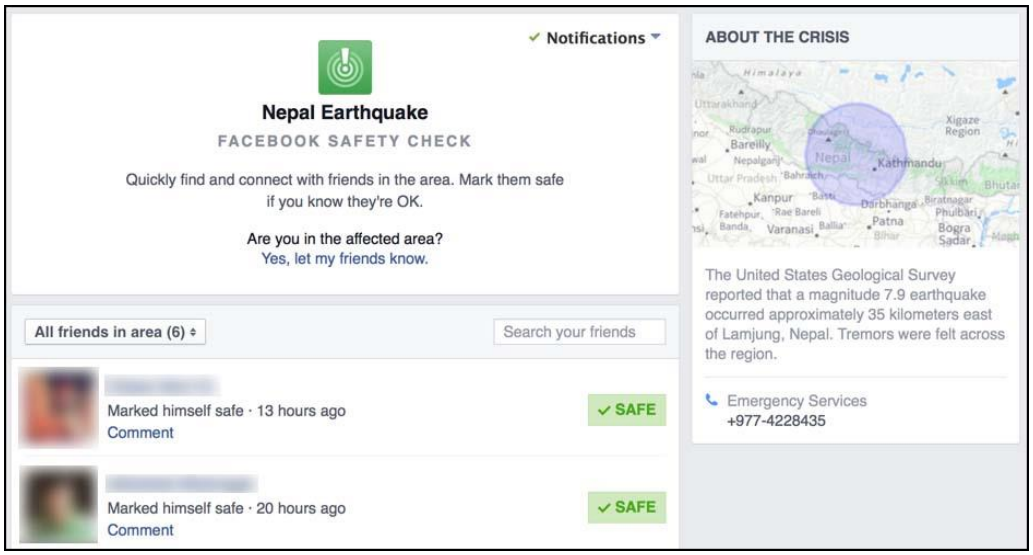


Figure 1.2: The user interface of Facebook safety check application.

In Microblogs context, Dou, Wang, Ribarsky, and Zhou(2012) defined an event as "*An occurrence causing change in the volume of text data that discusses the associated topic at a specific time. This occurrence is characterized by topic and time, and often associated with entities such as people and location*". Event Detection (ED) definition depends on the task held by the researcher. In general ED is the process of discovering and identifying new or previously unidentified events from a set of documents (Allan, 2002).

Twitter is a popular microblogging service. It allows users to send 280-character messages called tweets (Twitter, 2017). According to Omnicore(2018) , there are 100 million users that uses Twitter daily with 500 million tweets per day. It gained great attention among users and organizations due to its ease of use and simplicity and its ability to reach huge network of users in relatively no time. It attracted researchers due to its open Twitter Stream Application Programming Interface (API) that allows researchers to analyze live feeds from the stream to extract different types of knowledge such as events.

Event detection problem has been addressed by many researchers; they employed different techniques from many fields such as machine learning, data and text mining, and natural language processing. The first event detection research program is the Topic Detection and Tracking (TDT) conducted by (Allan, 2002). The techniques introduced are meant to monitor different newswire sources so that users can be aware about occurring events. The approach was applied on full text of well written news articles, however, with the emergence of microblogs, new challenges are introduced when using these techniques. For example, content generated by microblogs users is constrained to be very small, thus if the traditional Term Frequency-Inverse Document Frequency (TF-IDF) is used in such short documents it will result in a sparse vector issue. Also microblogs posts are noisy. According to Hurlock and Wilson(2011) posts in microblogs do not always refer to an actual event or a subject of importance, but most of the time the posts contain meaningless or uninteresting daily life activity. In addition to that, the content is generated by anyone, thus it is very likely the content will have grammatical errors, informal words, or incomplete context. **Figure 1.3** shows an example of such problems. Generally, event detection algorithm should tackle these challenges to produce better results.

Figure 1.3: An Arabic tweet that contains informal words in an incomplete context.

According to Atefeh and Khreich(2015) , event detection techniques are classified based on the event type, the detection task, and detection method. The type of event can be specified or unspecified, where the specified event detection techniques require prior knowledge about the event –mainly a related keyword or a query. On the other hand, unspecified event detection techniques does not require any prior knowledge, such techniques require an ongoing monitoring and analysis for the incoming documents to find an increase number of keyword appearance-count, which can be an indicator of a potential event. Based on the detection task, the detection process can be used for new or retrospective events.

Many existing approaches of event detection are tested on English data. For the best of our knowledge only few researches have been conducted for the Arabic Language (Alsaedi & Burnap, 2015). The Arabic language introduces many challenges in the Text mining and Natural Language Processing (NLP) fields, this is due to its vocabulary richness, and its morphological and orthographic nature Farghaly and Shaalan (2009). These challenges are inherited in the event detection problem. Other existing approaches utilize platform specific features such as hashtags, retweets, and followers and external knowledge to enhance event detection (Doulamis, Doulamis, Kokkinos, & Varvarigos, 2016), (Weng & Lee, 2011). The problem with these approaches is that when these attributes do not appear for different reasons in the process, the accuracy of the event detection will be affected. In addition to these limitations, most of the approaches found in the literature depend on the burst behavior of specific words, but it is not true that every word that shows burst is related to an event. For example, in the Arabic context, users usually post praising words to Allah such as Subhan Allah Wa Behamdeh "سبحان الله وبحمده". These words will show bursts sometimes but in reality they do not belong to any event.

In this thesis we propose an approach for detecting significant events from Arabic microblog posts that tackle these challenges. Our approach rely on extracting event triggers from post text using pre-defined rules applied to the Part of Speech (POS) tags of the tweet. This process is essential to separate posts that may contain event occurrence from posts that do not. A Soft Frequent Pattern Mining Approach is applied to the posts containing event trigger. The resulted cluster are treated as detected event.

The proposed approach targets the Arabic content to be a first stage for early event reporting, situational awareness and summarization for Arabic audience. Arabic event detection can be very useful in crisis response applications, as many places in the Middle East have conflicts, thus different types of events can occur.

1.2 Statement of the Problem

Many research efforts have been made in event detection that consider English text, but only a few researches that consider the Arabic text. Also existing approaches rely on platform dependent features such as hashtags, retweets, mentions, etc., thus, the accuracy of these approaches will be affected by the absence of these features. On the other hand, other approaches identify frequent words as the event word which is not always true, thus it will not differentiate between popular topics and real events.

Developing an event detection approach that depends on the text characteristics only instead of the platform dependent features can maintain the approach accuracy and can make it reusable on different platforms. Also using syntactic features of a word alongside its frequency can enhance detecting real events.

In this research we need to answer the question of how to detect events from Arabic text without relying on platform dependent features, but using only event triggers.

1.3 Objectives

1.3.1 Main Objectives

To develop an approach that detect events from Arabic Microblog posts text only.

1.3.2 Specific Objectives

- To collect and use an Arabic Twitter based dataset for proper for event detection task.
- Extract tweets data from the internet and preprocess it.
- Implement an algorithm method for extracting event triggers.
- Implement an algorithm for extracting significant events using event triggers.
- Evaluate the approach accuracy using recall, precision and F-measure.

1.4 Signification

- Early detection of event allows authorities to respond faster which can save lives.
- Few researches address the Arabic event detection, thus this research will help to cover this gab.
- Implement event detection in posts text only without relying in a specific Microblog feature such as hashtags, followers and retweets to make the approach portable to other platforms.
- Improve the performance of event detection by considering posts contain event triggers.

1.5 Scope and Limitations

We aim to build an event detection prototype from microblogs that adhere to the following limitations and assumptions:

1. We will use a Twitter based dataset.
2. Our work is limited to Arabic tweets only.
3. We assume event related tweets have one or more event triggers.
4. Experiments and Evaluation are held using offline dataset.
5. Our approach will rely on the text characteristics only.

1.6 Research Methodology

The following is the approach we used in this thesis to achieve our objectives:

- **Research and survey:** reviewing the recent literature related to the thesis problem statement and the research question. We analyze the existing methods and approaches. We identify the drawbacks and the lack of existing approaches. We then formulate the strategies and solutions to overcome these drawback.
- **Data Collection:** identifying a proper Arabic Twitter-based dataset and building a small tool for converting tweets' id into actual content.

- **Building the Approach:** the structure of our proposed approach include the following general steps:
    1. Data Collection: extract tweets content from the internet using Twitter API
    2. Data Preprocessing: prepare the data and enhance its quality with the following steps:
        - Tokenization.
        - Cleaning.
        - Normalization.
        - Stemming.
        - Part-of-Speech (POS) tagging.
        - Stop word removal.
    3. Event trigger extraction: identify and extract frequent event triggers, this includes the following steps:
        - Apply pre-defined rules on the identified part-of-speech (POS) tags.
        - Extract event triggers
    4. Significant event detection: similar event triggers are grouped together. This step includes the following
        - Identify the top event triggers
        - Applying Soft Frequent Pattern Mining Algorithm to create groups of similar event triggers.
- **Evaluation:** we evaluate our approach using different measure, recall, precision and F-measure.
- **Results and discussions:** we analyze the results and justify our findings.

1.7 Thesis Outline

This thesis is organized as follows: Chapter 2 present the literature review about event detection in general and Arabic event detection in specific. Chapter 3 presents the proposed approach for Arabic event detection. Chapter 4 describes the implementation of the approach. Chapter 5 presents the experiments and evaluations of the results. Finally chapter 6 concludes the thesis and suggests the future work.

Chapter 2
Literature Review

Chapter 2

Literature Review

In this chapter we introduce the literature review related to our work. We review event detection approaches applied on English and Arabic datasets, also we review approaches that depend on platform specific features and approaches that depend on bursty words.

2.1 Overview

Many approaches of event detection in microblogs uses platform specific features which make the approach accuracy dependent on these features and cannot be ported to other platforms. Also, other approaches focus on finding a burst of words in the stream to identify hot topics. These approaches will detect a topic without any consideration if that topic is an actual event or a viral general topic. In addition to that, considerable research have been done and tested on English data but only few were made that consider the Arabic language. In the following sections we review some of these researches.

2.2 Platform Feature-Based Approaches

2.2.1 English based Approaches

Phuvipadawat and Murata (2010) introduced an approach that tracks and ranks breaking news from Twitter stream. Tweets are retrieved using predefined search queries and indexed using Apache Lucene. Term Frequency-Inverse Document Frequency (TF-IDF) is used for tweet representation with a bias factor for named entities, hashtags, and usernames. Additional weight is added to the tweet based on a set of features such as the number of followers which represents reliability and the number of retweeted messages which represents popularity within a timeframe. A tweet is assigned to a cluster if it is similar to the first tweet added to that cluster and similar to the top K terms in the cluster. The authors claim that adding more weight to Named Entities, Hashtags, and Usernames produces betters results.

Becker, Naaman, and Gravano (2011) proposed an approach to detect real-world events from Twitter with avoiding trendy topics. They used incremental online clustering algorithm to cluster similar tweets without specifying the number of

clusters. A set of features are extracted from each clusters to classify event and non-event clusters. These features are temporal features which represent the most frequent terms in the cluster. Social features which are represented by the percentage of messages containing social interaction such as retweet, reply, and mentions. Topical features that represent the central topic of a cluster, since the authors assume that clusters containing different topics tend to be non-event clusters, and clusters that contains central theme is likely to be an event cluster. In addition to the previous features, twitter hashtag feature is also used. The authors claim that clusters with hashtags of more than one concatenated words are likely to be a general discussion and not real-world event. Event clusters are identified by the classifier. Top K clusters are selected using the technique of selection introduced by Petrović, Osborne, and Lavrenko(2010) . A manually collected dataset from Twitter is used for evaluation. Human annotators were used to label the resulted clusters using a subset of tweets in each cluster that belong to the top 10 frequent terms. A set of 100 randomly selected clusters were used to calculate the F-Measure of the classifier which gives 0.837. Precision@K and the Normalized Discounted Cumulative Gain (NDCG) evaluation metrics are also used.

Cordeiro (2012) proposed a lightweight approach based on continuous wavelet transformation analysis of Twitter hashtag occurrences. Peaks and local maxima are calculated to find hashtags with higher signal which indicate an event. Latent Dirichlet Allocation (LDA) is then used on the collection of tweets that belongs to the hashtags to create topic inference model. We do not predict user behavior, thus depending on hashtags only as an indicator of an event is not always accurate, as users may not attach a hashtag when tweeting about an event. Also hashtags could be the presence of spam tweets.

Guille and Favre (2014) an approach is proposed based on the anomaly of dynamic link (mentions) creation frequencies. A normal distribution is created for each word that co-occur with a mention. The anomaly of a word is detected in an interval of time when the distribution value of this word exceeded the expectation (mean). The magnitude of the word is also calculated by finding the algebraic area of the distribution at that interval. The word is identified as an event in an interval when it reaches the highest magnitude. To retrieve the most K words that co-occur with the

event word, the similarity between their temporal dynamics and the event word temporal dynamic is calculated and compared with a threshold value. The approach was evaluated over English and French datasets. Parameters of the approach were set as 30 minutes for window size, K=10, and a similarity threshold value of 0.7. The evaluation was conducted manually using human annotators. Precision and F-score were used as measurements. With the English corpus the results were 0.775 and 0.682 for precision and f-score respectively. With the French corpus the results were 0.825 and 0.825 for precision and f-score respectively. Despite the approach does not rely on external knowledge, it depends on Twitter specific features (mentions), also the approach performs poorly when mentions are not considered.

Doulamis et al. (2016) proposed an approach based on a multiple assignment graph partitioning algorithm where event is represented by a cluster of related words. The authors addresses the problem of message posting delays which lead to event attributes being scattered in different timestamps, thus the significance of event-related words will be decreased as time goes on. Words are modeled using three Twitter-based information theoretic metrics. Conditional Word Tweet Frequency-Inverse Trend Word Tweet Frequency (CWTF-ITWTF), which is a time varying measurement similar to the popular IDF-TF. The objective of this measurement is to decrease the weight of trendy ongoing events. Word Frequency-Inverse Trend Word Frequency (WF-ITW), which is a time varying measure that consider the frequency of keywords. Lastly, Weighted Conditional Word Tweet Frequency-Inverse Trend Weighted Conditional Word Tweet Frequency (WCWTF-ITWCWTF). This measure depends on features from Twitter such as number of followers, number of retweets to find the importance of a keyword. A fuzzy time series signal is produced from the three metrics. The approach is evaluated over a manually collected dataset using Twitter stream API. A time window of size 6 hours is used over a one month time horizon. To create a ground truth, for each time window the most frequent keywords are extracted and presented to experts to annotate them. The evaluation measurements used are keyword recall/precision and F1-Score. This approach cannot be used in microblogging streams that do not produce these features, however, our approach does not consider any Twitter based features. It also depends only on the bursty pattern of a set of keywords thus the resulted detected event can be a trending topic and not a

real-world event. The approach require different parameter tuning to achieve good F1-Score.

Yılmaz and Hero (2016) proposed an approach for event detection using multimodal factor analysis model. The approach depends on two features set. The hashtags' bag of words created from all the tweets containing the hashtag. Also, geolocation vector containing the latitude and longitude values of all tweets containing the hashtag. A probabilistic generative modal is used to fuse these features and an expectation maximization algorithm is derived for finding the maximum likelihood estimates of the model parameters. The approach assumes that hashtags are used during event occurrence and tweets containing these hashtags are geographically closed together.

Shao et al. (2017) proposed a generic framework for event detection that depends on dynamic multivariate graph. A user-to-user undirected graph is built where vertices are represented as users, and edges are represented as the follow relationship between the users. Every vertex contain a textual feature vector of domain-specific keywords. The approach focuses on the search of evolving subgraphs over time with anomalous features. The evaluation metrics used are false positive rate (FPR), true positive rate (TPR).

2.2.2 Arabic-Based Approaches

Alsaedi and Burnap (2015) proposed a novel Arabic event detection framework from Twitter dataset. In the framework, the data undergoes a preprocessing step to enhance the data quality. A Naïve Bayes classifier is used to distinguish event-related tweets from irrelevant tweets. The classifier is trained on 1500 tweets and their terms are used as features. Tweets are represented by a set of features which include temporal, spatial, and textual features such as retweet ratio, mention ratio, hashtag ratio, tweet sentiment, etc. Tweets are then clustered together to distinguish events using an online clustering algorithm. The average weight of each term in all document in a cluster is used as the centroid of the cluster. The approach output was evaluated by splitting the dataset into days and then calculate the average value of precision which was 80.24% for disruptive events. Our approach uses a same schema by filtering tweets before further processing. Using the terms in a small dataset to train the

classifier will have two drawbacks as stated in (Wang, Tokarchuk, & Poslad, 2014), First, it could decrease classifier accuracy as the keywords introduced in the dataset are subjective to specific events and can lead to undesirable result when new events emerges, Second using the bag of word will result in vector sparseness especially when used in dynamic and rapid changing corpus. However our approach uses a set of rules that examine the tweet's syntax and determine if it contains an event trigger or not. Tweets that do not have event trigger are filtered out.

These approaches that depend on platform-specific features will detect event in these platforms only and their approaches will fail or have low accuracy when used in other platforms. However, our approach depends on the text characteristics only which make the approach maintain its accuracy in the absence of these features.

2.3 Bursty-Words Based Approaches

Other approaches depends on the presence of words that create bursts or spikes in the words frequency distribution.

Petrović et al. (2010) presented a first story detection (FSD) in which the first text of a detected topic or event is extracted. A local sensitive hashing (LSH) is used to reduce the search space, thus similarity measure is calculated between a tweet and a subset of neighbors identified by LSH. If LSH fails to find a neighbor, a limited search is performed considering only 2000 tweets. Threads that grow faster are identified as events. A preprocessing step is conducted that removes non English characters and Twitter-specific features such as mentions and hashtags. A manual evaluation procedure was used to calculate precision. Limiting the search space could boost the performance, however, in twitter stream, events features are very likely to be scattered over time (Doulamis et al., 2016), thus this approach will fail to detect events in this scenario. In our work, we used a greedy search approach considering only important words.

Weng and Lee(2011) proposed an approach that depends on clustering wavelet signals. A signal is built for each word using wavelet analysis to reduce space and storage. Auto-correlation is calculated for each signal. Signals that produces skewed auto-correlation are identified as insignificant words and then removed. Similarity between words is calculated using the cross-correlation between every pair of words.

Similar words are determined using a threshold value where words that have a similarity higher than the threshold are clustered together. The approach is evaluated over a manually collected dataset from Twitter where non-English characters are removed from the text. The evaluation is conducted manually considering only precision, as the authors cannot enumerate all events that occurred at the time of collection, thus recall is discarded. Different experiments with different configuration are used and the best result achieved was 16.7%. Clustering based on only a pair of words will produce generic events or topics (Petkos, Papadopoulos, Aiello, Skraba, & Kompatsiaris, 2014). For example, if a bombing event occurred in the same country with different locations, then using such algorithm will detect event that does not differentiate between the two different locations.

Petkos et al. (2014) introduces a novel method called Soft Frequent Pattern Mining (SFPM). The method is used to tackle the problem of using patterns of only pairs of terms for event detection. The method uses the same concept of the Frequent Pattern Mining (FPM) technique in that, it examines the simultaneous co-occurrence patterns of degree greater than two, however, it is less strict than FPM as it does not require all terms in the pattern to be frequent in the same document, but only a large subset of the terms are frequent in same document. The approach consist of two main components. Term selection in which a fixed number of terms are selected for grouping. The selection process depends on the existence of a randomly collected tweets called reference corpus. The likelihood of appearance is estimated for each term in the reference corpus and the incoming tweets corpus. The ratio of the likelihoods of appearance is then computed. Terms with the highest ratio will be more significant as the term has a frequency higher than usual in the two corpora. The second component is the implementation of the algorithm itself on the selected top terms. Using a static reference corpus will result in a biased behavior for the term selection algorithm, as new emerging term will produce lower ratio, thus it will not be selected

Gaglio, Re, and Morana (2015) proposed an enhancement for SFPM that tackle the limitation of the selection process. The new method uses dynamic reference corpus. Also it depends on the product of a combination of measures such as the TF-IDF of the term, a bias factor used if the term is a Named Entity, and the likelihood measure. Terms with higher values are selected. Both approaches require identifying the number

of top terms with the presence of reference corpus. However, our approach identifies event triggers as important words, and these are extracted using syntactical analysis only.

Alkhamees and Fasli (2016) proposed an approach based on the traditional FP-Growth method. FP-Growth produces the most frequently used combinations of words that co-occur together in a tweet. Determining what is most frequent depends on a fixed threshold value. On the other hand, the approach introduces a dynamic procedure for calculating the threshold, so that it can handle the dynamic nature of words size over time. The procedure depends on a combination of statistical values which are the average and median of the words' frequencies. A preprocessing step is performed for text tokenization and removing stop words, mentions, URLs, and hashtags. A post processing step is also performed to eliminate duplicate patterns. Duplicate patterns are determined by calculating the cosine similarity between patterns with a threshold of 0.75. The approach was tested using two datasets manually collected by querying the Twitter stream API using a set of keywords that identify two event, the UK General Elections 2015 and the Greece Crisis 2015.

Katragadda, Benton, and Raghavan (2017) introduced a multiple source approach that collect data from twitter stream and newswire websites. Every source is considered as an independent stream. Every stream undergoes two stages. First a weighted graph is built in which nodes represent words and edges represent the number of documents in which the two connected words co-occur together. A pruning process is conducted on the graph to keep emerging and important words. The multiple sources are merged by merging the pruned graph. Events are detected using voltage based clustering algorithm on the resulted graph. The approach was tested using two sources Twitter and Tumblr and achieved F-Measure of 0.897.

Ferracani, Pezzatini, Landucci, Becchi, and Bimbo (2017) An event detection approach for specific regions is proposed by introduced and approach that detect events in cities and specified regions by analyzing tweets containing longitude and latitude information (geotags). For each area a time series (TS) is created containing the number of users posting a geotagged tweet in hourly basis. Abnormal TS are identified using Dynamic Time Warping which measure the distance between the TS

and the average TS. In every abnormal TS, Crest-Detection algorithm is used to find anomalies in the tweets distribution then the clustering algorithm Density-based spatial clustering of Applications with Noise (DBSCAN) is used to group similar tweets and geotags as the detected event. The algorithm was evaluated with a manually collected Twitter dataset using precision only as an evaluation metric with a result of 0.57.

Approaches that identify events using the presence of frequently used words only detect general topics along with event-centric topic. On the other hand, our approach focuses on the presence of frequently event triggers, thus we claim that our approach detects only event-centric topics.

2.4 Arabic Event Trigger Extraction

In our work, we depend on event triggers as important keyword of the text. They are words that indicate the occurrence of an event. Many researches have been made that uses event trigger for event information extraction.

Mohammad and Qawasmeh (2016) proposed an unsupervised approach for event extraction out of Arabic tweets. The approach tags the event expression and the related entities and link them to the knowledge base. Event expression are identified using a set of rule based on the guidelines provided by the Arabic Annotation Guidelines for Events (Consortium, 2005). This approach process each tweet independently from the other, thus it will fail to identify significant events. Our approach works on the burst behavior of event triggers, thus only trending/significant events are detected.

Abuleil (2007) presented an approach to extract events from Arabic text. The approach is based on maintaining a list of event keyword that represent an event such as disasters keywords, bombing keywords, etc. the objective of the approach is to extract events, its participants, place and time to answer questions about the event. The approach is tested on full news article from Aljazeera website.

Aliane, Guendouzi, and Mokrani (2013) introduced an approach to recognize verbal events based on unsupervised segmentation algorithm that identify morphemes and affixes of words without using lexicons or predefined list of affixes. A Part-of-Speech tagger is used on the morphemes to identify verbs and nouns. Verbs are annotated as event. The approach is tested on 30 full articles from the web.

Baradaran and Minaei-Bidgoli (2015) developed an approach to extract people death events from historical Arabic texts. The approach tested different classification methods to classify sentences if they are on event or off event. First, a rule-based method is used in which a set of rules are developed to extract events. Also, a supervised method based on support vector machine (SVM) is considered, where a combination of features are used to train the classifier. These features includes words and phrases of each sentence, word stem of each word in a sentence, and the words Part-of-Speech tags. Lastly, the third method used is semantic analysis using lexical chain. The approach is evaluated using a large dataset of death events. For classification the results showed that rule-based classification showed good results and the errors were due to the POS tagger used and the stemming errors. However SVM outperformed both rules-based and the semantic analysis method.

Hkiri, Mallat, and Zrigui (2016) events are extracted from unclassified news articles where verbs, nouns, and adjectives lemmas are compared with a predefined list of event verbal triggers and event nominal triggers to identify event sentences. Then event attributes are extracted using semantic analysis.

Most Arabic event trigger extraction approaches are used on sentences in large formal articles. We assume tweets that contain event trigger are event related and vice versa, thus it can be used as a mechanism to remove insignificant tweets. Also, by considering event triggers we can obtain important words in tweets, in addition to that, we avoid detecting trendy topics.

In our approach we used a set of rules introduced in (Consortium, 2005) to extract event triggers. The rules depends on the Part-of-Speech (POS) tags of a sentence. By using a rule-based method over POS we can cope with the dynamicity nature of microblogs. **Table 2.1** shows a summary of reviewed researches.

2.5 Summary

In this chapter we presented a review of some related works in the field of event detection, and identified approaches that depends on specific platform features and its drawbacks when these features are not presented. Also we showed the problem of depending on bursty keywords only, which make the approaches unable to identify similar events. We also showed the lack of research done for Arabic event detection, and the focus was only on event trigger extraction from well-written full articles.

Table 2.1: Summary of Reviewed Event Detection Approaches

Reference	Technique	General Features	Twitter Features	Dataset Language	Dataset Type	Evaluation Method	Detected Content	Drawbacks
(Petrović et al., 2010)	Clustering based on Locality Sensitive Hashing (LSH)	TF-IDF	-	English	Tweets	-	General Topics	Not suitable for scattered event details
(Phuvipadawat & Murata, 2010)	Online clustering	TF-IDF biased toward Named Entities, Hashtags, Usernames	# of followers, # of retweets, Hashtags, Mentions	English	Tweets	Precision, F-Measure (Manually)	General Topics	Limited to the events that are retrieved by the search query
(Weng & Lee, 2011)	Discrete wavelet analysis, graph-based clustering	Wavelet signal of individual words	-	English	Tweets	Topic recall, keyword precision, keyword recall (Manually)	General Topics	Detect generic details of the event, thus similar events will be treated as one
(Becker et al., 2011)	Support Vector Machine classifier, Online Clustering	TF-IDF for classifying and clustering, Term Frequency, Topical Features	# of retweets, # of replies, # of Mentions,	English	Tweets	Topic recall. Keyword precision. keyword recall (Manually)	Events	Depends heavily on Twitter features, thus the approach cannot be introduced to other platforms

Reference	Technique	General Features	Twitter Features	Dataset Language	Dataset Type	Evaluation Method	Detected Content	Drawbacks
(Cordeiro, 2012)	Continuous wavelet analysis, Latent Dirichlet Allocation (LDA)	-	- frequency	English	Tweets	-	General Topics	Depends on the presence of hashtags, which make the approach miss event tweets that do not contain a hashtag
(Guille & Favre, 2014)	Statistical model analysis using normal distribution and expectation	Individual words	Mentions	English	Tweets	Precision, F-Measure (Manually)	General Topics	The approach performs poorly when mentions are not present in the tweets
(Petkos et al., 2014)	Soft Frequent Pattern Mining	Document vector for top terms using likelihood of appearance in static reference random corpus	-	English	Tweets	Topic recall, keyword precision, keyword recall (Manually)	General Topics	Biased term selection behavior due to the use of static reference corpus, Require fixed number of terms for processing
(Gaglio et al., 2015)	Soft Frequent Pattern Mining	Document vector for top terms using product of the likelihood of appearance in dynamic reference corpus, TF-IDF, Named Entities weight	-	English	Tweets	Topic recall. Keyword precision. keyword recall (Manually)	General Topics	Require fixed number of terms for processing

Reference	Technique	General Features	Twitter Features	Dataset Language	Dataset Type	Evaluation Method	Detected Content	Drawbacks
(Alkhamees & Fasli, 2016)	Frequent Pattern Mining using FP-Growth	Combination of words of size N	-	English	Tweets	-	General Topics	Produces generic events details. Thus the approach will treat similar events as one
(Doulamis et al., 2016)	Multi-assignment graph partitioning algorithm over fuzzy time series signal	CWTF-ITWTF, WF-ITW, WCWTF-ITWCWTF, Fuzzy time series signal	# of followers, # of retweets	English	Tweets	Keyword recall, keyword precision, F-Measure (Manually)	General Topics	Depends heavily on Twitter enabled features, thus the approach performance cannot be determined
(Shao et al., 2017)	Dynamic Multivariate Graph analysis	Term Frequency	Users Followers, Geolocation	English	Tweets	False Positive Rate (FPR), True Positive Rate (TPR)	General Topics	Cannot detect global events
(Yılmaz & Hero, 2016)	Expectation-Maximization algorithm	Bag of Words	Hashtags, Geolocations	English	Tweets	-	General Topics	Cannot detect global events
(Ferracani et al., 2017)	Statistical and Spatial data analysis using DTW, Crest Detection and DBSCAN	-	Users, Geolocations	English	Tweets	-	General Topics	Requires the longitude and latitude of user location, which are not always present

Reference	Technique	General Features	Twitter Features	Dataset Language	Dataset Type	Evaluation Method	Detected Content	Drawbacks
(Katragadda et al., 2017)	Graph analysis	Combination of words of size 2, term frequencies	-	English	Tweets	F-Measure	General Topics	Using combination of two words fail to detect similar events
(Alsaedi & Burnap, 2015)	Naïve Bayes classifier, Online Clustering	TF-IDF for classifying and clustering, Term Frequency, Topical Features, Spatial Features list of event keywords	Retweets ratio, Mentions ratio, Hashtag ratio	Arabic	Tweets	Average precision (Manually)	Events	Bag of words is used for the classifier training, thus the range of events to be detected is limited.
(Mohammad & Qawasmeh, 2016)	Rule-based	Part of Speech tags, Named Entities	-	Arabic	Tweets	Accuracy	Events	Cannot detect significant events
(Abuleil, 2007)	Based on List of Event Keywords	Words	-	Arabic	Full Text Articles	-	Events	Require maintaining a huge list of event keywords, Require well-written text in Standard Arabic

Reference	Technique	General Features	Twitter Features	Dataset Language	Dataset Type	Evaluation Method	Detected Content	Drawbacks
(Aliane et al., 2013)	Unsupervised Segmentation Algorithm, Rule-based	Verb Words	-	Arabic	Full Text Articles	-	Verbal Events	Identifying only verbs as event will make detected event very generic, Require well-written text in Standard Arabic
(Baradaran & Minaei-Bidgoli, 2015)	Support Vector Machine	Word Stem, Word's Part of Speech Category	-	Arabic	Full Text Articles	-	Death Events	Detect death events only, Require well-written text in Standard Arabic
(Baradaran & Minaei-Bidgoli, 2015)	Based on list of verbal event triggers	Words	-	Arabic	Full Text Articles	-	Verbal	Depends on a list of event keywords, thus it is not feasible to be used in unpredictable environment such as microblogs

Chapter 3
Theoretical and Technical Foundation

Chapter 3

Theoretical and Technical Foundation

This chapter gives an overview of the Twitter environment and the common terms used. Also a brief overview about frequent pattern mining algorithm is introduced.

3.1 Twitter Environment

Twitter is a microblogging services on which users interact with special messages called “tweets” Twitter (2017). It introduces its own expressions to the social networking services as follows:

- **User:** A Twitter user is a person or an organization that post content about specific or generic topics.
- **Tweet**: The name of the messages posted by users. It is limited to 280 characters only. Also it can contains other media such as images and short length videos.
- **Followers**: User followers are a group of users that receive tweets posted by that user. It is the mean of forming a networking of users in Twitter.
- **Follow**: It is the action of forming a friendship and following a user to receive his/her generated tweets.
- **Unfollow**: It is the action of breaking the friendship with users to avoid receiving his/her generated tweets.
- **Mute/Unmute**: It is the action of avoiding receiving tweets with keeping the friendship. Unmute is an action in which user resumes receiving tweets from the muted user.
- **Hashtags**: a special sequence of words that start with a hash (#) symbol. They are a type of metadata generated by users that allows other users to easily find tweets with similar content and theme.
- **@Replies**: users can interact with other users by replying to existing tweets.

- **Retweets**: In Twitter, reposting existing tweets is called retweet. When users retweet content it shows up to their followers.

3.2 Twitter API

Twitter provides a wide set of Application Programming Interfaces (API's) that can be used to interact with Twitter's platform.

3.2.1 Twitter Stream API

A Restful API that utilizes streaming HTTP protocol. A client App send a connection request to the API and a pipeline is opened to deliver tweets as they occur. Different types of messages are delivered through the pipeline, such as JSON-encoded activities, system message, and blank lines. All message are delimited with carriage-return (\r\n).

3.2.2 Tweets API

It provides a variety of APIs that handle tweets directly. The most important API is the lookup endpoint. A typical call uses the link *https://api.twitter.com/1.1/statuses/lookup.json?id=tweetID* to retrieve the full content of a tweet using its ID. In our work, we used this API to retrieve the actual content of the Twitter-based dataset.

To be able to use Twitter API, user need to be authenticated. Twitter authentication is based on the widely used protocol OAuth.

3.2.3 OAuth

It is an open standard used for access delegation. This means that, a user can grant access to all or a subset of his/her owned resources to a website or an application without giving them the password.

Twitter uses OAuth to provide authorized access to its API. It provides to authentication models:

- **User Authentication**: It allows an app to act on behalf of the user as the user himself.
- **Application-only authentication**: In which an application make API request on its own behalf

3.3 Frequent Pattern Mining (FPM)

FPM techniques focuses on finding relationships among items in a database to discover rules. "*Given a database D with transactions T1 … Tn, determine all patterns P that are present in at least a fraction s of the transactions*" (Aggarwal & Han, 2014). The fraction s is a threshold value also called minimum support. Popular FPM techniques are Aprior and FP-Growth algorithms. These techniques require a fixed value for the minimum support. To illustrate how FPM works, suppose we have a set of database transaction as shown in **Table 3.1.** We assume a pattern is frequent if it appears in 3 transactions, thus using Aprior algorithm we can determine the frequent pattern as follows:

1. Scan the database and count each item separately to achieve the result shown in **Table 3.2**.
2. Remove items that fall under the minimum support. In this scenario all items are above the minimum support, thus all items are frequent.
3. Generate a list of all pairs of the frequent items and count the number of transactions that these patterns appeared in. The result shown in **Table 3.3**. Shaded items count falls under the minimum support.
4. Remove items that fall under the minimum support.
5. Generate a list of triplets of the frequent items and count the number of transactions that these patterns appeared in. The result shown in **Table 3.4**. Shaded items count falls under the minimum support.
6. At this point the algorithm is stopped, because we cannot produce any combination of items that could be frequent. Also frequent pattern are those shown in **Table 3.3**.

Table 3.1: A set of Transactions

Transaction	Item sets
T1	[1,2,3,4]
T2	[1,2,4]
T3	[1,2]
T4	[2,3,4]

T5	[2,3]
T6	[3,4]
T7	[2,4]

Table 3.2: A list of item and their appearance count

Item	Count
1	3
2	6
3	4
4	5

With frequent pattern mining all items in the resulted pattern co-occur together in the same transaction.

Table 3.3: A list of combined items and their appearance count

Item	Count
1,2	3
1,3	**1**
1,4	**2**
2,3	3
2,4	4
3,4	3

Table 3.4: A list of triplets containing combination of items.

Item	Count
1,2,3	**1**
1,2,4	**2**
2,3,4	**2**

3.4 Soft Frequent Pattern Mining (SFPM)

It is a novel frequent pattern mining technique developed by Petkos et al. (2014). It is a less strict version of FPM, in which, we find frequent patterns without the condition that all items in the pattern co-occur together frequently. This means subsets of the pattern items may not occur together. SFPM is used in event detection to tackle

the limitation of using pairs for co-occurrence pattern examination, also to tackle the problem of strict behavior of FPM that requires all items to be frequent and co-occur together. The algorithm consists of three stages:

- **Term selection:** a set of k terms are selected from the dataset. The selection process depends on the term value of likelihood of appearance using a reference corpus.
- **Co-occurrence-vector formation:** a set of terms S is maintained and a corresponding vector Ds is created that exposes the features of S. a term t with vector Dt is added to S by comparing the similarity between Ds and Dt. After the inclusion of the term t, Ds values are updated. Both Ds and Dt are of length n, where n the number of tweets. Ds(i) denotes how many of the terms in S co-occur in the i^{th} tweet, and Dt(i) = 1 if the term t co-occur in the i^{th} tweet.
- **Post-processing:** in this step duplicate pattern are removed

To illustrate how SFPM works, suppose we have the following vectors, where Ds represent the current topic and D1 and D2 represents two different terms. Ds contains one term that co-occur in the first tweet, and two terms co-occur in the third tweet. D1 co-occur in the first tweet and the third tweet. D2 co-occur in the second tweet.

Ds	1	0	2	0	0	0	0	0	0
D1	1	0	1	0	0	0	0	0	0
D2	0	1	0	0	0	0	0	0	0

Figure 3.1: An example of vector Ds along with two terms vectors D1 and D2

As we can see, D1 and Ds have high similarity as both co-occur in the same tweets, thus D1 is merged with S and Ds values are updated using vector summation. The resulted vector of Ds will be as follows

Ds	2	0	3	0	0	0	0	0	0

Figure 3.2: The resulted vector of Ds after inclusion D1

We can interpret the result and the values of Ds as follows:

- The term of D1 co-occur frequently with a term in S in the first tweet.
- The term of D1 co-occur frequently with two terms in S in the third tweet.
- In S, two terms co-occur frequently on the first tweet, and three terms co-occur frequently in the third tweet, but these terms do not co-occur together at the same time.

3.5 Summary

In this chapter, we have presented a theoretical foundation for this research. We discussed Twitter environment and its characteristics. Also we introduced Twitter API and its usage. We also presented the Frequent Pattern Mining (FPM) and explained how it works and showed how strict it is. Finally we discussed the Soft Frequent Pattern Mining and showed its less strict behavior.

Chapter 4
Approach and Methodology

Chapter 4

Approach and Methodology

In this chapter we introduce our approach of event detection. We first describe the dataset used in the development of the approach. Then we explain and discuss the various steps included.

4.1 Overview

Our approach for event detection consists of the following steps: First, Data collection in which we acquire data and divide it into subsets called windows. A window is constrained by a time period. Second, a sequence of data preprocessing steps that include tokenization to identify keywords, a set of cleaning steps to remove noisy characteristics, normalization and stemming step to enhance word similarity, Part-of-Speech (POS) tags extraction step, and stop words removal to remove unnecessary keywords. Third, we extract event triggers by applying a set of pre-defined rules in the extracted Part-of-Speech tags. The top event triggers are extracted. Finally, The Soft Frequent Pattern Mining algorithm is applied on the top event triggers to detect events. Detected events are outputted as xml file. **Figure 4.1** shows these steps.

4.2 Data Collection

At this stage we collect the data that need to be analyzed. Usually, tweets are collected from Twitter stream. But in our case, an existing dataset that support our task is used. This is due to the following reasons:

- To evaluate our approach we cannot use Twitter stream directly because we do not know about occurring events or what tweets belong to that event.
- To create a baseline for comparison with other approaches.
- Collecting data manually will take too much time due Twitter API usage-time constraints. Also tweets need to be processed, annotated and assigned to an existing event at the time of its emergence.
- Assigning tweet to an event is subjective to the annotator, thus a well-established mechanism is needed for this task.

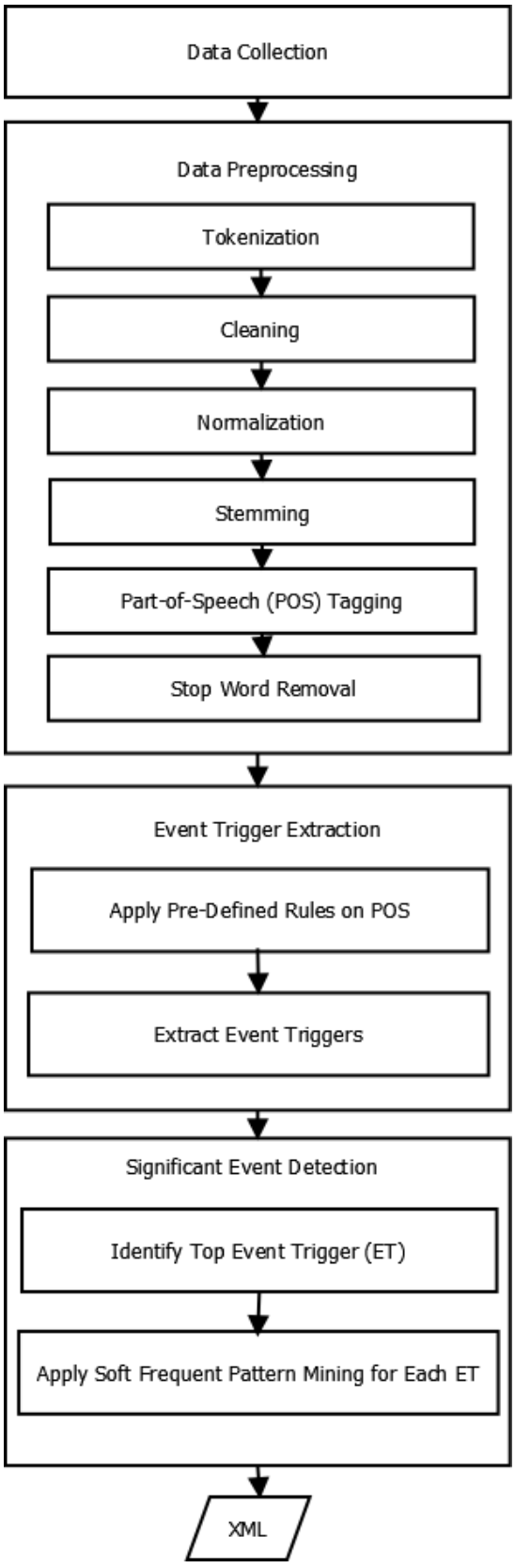


Figure 4.1:The different steps used in our approach.

For the best of our knowledge Evetar Almerekhi, Hasanain, and Elsayed (2016) is the only dataset that supports the Arabic event detection task. It covers different events and it contains labeled tweets. Also three widely known event detection implementations are performed over the dataset, thus we can use the results as a reference for evaluating our approach. **Table 4.1** show the distribution of event categories covered in the dataset.

Due to Twitter restriction on the tweet-based datasets, a published dataset must include the tweet ID -not the actual content. Users of the dataset should crawl the IDs and fetch the content. **Figure 4.2** shows a snapshot of the raw data of Evetar, where the first column represents a serial number, and the second column represents the tweet's id. **Figure 4.3** shows a fetched version of the dataset, where the first column represents the tweet's id, the second column shows the actual content of the tweet, the third column shows the creator of the content, and the last column represents the date and time of the tweet creation.

Table 4.1: Categories of events covered in Evetar dataset

Event Category	Number of Events Covered
Armed Conflicts & Attacks	45
Business & Economy	1
International Relations	3
Disasters & Accidents	3
Sports	5
Arts & Culture	2
Law & Crime	2
Politics & Elections	5

#	Tweet ID
1	549670988703862787
2	549670988774780928
3	549670988783169537
4	549670989856903169
5	549670992901971969
6	549670992943931392
7	549670993120493568
8	549670993514737665
9	549670993590226945
10	549670993715679232
11	549670996421410816
12	549670996903751680
13	549670997603782656
14	549670997612179456
15	549670998002241538
16	549670998216151040
17	549671000561156096
18	549671000946647040
19	549671001374486528
20	549671002041360384
21	549671004910268416
22	549671005157724160
23	549671006038540288
24	549671006194118656
25	549671006197923842
26	549671006567026689
27	549671009184272384
28	549671009318494208
29	549671009427529729
30	549671009557549057
31	549671009612480512

Figure 4.2: a snapshot of raw data from Evetar dataset

id	content	user	real_time
549694154486599680	RT @Holydefince: الحكم بالاعدام لشخصين ومؤيد لثالث في التفجير المزعوم في الدير http://t.co/e03e2Jas2L #Bahrain #Alwefaq #U# التورة_مستمرة...	silentwaves2	2014-12-29 10:31:13.000
549711593487888387	@helale9999 عشآن أعطينك وحده صميم صرت ترهي أعذار ...حقق العالمية و أرجع كلمني كيّ الأياب الانتحاري	AbdallhAoao1301	2014-12-29 11:40:31.000
549720880717508608	البحرين: ضبط مطلوبين متورطين في التفجير بالعكر الشرقي بقية الموضوع اضغط هنا http://t.co/t4A5bNrqyh	majjih	2014-12-30 12:17:25.000
549780004230283264	الطفل الانتحاري الذي سلم نفسه لحظات قبل التفجير قرب حسينيه في الباع هو سوري وليس عراقي "بيوبورك تايمز" صحيفة... http://t.co/PH1a97rvtb	basharkafy	2014-12-30 04:12:22.000
549828916488052736	RT @AlhujjaTv: الداخلية تكشف عن الحصيلة النهائية لشهداء وجرحى التفجير الانتحاري بقضاء التاجي شمالي بغداد -... http://t.co/bTL52rYElc	habd35	2014-12-30 07:26:43.000
549846940284764161	@AiooMah وش صار بالإياب الانتحاري هههههههههههههههههه	saudAa3A	2014-12-30 08:38:20.000
549879792745476096	خاص حلب الان حتى الان معلومات عن 4 شهداء من الجيش العربي السوري ومفقودين 2 تحت الانقاض نتيجة التفجير الارهابي	aleppo_now	2014-12-30 10:48:53.000
549880138033135617	التفجير الارهابي حدث تحت مبنى المحكمة الشرعية في محيط السبع بحرات	aleppo_now	2014-12-30 10:50:15.000
549885694814863360	حلب : الحمدلله جريح واحد ولا شهداء في صفوف الجيش العربي السوري حصيلة التفجير تحت الارض في منطقة السبع بحرات#	BtwlatAljyshAls	2014-12-30 11:12:20.000
549887667492503554	شهداء و 4 جرحى بصفوف الجيش السوري نتيجة التفجير الذي ضرب حلب القديمة 3	syriamother	2014-12-30 11:20:11.000
549887952026107904	عـــاجـــل /3/ شهداء و /4/ جرحى بصفوف الجيش السوري نتيجة التفجير الذي ضرب #حلب القديمة	7orasBWAB	2014-12-30 11:21:18.000
549889705936826368	خاص حلب الان التفجير الارهابي حدث تحت مبنى المحكمة الشرعية في محيط السبع بحرات http://t.co/UsVkEQmcaT #حلب	aleppo_news1	2014-12-30 11:28:17.000
549899268161814528	مجلس النواب يطمئن الشعب الليبي بأن أعضاء المجلس بخير بعد التفجير الانتحاري الذي استهدف مقره في فندق السلام بطبرق	Elmessallati	2014-12-30 12:06:16.000
549918122573324288	التفجير الذي أستهدف فندق دار السلام في مدينة طبرق صباح اليوم نُفذ عن طريق عملية إنتحارية بسيارة مليئة بالمتفجرات ,,	Libyan_News	2014-12-30 01:21:12.000
549920202193121280	التفجير اللي صار قدام مجلس النواب ~ سوق جماعة تحن نبوا مزلسس عسكري بس ,,	ragababrik22	2014-12-30 01:29:27.000
549925449204527104	Malek Alshrif كلب حشاكم من التفجير الانتحاري في طبرق -_ لازم ليون وبعنة الامم المتحدة تستنكر الموضوع كيف إصابة #... http://t.co/OPMjgI0HM0	wld_benghazi	2014-12-30 01:50:18.000
549928820829347842	لتحليل خبر التفجير أمام فندق السلام بطبرق لابد من استدعاء "الخبير العسكرى والأستراتيجى" محمد النعاس ..!	Baz_Ali_	2014-12-30 02:03:42.000
549929257988677634	RT @AlHadath: #ليبيا – نائب في البرلمان: تنظيم داعش وراء التفجير الذي طال مبنى البرلمان	AHAlosaimi	2014-12-30 02:05:26.000
549929891744219136	#حقيقة منشور لأحد الاصدقاء التشديد الامنـى عندنا ديما بعد التفجير مش قبله	hasnfitory	2014-12-30 02:07:58.000
549930037353660416	RT @fadelullahbujw2: التشديد الامنـى عندنا ديما بعد التفجير مش قبله	SAMIAHMED90...	2014-12-30 02:08:32.000
549931467695222786	#طبرق# ليبيا.... صورة من داخل فندق السلام وأثار التفجير فيه في مدينة http://t.co/IAHEPbkCBO	shbabt2014	2014-12-30 02:14:13.000
549934172413116418	#ليبيا #دار_السلام بخصوص التفجير الانتحاري لا توجد أضرار بشرية الأضرار طالت الفندق و كلب على الطريق في #طبرق	sanad_sh9	2014-12-30 02:24:58.000
549943345670139904	#قسورة حلب : الحمدلله جريح واحد ولا شهداء في صفوف الجيش العربي السوري حصيلة التفجير تحت الارض في منطقة السبع بحرات	kmhamad1	2014-12-30 03:01:25.000

Figure 4.3: a snapshot of fetched data from Evertar raw dataset

4.3 Data Preprocessing

This is a preliminary step required for preparing the dataset. We perform tokenization on every tweet. Then a cleaning and feature reduction step is performed. Then the resulted tokens are normalized. After that tokens are stemmed using a light Arabic stemmer. Part-of-Speech (POS) tagging is performed on every processed tweet. Lastly, stop words are removed from the dataset.

4.3.1 Tokenization

An important step to process textual data. It splits a text into whole-words, symbols, or other elements that has a meaning for the next steps. In this approach a unigram strategy is used where a token contains only one word.

4.3.2 Cleaning

Due to the nature of Twitter, tweets may contain noisy data. Thus, we need to remove such data to reduce the feature space for more accurate event detection. Our approach focuses only on the tweet text, thus twitter based features such as hash tags are normalized to normal text, and tweet handles or mentions are removed from the content. We perform the following:

- **Remove Latin Alphabets and Special character**: as we focus only on Arabic text, Latin alphabets are removed from tweets. Also special characters such as "#", "@", "(", ")", "<", ">", etc. are also removed.
- **Remove Emoticon**: an emoticon is a sequence of characters that represents user emotion e.g. ": D", ":)", etc. They are frequently used in Twitter, thus they may effect the accuracy of the detection process.
- **Remove URLs**: tweets may contain URLs. They are not required in our approach.
- **Remove Hash Tags**: a hash tag begins with the hashtag symbol "#" followed by a connected sequence of characters. We remove the hashtag symbol and keep the followed sequence, for example, the token "#حرب" will become "حرب". Sometimes the hashtag may contain more than one word, usually tweets authors use the underscore symbol for separation. With this case we

remove the hashtag symbol and replace the underscore symbol with a whitespace, for example, "#حرب_على_غزة" will become "حرب على غزة".

- **Tweets Filtering**: tweets that contains less than three tokens are dropped from the dataset.

4.3.3 Normalization

Due to the nature of Arabic language, similar words may appear differently as tweets authors can have different preferences. For example the word إحسان is written with Hamza, but also it can be written احسان without Hamza. Thus characters that can be written differently need to be converted to a standard format. A similar problem in the Arabic words is the stretch character known as Tatweel. Tatweel character are removed from tweets e.g. العثـــور will become العثور. Repeated characters can make a word appear differently in the dataset. These are also removed e.g. العثوووور will become العثور.

4.3.4 Stemming

Stemming is the process of reducing different forms of word to one form called stem or root Larkey, Ballesteros, and Connell (2002). Different stemming techniques for Arabic language exist. Root Stemmers extract the words root for example the word ذهبوا will become ذهب. A popular algorithm of this type called Khoja stemmer Taghva, Elkhoury, and Coombs (2005). Another technique called light stemming. Light stemming removes common prefixes and suffixes from words without converting them to their root for example the word كالعبيد will become عبيد. In our approach we will use the light stemmer because converting a word to its root may change its meaning in the context it was used in. For example, when a suicide event occur the word انتحار will be introduced, thus if we convert it to its root it will become نحر which has different meaning.

4.3.5 Part of Speech (POS) Tagging

Part of Speech tagging is the process of identifying the category of each word in the sentence based on its definition and context i.e. Noun, Verb, Adverb, etc. It is also called word-category disambiguation as it can disambiguate same words that could have different meaning in different contexts.

This step is very crucial in our approach. POS tags will be analyzed in further step to extract event triggers.

4.3.6 Stop Word Removal

Stop words are commonly used words in a context. An Example of such words is أنتم، نحن، هو، أنت، إذا etc. In social media stop words will appear frequently which if they are not handled well they will be treated as an event. Eliminating stop words from the dataset will increase the event detection process.

4.4 Event Triggers

An event trigger is a term or a group of terms that represent the event itself. In our approach we use event triggers as an indicators for the occurrence of an event in a tweet. Also it represents the important words in the text. It help us shortening the mining process and extract real events instead of popular topics. A set of pre-defined rules are used to extract event triggers and then frequently occurring event triggers are extracted. In the next subsections we explain the process deeply.

4.4.1 Apply Pre-Defined Rules on POS

The rules used in our approach are based on the Arabic Annotation Guidelines for Events Consortium (2005). These rules use the POS tags of a sentence to identify event triggers as follows:

1. If a tweet contains a verb phrase (VP) tag then we check the phrase for the following:

Rule.1.1. If it contains a Verb in base form (VB), Verb in past tense form (VBD), Verb in non-3rd person singular present form (VBP) or Verb in past participle form (VBN) tag followed by a Noun (NN) tag then we consider both tags as an event trigger. This rule is illustrated in **Table 4.3**.

Rule.1.2. If it contains (VB), (VBD), (VBP) or (VBN) tag followed by Adjective (JJ) tag then we consider both tags as an event trigger. This rule is illustrated in **Table 4.4**.

Rule.1.3. If the above rules does not apply then we consider (VB), (VBD), (VBP) or (VBN) tag as the event trigger.

2. If the tweet contains a Noun Phrase (NP) tag then we check the phrase for the following:

Rule 2.1 If it contains Noun (NN) tag followed by (NN) or Singular Proper Noun (NNP) tag then we consider both tags as an event trigger. This rule is illustrated in **Table 4.5**.

Rule 2.2 If it contains a Noun with (NN) or (NNP) tag followed by a Verb with (VB), (VBD), (VBP) or (VBN) tag then we consider both tags as an event trigger. This rule is illustrated in **Table 4.6**.

Table 4.2: List of Part-of-Speech tags used in event triggers extraction rules

Tag	Description
VP	Verb Phrase
VB	Verb in base form
VBD	Verb in past tense form
VBP	Verb in non-3rd person singular present form
VBN	Verb in non-3rd person singular present form
JJ	Adjective
NP	Noun Phrase
NN	Noun
NNP	Singular Proper Noun

Table 4.3: An Example of Rule 1.1

Tweet	قضت محكمة الجنايات ببراءة خبير هندسي في وزارة العدل
POS Tags	(ROOT (S (VP (**VBD قضت**) (NP (**NN محكمة**) (NP (DTNNS الجنايات) (DTJJ ببراءة))) (NP (NP (NN خبير) (JJ هندسي)) (PP (IN في) (NP (NN وزارة) (NP (DTNN العدل))))))))
Rule	Rule 1.1
Event Trigger	قضت محكمة

Table 4.4: An Example of Rule 1.2

Tweet	البحرين: ضبط مطلوبين متورطين في التفجير بالعكر الشرقي بقية الموضوع اضغط هنا
POS Tags	(ROOT (S (NP (DTNNP البحرين)) (VP (**VBD ضبط**) (NP (**JJ مطلوبين**)) (S (VP (VN متورطين) (PP (IN في) (NP (NP (DTNN التفجير) (DTJJ بالعكر) (DTJJ الشرقي)) (NP (NN بقية) (NP (DTNN الموضوع) (DTJJ اضغط))))) (ADVP (RB هنا)))))))

Rule	Rule 1.2
Event Trigger	ضبط مطلوبين

Table 4.5: An Example of Rule 2.1

Tweet	قتيلان على الأقل في اطلاق نار خلال عملية احتجاز رهائن شرق باريس
POS Tags	(ROOT (S (VP (VBD قتيلان) (PP (IN على) (NP (NP (DTNN الأقل)) (PP (IN في) (NP (**NN اطلاق**) (NP (**NN نار**)))))) (NP (NN خلال) (NP (**NN عملية**) (NP (**NN احتجاز**) (NP (NP (NN رهائن)) (NP (NN شرق) (NP (NNP باريس)))))))))))
Rule	Rule 2.1
Event Trigger	إطلاق نار، عملية إحتجاز

Table 4.6: An Example of Rule 2.2

Tweet	ألمانيا تحذر من انقسام المجتمع عقب هجوم شارلي إبدو

POS tags	(ROOT (S (NP (**NNP ألمانيا**)) (VP (**VBP تحذر**) (PP (IN من) (NP (NN انقسام))) (NP (DTNN المجتمع)) (NP (NN عقب) (NP (NN هجوم) (JJ شارلي) (JJ إبدو))))))
Rule	Rule 2.2
Event Trigger	ألمانيا تحذر

4.4.2 Extraction of Event Triggers

In the previous step we identify all event triggers in the dataset. A List of the event triggers and their frequencies is maintained. As the approach focuses only on significant events, we assume high-frequency event triggers are the result of the surge use of similar writing pattern about specific event. Consequently, we remove event triggers that has small frequencies. Alkhamees and Fasli (2016) used threshold for selecting frequent terms pattern, however, in our work we used a threshold value to determine frequent event triggers. For simplicity, we used the average of all frequencies as the threshold value as used in Lin, Ren, and Fournier-Viger (2018).

4.5 Significant Event Detection

In this step we generate groups of similar event triggers, where each group represents an event. We use an adapted version of the soft frequent pattern mining algorithm Petkos et al. (2014) to cluster event triggers that co-occur frequently but not necessarily all the terms are frequent in the same document.

Originally, Petkos et al. (2014) extracts important words by comparing the words in dataset with a reference corpus. The problem with this solution is that maintaining a reference corpus require a lot of efforts, also new event's related-words that do not appear in the reference corpus will be considered not important, thus identifying

important words is very biased. However, in our work, we select important words by selecting event triggers using rules that depends on the part-of-speech tags.

4.5.1 Soft Frequent Pattern Mining (SFPM)

Soft Frequent Pattern Mining (SFPM) is derived from the concept of Frequent Pattern Mining (FPM). FPM is the process of finding frequent items in a set of transactions Aggarwal and Han (2014). An item is said to be frequent if its frequency is above a pre-defined threshold or support count. A frequent pattern is a set of items that co-occur together in the same transaction and their frequency is above a threshold. We consider the task of event detection as a frequent pattern mining problem. When an event occurs, a group of users will start tweeting about it using similar words pattern. **Figure 4.4** shows a list of tweets describing an event. Thus the pattern used for the event will have higher frequency in the dataset when more users are tweeting about it. One problem of using FPM is that all terms in the selected frequent pattern must be frequent in the same transaction. To illustrate this problem in Twitter context: Suppose we have a dataset of 3 tweets containing an event as shown in **Table 4.7**.

If we apply FPM algorithm in this dataset it will produce [إستشهاد، مواطن], as the frequent pattern because it appeared in the three tweets. We lost important features such as [محمد، رفح، محمود، أحمد، سلامة] which can indicate different events because these features are not frequent. Thus FPM will produce a generic description of the event.

To tackle this problem we use the Soft Frequent Pattern Mining (SFPM). It is a technique that identify frequent pattern without the condition that all words are frequent in the same tweet. In the example above if we apply the SFPM then it will produce [رفح، محمود، محمد، مواطن، إستشهاد], [سلامة، أحمد، مواطن، إستشهاد] as a frequent pattern. Next the algorithm is explained more deeply.

Figure 4.4: A list of tweets describing the event of 16 journalist who was killed by Israeli

Table 4.7: An example of different events with similar features

Tweet #	Content
1	إستشهاد المواطن محمد محمود من رفح
2	إستشهاد مواطن في رفح
3	إستشهاد المواطن أحمد سلامة

4.5.2 Adapted SFPM

We assume that an event is represented by a set of event triggers not just only one. This is due to the difference in writing style between users as shown in **Table 4.8** which make the POS tagger produce different tags, thus different event triggers for the same event.

Table 4.8: An example of two tweets representing the same event with different event triggers

Tweet 1	
Tweet Text	**أصبحت ليتوانيا** اليوم الدولة الـ 19 التي تنضم لمنطقة اليورو، بعد 15 عاما من إطلاق أوروبا العملة الموحدة لأول مرة#. الرايـة# أوروبا
Processed	**اصبحت ليتوانيا** دول ال تي تنضم لمنطق اطلاق اوروبا عمل موحد لاول مر را اوروبا

POS	VBP/تنضم NN/تي NN/ال NN/دول **NNP/ليتوانيا VBD/اصبحت** NN/مر JJ/لاول JJ/موحد NN/عمل NNP/اوروبا VBG/اطلاق NN/لمنطق NNP/اوروبا NNP/را
Rule	**1.1**
Tweet 2	
Tweet Text	#ليتوانيا **تصبح# العضو** التاسع عشر في# منطقة_اليورو مع أول أيام عام2015
Processed	ليتوانيا **تصبح عضو** منطقه اليورو
POS	DTNNP/اليورو NN/منطقه **NN/عضو VBP/تصبح** NNP/ليتوانيا
Rule	**1.1**

In addition to that, the POS tagger may incorrectly tag a set of words that hold true to the rules in section 4.4.1. **Table 4.9** shows an example of this scenario. In our work, this incorrectly captured features will still have high frequency in case of events, thus it will not affect the detection process.

Table 4.9: An example of two tweets that represent an event with incorrectly extracted event trigger

Tweet 1	
Tweet Text	#عاجل | **قتلى وجرحى** في تفجير انتحاري استهدف المركز الثقافي في# إب وسط# اليمن
Processed	عاجل **قتل جرح** تفجير انتحار استهدف مركز ثقاف اب وسط يمن
POS	VBD/استهدف NN/انتحار NN/تفجير **NN/جرح VBN/**قتل JJ/عاجل NNP/يمن NN/وسط NNP/اب NNP/ثقاف NN/مركز
Rule	**1.1**
Tweet 2	
Tweet Text	**#اليمن# نيوز_يمن-** الداخلية :ارتفاع حصيلة ضحايا التفجير الإنتحاري بإب إلى 23ف
Processed	**يمن نيوزيمن** داخل ارتفاع حصيل ضحايا تفجير انتحار باب ال
POS	NN/ضحايا NN/حصيل NN/ارتفاع NN/داخل **NN/نيوزيمن VBP/يمن** DT/ال NN/باب NN/انتحار NN/تفجير
Rule	**1.1**

To formulate the algorithm, suppose we have a set of tweets T of size N, and a K number of event triggers (ET) where all tweets in T contain at least one event trigger. Our task is to group similar event triggers together and retrieve their common tweets as the detected event. Thus the objective of the algorithm is to produce a set of grouped event triggers.

Initially, every event trigger ET is treated as a single event. To be able to merge ETs, a numerical representation is needed. A vector D_S of size N is calculated for each ET where $D_s(n)=1$ when ET appears in the n^{th} tweet and $D_s(n)=0$ otherwise. The popular Cosine similarity measure is used for comparison, where two event triggers are merged when their calculated similarity is above a threshold value. After the merge, a vector summation is performed on both Ds vectors, and the newly created vector is assigned to the newly created group. **Figure 4.5** show pseudocode of the implementation of the vector sum process.

Algorithm: Merge(ET1, ET2)
Input: two event triggers **Output:** a merged version of two event triggers 1 newET = **[]** 2 newET.EventWords.Add(ET1.EventWords) 3 newET.EventWords.Add(ET2.EventWords) 4 newET.Ds = Integer Array of Size ET1.Ds.Count 5 **FOR** i=0 **to** ET1.Ds.Count 6 newET.Ds[i] = ET1.Ds[i] + ET2.Ds[i] 7 **END FOR** 8 **RETURN** newET

Figure 4.5: A pseudocode of merging two vectors.

In a single full iteration, if no event trigger is merged, then the produced group of event triggers is treated as an event. The process is repeated for all event triggers that are not assigned to any group and the algorithm is terminated when there is no further merge. **Figure 4.6** shows pseudocode of the full implementation of the adapted soft frequent pattern mining.

Algorithm: Adapted Soft Frequent Pattern Mining (SFPM)

Input: List of Event Triggers (ET)
Output: Sets of Grouped Event Triggers

```
1    FOR i=1 to ET.Count
2        Calculate ET[i].Ds
3    END FOR
4    events = ET
5    isRepeat = TRUE
6    WHILE (isRepeat) Do
7        Temp = events
8        isAssigned = Integer Array of Size Temp.Count
9        newEventSet = Empty Object
10       isRepeat = FALSE
11       FOR i=1 to Temp.Count - 1
12           IF isAssigned [i] = 1 THEN
13               Skip
14           END IF
15           FOR j=i+1 to Temp.Count
16               IF Similarity(Temp[i].DS, Temp[j].DS) >θ(|Temp[i]. DS|) THEN
17                   isAssigned[j] = 1
18                   Temp[i] = Merge(Temp[i], Temp[j])
19                   isRepeat = TRUE
20               END IF
21           END FOR
22           newEvents.ADD(Temp[i])
23       END FOR
24       events = newEvents
25   END WHILE
```

Figure 4.6: Pseudocode of the Adapted Soft Frequent Pattern Mining Algorithm

4.5.3 Similarity Threshold

In our work we use the following sigmoid function to create an adapted similarity threshold value. The sigmoid function is a function of S which is the set of event triggers that belong to one event. |S| is the number of event triggers in the set. Similar to Petkos et al. (2014) we choose b=5, and c=2 **Figure 4.7** shows the graph of the sigmoid function.

$$\theta(S) = 1 - \frac{1}{1 + e^{\frac{|S|-b}{c}}} \qquad (1)$$

One benefit from using a sigmoid function of |S| is that, when S contains low number of event triggers a low threshold value will be used, thus at this state it become more easily to accept similar event triggers. When the cluster grows the threshold

value will be increased, thus it become harder to accept new event triggers. The cluster will become saturated.

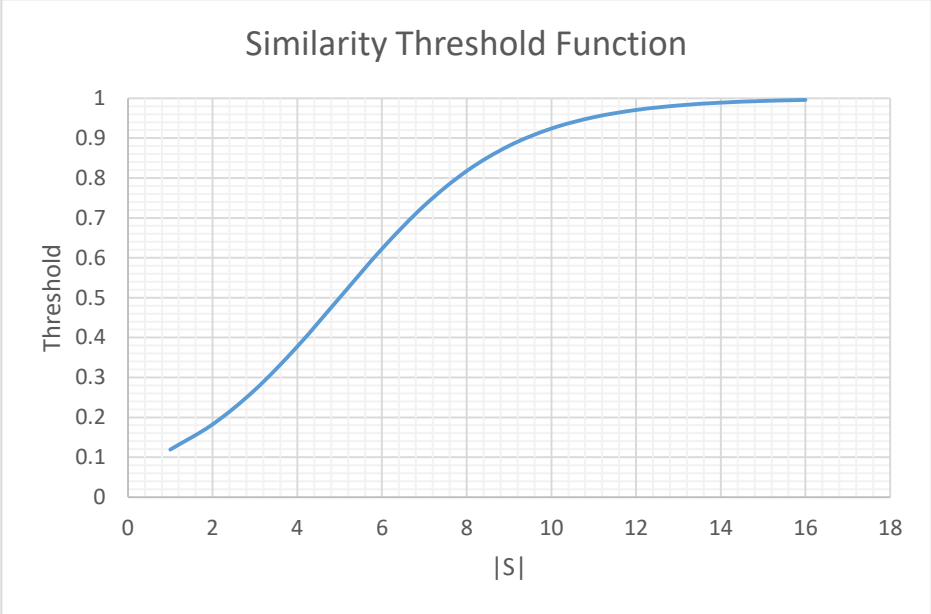


Figure 4.7: Graph of Similarity Threshold Function

4.6 Summary

In this chapter we introduced our approach for detecting Arabic significant events. We presented the various stages included in the approach starting from data collection. We discussed the dataset used to develop the approach, then we discussed the preprocessing steps which includes tokenization, cleaning, normalization, stemming, part-of-speech tagging, and stop word removal. We explained the step of event trigger extraction using predefined rules. We also discussed the detection of Arabic significant events using the Soft Frequent Pattern Mining (FPM) over the extracted top event triggers. Finally, we introduced the similarity function used in FPM.

Chapter 5
Implementation

Chapter 5

Implementation

This chapter describes the implementation of the proposed approach. We introduce the development process and libraries used to implement every step in the approach.

5.1 Dataset

To test the correctness our approach of event detection, we need to find a dataset for such task. Evetar Almerekhi et al. (2016) is an Arabic dataset targeted for the event detection task. It contains a total of 590,066,789 tweets and covers 66 significant events. It has been collected in a one-month period using Wikipedia's Current Events Portal at that time. An event is represented by a set of tweets that are relevant to that event within a time period surrounding the time of the event.

Published Twitter-based datasets are constrained to contain only tweets IDs –not the actual content. This is due to privacy concerns imposed by Twitter, as the tweet may become unavailable because it is deleted or its author constrained his profile access or due to other reasons. Thus, Twitter-based datasets users need to fetch the actual content from the IDs using Twitter statuses API. This process called tweet ID hydrating Twitter (2018), moreover, Twitter apply a rate limit on the number of requests made. This rate limit is 900 requests per 15 minutes. This make it very hard to fetch all the dataset, thus we choose to use a subset of the original dataset. Evetar's authors provide a sample of 134,069 tweets that covers the same time period and same events.

First we created a Twitter Application on Twitter development platform to generate an access token and an access token secret to be able to access twitter over OAuth. We then developed a small tool that iterates over the IDs dataset and fetch the corresponding content. We used a library called TweetSharp. It is a C# .NET library that simplifies the use of Twitter API. It has *TwitterServices* class which provides a method called *GetTweet* that accepts a tweet ID and returns a *TwitterStatus* object. *TwitterStatus* imitates the real JSON object returned from Twitter. **Table 5.1** shows some of the important fields returned by *TwitterStatus*.

Table 5.1: Important fields of the TwitterStatus object.

Field Name	Description
Id	Tweet ID
Author	The author of the tweet
CreatedDate	The creation date and time of the tweet
Text	The actual content of the tweet

After iterating over the dataset we was able to fetch only 59,732 tweets that covers 50 significant events, as most of the tweets were deleted or their authors accounts were suspended, or deleted. The Fetched tweets are stored in SQL Server Database. **Figure 5.1** shows all the events covered by the retrieved tweets and the number of tweets belong to them. A subset of 4200 tweets were used for testing and building the system.

5.2 Software Specification

As there is no well-known framework to test our approach we built a software that implements the many parts of the approach. Next we will show the tools and libraries used to build the software.

5.2.1 Visual Studio & C# .NET

Visual Studio is an integrated development environment (IDE) from Microsoft It is used to develop a wide variety of computer software such as desktop, web, and mobile applications. C# is an elegant and type-safe object-oriented language that enables developers to build a variety of robust applications that run on the .NET framework Microsoft (2017).

5.2.2 Stanford NLP .NET

The .NET version of Stanford NLP library. A natural language processing library built in java Manning et al. (2014). We used the Stanford Parser for the POS tagging task. We choose to use this library as it supports the Arabic language and it is widely used between researchers.

5.2.3 Lucene

It is an open source information retrieval library originally built in Java. It is mainly used for document indexing and searching. Documents are represented as a collection of fields. It provides a highly expressive search API that takes a search query

and returns a set of documents. It also provides text processing features such as Arabic Light Stemmer, etc. Apache (2014).

5.2.4 Microsoft SQL Server

A database management system developed by Microsoft. Its primary function is to store and retrieve as requested by other applications using SQL. SQL server database can exist in a separate server or on the same machine of the requester applications (Microsoft).

5.2.5 TweetSharp

TweetSharp is a Twitter API library built in C# .NET. It greatly simplifies the task of adding Twitter functionality to applications. It also simplifies the way we handle the JSON objects returned from Twitter and saves the efforts of parsing them manually TweetSharp (2017).

5.3 Framework Implementation

As we gave details about our proposed approach in Chapter 4 and details about the hardware and software specification used to implement it in Section 5.2. In this Section we put all this together. All software development was done using Visual Studio IDE.

5.3.1 Dataset Preprocessing

In this subtask we performed the following:

5.3.1.1 Tokenization

The first step in the preprocessing task is to segment text into tokens. This is done by the Stanford NLP library. At first the text is converted to a *StringReader* object. Then the *PTBTokenizer* class provided by the library accepts the tokenizer options through the method *factory* and returns a *TokenizerFactory* object. *TokenizerFactory* accepts the *StringReader* object and returns a list of tokens. We used the default configuration of the tokenizer which specifies a token as a sequence of characters surrounded by whitespace. **Table 5.2** shows an example of this process.

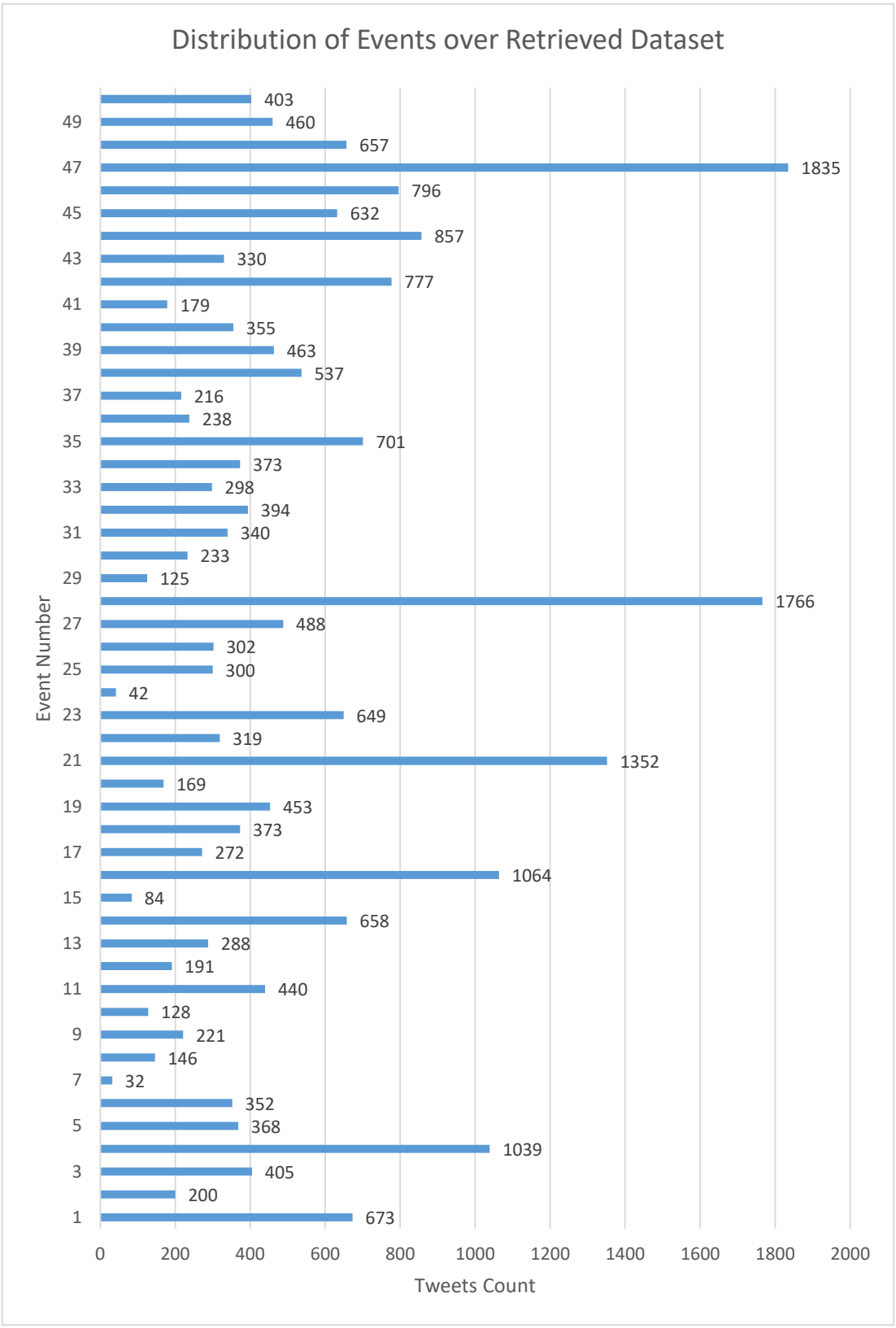


Figure 5.1: Bar graph shows events covered and the count of tweets belong to them

Table 5.2: An example of tokenization performed on a tweet.

الاعدام لعامل مطعم قتل زميله طعناً في "البيادر" أيدت محكمة التمييز الحكم الصادر عن محكمة الجنايات الكبرى والقاضي... http://t.co/H0txdjv3Kn
[الاعدام, لعامل, مطعم, قتل, زميله, طعناً, في, ", البيادر, ", أيدت, محكمة, التمييز, الحكم, الصادر, عن, محكمة, الجنايات, الكبرى, والقاضي, ..., http://t.co/H0txdjv3Kn]

5.3.1.2 Remove Latin Alphabets and Special Characters

As we are mainly focusing on the Arabic language and the text only of the tweet. We remove any Latin alphabets and special characters from the text. We approach this by using regular expression through the *Regex* class provided by the .NET Framework. The regular expression used is [\x00-\x7F]. With this we were able to remove Emoticons, URLs, and Hashtags. **Table 5.3** shows an example of this process.

Table 5.3: The resulting tweet after removing Latin Alphabets and special characters

الاعدام لعامل مطعم قتل زميله طعناً في "البيادر" أيدت محكمة التمييز الحكم الصادر عن محكمة الجنايات الكبرى والقاضي... http://t.co/H0txdjv3Kn
الاعدام لعامل مطعم قتل زميله طعنا البيادر أيدت محكمة التمييز الحكم الصادر محكمة الجنايات الكبرى والقاضي

5.3.1.3 Normalization

We created a class for normalization as follows: A token is normalized by converting the character of Alef with Hamza or Madda [أ – إ – آ] to Alef without Hamza or Madda [ا], also the characters Taa Marbota [ة] is converted to Haa [ه]. Tatweel character [-] is also removed and repeated characters are reduced to one character. **Table 5.4** shows an example of this process.

Table 5.4: The result of using normalization in Arabic text

الاعدام لعامل مطعم قتل زميله طعنا البيادر أيدت محكمة التمييز الحكم الصادر محكمة الجنايات الكبرى والقاضي
الاعدام لعامل مطعم قتل زميله طعنا البيادر ايدت محكمه التميز الحكم الصادر محكمه الجنايات الكبرى والقاضي

5.3.1.4 Stemming

To achieve higher similarity between Arabic words. We stem each token using Lucene *ArabicStemmer* which is a Light Arabic Stemmer. **Table 5.5** shows an example of this process.

Table 5.5 The result of using stemming in Arabic text.

الاعدام لعامل مطعم قتل زميله طعنا البيادر ايدت محكمة التمييز الحكم الصادر محكمه الجنايات الكبرى والقاضي
اعدام لعامل مطعم قتل زميل طعنا بيادر ايدت محكم تمييز حكم صادر محكم جنا كبر قاض

5.3.1.5 Part of Speech (POS) Tagging

We used Stanford NLP Parser to annotate text with its POS tags. First we load the Arabic trained model *arabicFactored.ser.gz* using the method *loadModel* from the *LexicalizedParser* class. The *LexicalizedParser* also provides *apply* method that takes a list of tokens and return a *Tree* object which contains POS tree of the inputted tokens. **Table 5.6** shows a processed tweet with the Stanford parser.

Table 5.6: The Part of Speech Tree of a tweet resulted from LexicalizedParser

اعدام لعامل مطعم قتل زميل طعنا بيادر ايدت محكم تمييز حكم صادر محكم جنا كبر قاض
(VBD كبر) (NNP جنا) (NNP محكم) (NNP صادر) (NN حكم) (NN تمييز) (NN محكم) (NN ايدت) (NN قاض)

5.3.1.6 Stop Words Removal & Tweets Filtering

We iterate over each token and check if it exists in the stop words list. The list contains 750 stop words. After this process we check if the remaining tokens number in a tweet if the remaining number less than three then the tweet is removed.

5.3.2 Event Extraction

The objective of this step is to find if a tweet has an event trigger which can help us in the process of finding the significant event. If a tweet does not have an event trigger then it is dropped from the dataset. As we mentioned in Section 4.4.1 we used Consortium (2005) rules for extracting event triggers. We approach this by implementing Algorithm 1. First we initialize an empty list ET which will eventually contain event trigger tokens. For each tweet we retrieve the resulted POS tree. The tree

is flattened using the method *taggedYield* which returns a list of *TaggedWord* object. **Table 5.7** shows the flattened version of POS tree.

Table 5.7: A flattened version of POS tags

مراسل العربية انفجار في محيط مسجد شمال فرنسا و لا إصابات
مراسل/NN, العربية/DTNN, انفجار/NN, محيط/NN, مسجد/NN, شمال/NN, فرنسا/NNP, إصابات/NN

We iterate over the list of TaggedWord and apply the rules showed in section 4.4.1. Every extracted event trigger is added to the list ET and the frequency is updated. An event trigger is represented by the class *EventToken* as shown in **Figure 5.2** where the Word attribute represents the initial event trigger word and Dt is the numerical representation of the event trigger. It is calculated and updated in the next step. Finally, the WordList attribute is the list of event triggers that are merged with the initial event trigger. As we mentioned earlier, a threshold is used to retrieve top event trigger. We use the value of the average frequencies as the threshold. **Figure 5.3** shows a snapshot of the output of this step as an XML file using the test dataset. Each tag contain Name that represent the event trigger word and frequency that represent its frequency. Event trigger and documents are indexed using a *Dictionary* object for fast retrieval of document.

```
class EventToken
{
    public string Word { set; get; }
    public List<string> WordList { set; get; }
    public int[] Dt { set; get; }

    public EventToken()
    {
        WordList = new List<string>();

    }
}
```

Figure 5.2: A snapshot of the class EventToken

```
<EventTrigger name="تفرض عقوب" frequency="91" />
<EventTrigger name="يوقع عل" frequency="89" />
<EventTrigger name="تسمح ببناء" frequency="71" />
<EventTrigger name="تصبح عضو" frequency="46" />
<EventTrigger name="قتل جرح" frequency="45" />
<EventTrigger name="قال لكل" frequency="45" />
<EventTrigger name="عل كوريا" frequency="39" />
<EventTrigger name="تخطف شابا" frequency="32" />
<EventTrigger name="يفوز بترشيح" frequency="30" />
<EventTrigger name="عل توقيع" frequency="28" />
<EventTrigger name="بوكو حرام" frequency="26" />
<EventTrigger name="يفرض تاشير" frequency="26" />
<EventTrigger name="يفرض مر" frequency="25" />
<EventTrigger name="غضب اسرائيل" frequency="24" />
<EventTrigger name="قنا هلال" frequency="22" />
<EventTrigger name="يعتزم تاسيس" frequency="21" />
<EventTrigger name="اصاب اشخاص" frequency="20" />
```

Figure 5.3: An XML snapshot of the outputted event trigger from the test dataset and their corresponding frequencies

5.3.3 Significant Event Detection

At this point top event triggers are extracted and stored in a list ET. For each event trigger we query the dictionary object to retrieve the list of documents associated with the event trigger. Every unique token is retrieved from the document list and added to the token list. For each event trigger, the Ds vector is calculated and then we apply the implementation of the adapted SFPM on the vectors. **Figure 5.4** shows a snap shot of the detected events. Every event is represented by an *Event* tag with a list of *Tweet* tags. For the *Event* tag, The *EventTriggers* attribute represents the list of merged event triggers. EventNo represent the order of the detection. The Count attribute indicates how many tweets belong to the event. For the *Tweet* tag, the user attribute represents the username of the user who posted it, id attribute represents the tweet id, the date attribute represents the date of the tweet, and finally the event attribute which represents the event to which the tweet belongs. This attribute is very important for the evaluation. A *Tweet* tag contains three other tags, the *Original* tag which represents the original text before the preprocessing, the *Text* tag which represents the processed tweet content before applying stop word removal and the *TextAfterSW* tag which represents the processed content after removing the stop words.

```
<Events>
 <Event
   EventTriggers="اوباما , امريك عقوب , بيونغ عل , عقوب يجيز , عقوب تفرض
      , اميرك عقوب , كوريا عل , امريكا ان , كورياالشمال عل , قرصن عل , مزيدا يقر
      , اوباما , امريك عقوب , متحد ولا , بيونغيانغ عل , بيونغ عل , عقوب يجيز
      , معاقب تبدا , امريكا ان , كورياالشمال عل , قرصن عل , مزيدا يقر"
   EventNo="1"
   Count="214">
   <Tweet user="MTVLebanonNews" date="02/01/2015 07:04:14 AM"
id="551091616459415552" event="E05">
     <Original>
" سوني "طالت التي عملية القرصنة بعد الشمالية كوريا على عقوبات تفرض واشنطن :أ.ف.ب
http://t.co/8DSZ9ogkwC
     </Original>
     <Text>
افب اشنطن تفرض عقوب عل كوريا شمال بعد عمل قرصن تي طالت سون
     </Text>
     <TextAfterSW>
افب اشنطن تفرض عقوب كوريا عمل قرصن تي طالت سون
     </TextAfterSW>
   </Tweet>
   <Tweet user="LatestNews_AR" date="02/01/2015 07:42:14 AM"
id="551101178998562817" event="E05" cluster="15">
     <Original>
 مصر واشنطن تفرض عقوبات مالية على كوريا الشمالية بعد عملية قرصنة“ سوني :”فرضت الولايات #
... http://t.co/5wErrEdMyL #Egyptم المتحدة عقوبات
     </Original>
     <Text>
مصر اشنطن تفرض عقوب مال عل كوريا شمال بعد عمل قرصن سون فرضت ولا متحد عقوب
     </Text>
     <TextAfterSW>
مصر اشنطن تفرض عقوب مال كوريا عمل قرصن سون فرضت متحد عقوب
     </TextAfterSW>
   </Tweet>
   <Tweet user="housam_abdrabo" date="02/01/2015 08:01:24 AM"
id="551106002884374528" event="E05" cluster="16">
     <Original>
 واشنطن تفرض عقوبات على كوريا الشمالية بعد عملية القرصنة التي طالت" سوني ."والقائمة تشمل 10
.  #sonyمسؤولين في النظام و 3منظمات وشركات
     </Original>
     <Text>
اشنطن تفرض عقوب عل كوريا شمال بعد عمل قرصن تي طالت سون قائم تشمل مسؤول في نظام منظم شرك
     </Text>
     <TextAfterSW>
اشنطن تفرض عقوب كوريا عمل قرصن تي طالت سون قائم تشمل مسؤول نظام منظم شرك
     </TextAfterSW>
   </Tweet>
 </Event>
</Events>
```

Figure 5.4: An XML snapshot of the detected events

5.4 Summary

In this chapter, we discussed several tools that aid our software development and implementation of our approach. We showed the process of extracting the actual tweets content from the internet using Twitter API and the specific tools for this task. We discussed the retrieved data. We also introduced the libraries used in data preprocessing and showed the actual output of each step. In addition to that, we showed the output of the event trigger extraction step. Finally, we showed the final output of the detected events.

Chapter 6
System Experiments & Evaluation

Chapter 6

System Experiments and Evaluation

In this chapter experiments and evaluation of our implemented approach for event detection is conducted. As the approach targets the Arabic Language we used the Evetar dataset for evaluation. For the best of our knowledge there is no well-known framework for evaluating event detection task. Evetar authors evaluated three popular off-the-shelf implementations of event detection Guille and Favre (2014), Weng and Lee (2011), Shamma, Kennedy, and Churchill (2011) with their dataset using the evaluation approach proposed by Petrovic (2013), thus we used the same evaluation method. Also we used the results of the evaluated approaches as a baseline for comparison.

6.1 Event Trigger Extraction

The Fetched Evetar dataset contains 59,732 tweets where 23,973 are labeled as events and their corresponding event is also identified. Even though the 35,759 tweets are not labeled, this does not mean they do not refer to an actual event or do not contain an event trigger. For example, the tweet in **Table 6.1** expresses an event of killing "يتبني قتل" but the tweet itself was not labeled in the dataset because it does not belong to any of the covered events. Also the tweet in **Table 6.2** contains an event trigger but the tweet does not express the occurrence of any event. An example of the output of this component is shown in **Figure 5.3**

Table 6.1: A tweet that describes an event but not labeled in the dataset

Tweet	تبنى تنظيم داعش امس :(داعش) **يتبنى قتل** مستشار عسكري إيراني في العراق" عاجل #السعودية# http://t.co/NSWrZTmNP2... قتل ضابط عسك"
Event Trigger	يتبني قتل
Rule	Rule 2.1

Table 6.2: A tweet that does not describe any event but it contains an event trigger

Tweet	RT @mesfhel: أمرٍ لا دخل لك فيه **تبحث خلف** لا.. !
Event Trigger	تبحث خلف
Rule	Rule 2.1

There is no well-established method to evaluate event trigger extraction, also the dataset does not provide any annotation for event triggers, and no clear knowledge about the number of event triggers is provided. Thus we cannot evaluate this component. We will focus on evaluating the overall output of our approach.

6.2 Significant Event Detection Evaluation

6.2.1 Measurements

In our experiments we used the same measurements and evaluation method introduced by Petrovic (2013). The following subsections describe these measurements.

6.2.1.1 Event Recall

Recall is a measurement used in text retrieval. It is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. A detected event is a cluster of tweets. A cluster is said to be *true positive* if its containing tweets cover any of the reference events. As a cluster may contain hundreds or thousands of tweets, it is very likely to find tweets that do not belong to the event. Thus Petrovic (2013) computes the proportion of tweets in the cluster that are part of a single reference event. If the proportion is greater than a threshold then it is true positive. We used the value of 0.5. Thus recall in event detection is calculated as follows:

$$recall = \frac{number\ of\ covered\ events}{number\ of\ reference\ events} \qquad (2)$$

Where reference events are the events covered in the dataset.

6.2.1.2 Event Precision

Precision is the fraction of relevant instances among the retrieved instances. Precision of event detection is calculated as follows:

$$precision = \frac{number\ of\ covered\ events}{number\ of\ detected\ events} \qquad (3)$$

6.2.1.3 F-Measure

It is the harmonic average of the precision and recall. It is used to determine the accuracy of classification problems.

$$f - measure = \frac{2 * precision * recall}{precision + recall} \quad (4)$$

6.2.2 Experiments Setup

We conducted our experiments over Evetar to compare our results with other approaches of event detection. In reality, Twitter data comes as a stream of tweets in chronological order, thus it is not feasible to apply the event detection approach on the whole dataset. To mimic this behavior we split the tweets into chronologically ordered chunks/subsets. The number of subsets depends on the time interval applied. We choose two time intervals, a one-day time interval, and a 6-hour time interval. Unlike other approaches, the reason we choose big time intervals is the small amount of tweets retrieved from the dataset IDs. **Table 6.3** shows the resulted subsets number for each time interval. The overall Precision, Recall and F-Measure for each time interval is the average value of each measure for every subset.

Table 6.3: Time intervals and their corresponding subsets

Time Interval	Number of Subsets
One Day	22 Subsets
6 Hours	11 Subsets

6.2.2.1 6-Hours Time Interval Experiment

After running the system on the resulted subsets of the 6-Hours interval we achieved the results shown in **Table 6.4**. We can observe at window 15 we achieved the lowest f-measure, this is due to the bad distribution of events at that window. As shown in **Table 6.5** most events have lower number of tweets. Consequently, frequencies of event triggers belong to those events will be low too. Thus, using the average as a threshold value will eliminate those event triggers. This can be a limitation of our threshold value selection, but our objective in this work is to detect significant events which are the ones with the highest frequencies. On the other hand, window 19 produced the highest f-measure. The distribution of events in that window are shown in **Table 6.6.** The calculated average of recall, precision and f-measure is 0.607, 0.684, and 0.644 respectively.

Table 6.4: Summary of the results achieved using 6-hours time interval

S.	Tweets Count	Event Count	Recall	Precision	F-Measure
1	1010	4	0.5	1	0.666
2	1284	4	0.5	1	0.666
3	1204	5	0.8	0.571	0.666
4	1421	4	0.75	0.6	0.666
5	948	7	0.428	0.5	0.461
6	1247	9	0.555	0.833	0.666
7	2472	10	0.6	0.857	0.705
8	3296	13	0.615	0.8	0.695
9	3681	14	0.642	0.818	0.72
10	3200	13	0.769	0.666	0.714
11	2313	12	0.75	0.75	0.75
12	2845	11	0.727	0.533	0.615
13	1222	7	0.571	0.444	0.5
14	1496	11	0.454	0.5	0.476
15	4344	13	0.230	0.428	0.3
16	3742	10	0.5	0.454	0.476
17	2952	12	0.583	0.7	0.636
18	3613	10	0.6	0.666	0.631
19	4465	9	0.888	0.8	0.842
20	5421	11	0.727	0.8	0.761
21	3554	8	0.625	0.625	0.625
22	4000	9	0.555	0.714	0.625
Average			0.607	0.684	0.644

Table 6.5: Distribution of tweets across events in window 15

Event Label	Number of Tweets
E13	11
E14	16
E16	13
E18	11
E19	134
E20	513
E21	11
E22	1421
E23	24
E24	3
E25	5
E26	3
E30	3

Table 6.6: Distribution of tweets across events in window 19

Event Label	Number of Tweets
E20	32
E21	59
E22	231
E23	322
E24	62
E26	99
E27	521
E30	183
E33	5

6.2.2.2 One-Day Time Interval Experiment

After running the system on the resulted subset of the one-day interval we achieved the results shown in **Table 6.11**. In window 1 we achieved the highest f-measure with a value of 0.888. Events distribution in that window is shown in **Table 6.7**. At the first look, events E01 and E03 display higher frequencies compared to E02 and E05, thus using the average as a threshold both E02 and E05 will be eliminated, but in fact, both E01 and E03 produced different event triggers, which made the frequencies to be distributed among these event triggers, and as a result, a good value of threshold is generated. **Table 6.9** and **Table 6.10** shows a sample of tweets that belong to event E01 and E03 and produce different event triggers. The lowest f-measure value is 0.533 achieved in window 11 and its events distribution is shown in **Table 6.8**. The calculated average of recall, precision and f-measure is 0.654, 0.793, and 0.717 respectively.

Table 6.7: Distribution of tweets across events in window 1 of one-day dataset

Event Label	Number of Tweets
E01	1040
E02	64
E03	462
E05	30

Table 6.8: Distribution of tweets across events in window 11 of one-day dataset

Event Label	Number of Tweets
E23	21
E26	113
E27	118

E30	21
E32	2711
E33	191
E36	494
E39	3
E24	8
E25	3

Table 6.9: Sample of tweets labeled as E01 produces different event triggers

Event Label	E01	
1	**Text**	عاجل | قتلى وجرحى في تفجير انتحاري بحزام ناسف # استهدف حفلاً لجماعة الحوثي بمناسبة المولد النبوي في إب وسط #اليمن (مصدر أمني
	Processed	عاجل قتل جرح تفجير انتحار بحزام ناسف استهدف حفلا لجماع حوث بمناسب مولد نبو اب وسط يمن مصدر امن
	Event Triggers	قتل جرح، تفجير انتحار، استهداف حفلا
2	**Text**	إصابة محافظ إب في التفجير الانتحاري الذي استهدف حفلا :حوثيا في المركز الثقافي وارتفاع القتلى إلى 10
	Processed	اصاب محافظ اب تفجير انتحار ذي استهدف حفلا حوثيا مركز ثقاف ارتفاع قتل ال
	Event Triggers	اصاب محافظ، تفجير انتحار، استهدف حفلا، ارتفاع قتل
3	**Text**	مراسل الجزيرة: إصابة محافظ #إب يحيى الإرياني في التفجير الذي استهدف مركزا ثقافيا في المدينة. #اليمن
	Processed	مراسل جزير اصاب محافظ اب يحي اريان تفجير ذي استهدف مركزا ثقافيا مدين يمن
	Event Triggers	اصاب محافظ، تفجير ذي، استهدف مركزا

Table 6.10: Sample of tweets labeled as E03 produces different event triggers

Event Label	E03	
1	**Text**	بي بي سي: غضب إسرائيلي وأمريكي عارم على توقيع عباس طلب الانضمام للمحكمة الجنائية الدولية: توقيع رئيس السلطة ال
	Processed	بي بي سي غضب اسرائيل امريك عارم توقيع عباس طلب انضمام محكم جنائ دول توقيع رئيس سلط ال
	Event Triggers	توقيع عباس، توقيع رئيس
2	**Text**	نتنياهو في رده على توقيع الرئيس عباس على معاهدة روما يقول ان السلطة هي من يجب ان تقلق من المحكمة الجنائية الدولية لتحالفها مع حركة حماس
	Processed	نتنياهو رد توقيع رئيس عباس معاهد روما يقول سلط يجب تقلق محكم جنائ دول لتحالف حرك حماس
	Event Triggers	رد توقيع، توقيع عباس
3	**Text**	عباس يوقع اليوم طلب الانضمام الى المحكمة الجنائية الدولية بعد رفض مشروع القرار الفلسطيين
	Processed	عباس يوقع طلب انضمام ال محكم جنائ دول رفض مشروع قرار فلسط
	Event Triggers	يوقع طلب، رفض مشروع

Table 6.11: Summary of the results obtained using one-day time interval

#	Tweets Count	Event Count	Recall	Precision	F-Measure
1	2294	4	1	0.8	**0.888**
2	2625	5	0.6	0.75	0.666
3	2195	10	0.5	0.833	0.625
4	5768	13	0.769	0.909	0.833
5	6880	14	0.714	0.833	0.769
6	5158	13	0.769	0.909	0.833
7	2719	11	0.636	0.777	0.7
8	8086	13	0.462	0.75	0.571
9	6566	12	0.667	0.8	0.727
10	9886	11	0.636	0.7	0.666
11	7555	9	0.444	0.666	0.533
Average			0.654	0.793	0.717

6.2.2.3 Evaluation Baseline

Evetar authors used SONDY application which implements three state of the art event detection approaches Guille (2016). They used the same dataset sample that we tried to fetch. They achieved the results shown in **Table 6.12**. We tried to use the same tool on the retrieved data from section 5.1, but unfortunately, the tool require a network file that represents the relation between users. This type of data is not delivered with the dataset. Nevertheless, we used the obtained result as a reference for comparison.

Table 6.12: Summary of the results achieved by other event detection approaches over Evetar

Algorithm	Recall	Precision	F-Measure
EDCoW	0.15	0.09	0.11
Peaky Topics	0.80	0.11	0.19
MABED	0.92	0.61	0.73
Our Approach (6-Hours Interval)	0.607	0.684	0.644
Our Approach (1-Day Interval)	0.654	0.793	0.717

With comparison to the baseline, our approach outperformed both EDCoW and Peaky Topics, however, compared to MABED, we achieved better precision, but there

was a wide difference in recall in favor of MABED. The reason for this is the fetched dataset was incomplete and event-related tweets are lost during the fetch process. Thus our approach failed to detect events with lower frequencies. This is acceptable as our objective is detect significant events.

Overall, our experiments show that having wider time intervals produce good results. Our findings coincide with both Doulamis et al. (2016) and Gaglio, Re, and Morana (2016). Since wide intervals can cover the full context of an event thus all its feature will be present and showing higher frequent pattern.

6.3 Summary

This chapter presented the experiments conducted and the evaluation of our approach. It discussed the evaluations measurements and introduced event recall, event precision, and the F-measure. Also, we explained the experimental setup by discussing both the 6-hours time interval and the one-day time interval experiments. We presented the resulted datasets produced by each time interval. Then we showed and discussed the results of each experiment and compared them to a baseline. We achieved an F-measure of 0.644 for 6-hours time interval and 0.717 for one-day time interval. We have shown that our results are acceptable compared to other related works.

Chapter 7
Conclusion and Future Work

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Users of microblogs tend to write about what they are experiencing. Especially, when a real-world event occur in their surrounding environment, they tend to write about it to inform their network of users. Information about occurring events are buried within a huge amount of unnecessarily textual data. There is an urgent need to detect events for decision makers to take actions early.

In this research we have designed an approach that detects events from Arabic text founded in microblogs. The approach based solely on the textual features of the text, without relying on any platform-specific features such as hashtags, mentions, retweets that are present in case of the Twitter platform. We depended only on event triggers extracted from the text which shown higher importance than other keywords.

Our approach consists of the following components:

- Data collection
- Data Preprocessing
- Event Trigger Extraction
    - Apply Pre-Defined Rules on Part of Speech Tags
    - Extract Event Triggers
- Significant Event Detection
    - Identify Top Event Triggers
    - Apply Soft Frequent Pattern Mining

We have shown that our approach has the ability to detect events by clustering event triggers only.

In our experiments, we used an existing Twitter-based dataset called Evetar. The dataset was a collection of tweets IDs. We built a small tool that fetch the actual content from Twitter. We were able to fetch 59,732 tweets. We used two configurations that had different time-intervals. First a 6-hour time interval were used which resulted in

22 subsets, then a one day time interval were used which resulted in 11 subsets. We achieved an F-measure of 0.644 and 0.717 respectively. We have shown that our results are acceptable compared to other related works.

7.2 Recommendations

There are some recommendation for enhancing the field of event detection from microblogs, these are:

- A well-established framework for evaluating event detection method is highly needed.
- Twitter-based datasets are very likely to decay over time as tweets may disappear from twitter platform for different reason, thus a robust way to establish a dataset that target the event detection task is required.

7.3 Future Work

Based on the results of the experiments and limitations we faced in our thesis, this work can be improved as follows:

- Use an efficient and adaptive threshold function to extract event triggers instead of using the average of the frequencies.
- Adapt the approach to work with real-time streams.
- Add a text summarization component to the existing methodology so that the approach can be more useful to end users.
- Use dynamic time-intervals instead of the fixed one used in this work.
- Enhance the significant event authenticity by providing a solution for retweeted event.

References

Abuleil, S. (2007). *Using nlp techniques for tagging events in arabic text.* Paper presented at the Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on.

Aggarwal, C. C., & Han, J. (2014). *Frequent pattern mining*: Springer.

Aliane, H., Guendouzi, W., & Mokrani, A. (2013). *Annotating events, time and place expressions in arabic texts.* Paper presented at the Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013.

Alkhamees, N., & Fasli, M. (2016). *Event detection from social network streams using frequent pattern mining with dynamic support values.* Paper presented at the Big Data (Big Data), 2016 IEEE International Conference on.

Allan, J. (2002). Introduction to topic detection and tracking *Topic detection and tracking* (pp. 1-16): Springer.

Almerekhi, H., Hasanain, M., & Elsayed, T. (2016). *Evetar: A new test collection for event detection in arabic tweets.* Paper presented at the Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval.

Alsaedi, N., & Burnap, P. (2015). *Arabic event detection in social media.* Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics.

Apache. (2014). Apache Lucene. Retrieved from https://lucene.apache.org/

Atefeh, F., & Khreich, W. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence, 31*(1), 132-164.

Baradaran, R., & Minaei-Bidgoli, B. (2015). Event Extraction from Classical Arabic Texts. *Int. Arab J. Inf. Technol., 12*(5), 494-502.

Becker, H., Naaman, M., & Gravano, L. (2011). Beyond Trending Topics: Real-World Event Identification on Twitter. *Icwsm, 11*(2011), 438-441.

Consortium, L. D. (2005). ACE (Automatic Content Extraction) Arabic Annotation Guidelines for Entities.

Cordeiro, M. (2012). *Twitter event detection: combining wavelet analysis and topic inference summarization.* Paper presented at the Doctoral symposium on informatics engineering.

Dou, W., Wang, K., Ribarsky, W., & Zhou, M. (2012). *Event detection in social media data.* Paper presented at the IEEE VisWeek Workshop on Interactive Visual Text Analytics-Task Driven Analytics of Social Media Content.

Doulamis, N. D., Doulamis, A. D., Kokkinos, P., & Varvarigos, E. M. (2016). Event detection in twitter microblogging. *IEEE transactions on cybernetics, 46*(12), 2810-2824.

Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP), 8*(4), 14.

Ferracani, A., Pezzatini, D., Landucci, L., Becchi, G., & Bimbo, A. D. (2017). *Separating the Wheat from the Chaff: Events Detection in Twitter Data.* Paper presented at the Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing.

Gaglio, S., Re, G. L., & Morana, M. (2015). *Real-time detection of Twitter social events from the user's perspective.* Paper presented at the Communications (ICC), 2015 IEEE International Conference on.

Gaglio, S., Re, G. L., & Morana, M. (2016). A framework for real-time Twitter data analysis. *Computer Communications, 73*, 236-242.

Guille, A. (2016). SONDY an open source social media data mining software. Retrieved from https://github.com/AdrienGuille/SONDY

Guille, A., & Favre, C. (2014). *Mention-anomaly-based event detection and tracking in twitter.* Paper presented at the Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on.

Hkiri, E., Mallat, S., & Zrigui, M. (2016). Events Automatic Extraction from Arabic Texts. *International Journal of Information Retrieval Research (IJIRR), 6*(1), 36-51.

Hurlock, J., & Wilson, M. L. (2011). *Searching Twitter: Separating the Tweet from the Chaff.* Paper presented at the ICWSM.

Katragadda, S., Benton, R., & Raghavan, V. (2017). Framework for real-time event detection using multiple social media sources.

Larkey, L. S., Ballesteros, L., & Connell, M. E. (2002). *Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis.* Paper presented at the Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.

Lin, J. C.-W., Ren, S., & Fournier-Viger, P. (2018). MEMU: More Efficient Algorithm to Mine High Average-Utility Patterns With Multiple Minimum Average-Utility Thresholds. *IEEE Access, 6*, 7593-7609.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). *The Stanford CoreNLP natural language processing toolkit.* Paper presented at the Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations.

Microsoft. (2017). Visual Studio. Retrieved from https://www.visualstudio.com/

Mohammad, A.-S., & Qawasmeh, O. (2016). Knowledge-based Approach for Event Extraction from Arabic Tweets. *International Journal of Advanced Computer Science & Applications, 1*, 483-490.

Omnicore. (2018). Twitter by the Numbers. Retrieved from https://www.omnicoreagency.com/twitter-statistics/

Petkos, G., Papadopoulos, S., Aiello, L., Skraba, R., & Kompatsiaris, Y. (2014). *A soft frequent pattern mining approach for textual topic detection.* Paper presented at the Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14).

Petrovic, S. (2013). Real-time event detection in massive streams.

Petrović, S., Osborne, M., & Lavrenko, V. (2010). *Streaming first story detection with application to twitter.* Paper presented at the Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics.

Phuvipadawat, S., & Murata, T. (2010). *Breaking news detection and tracking in Twitter.* Paper presented at the Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on.

Shamma, D. A., Kennedy, L., & Churchill, E. F. (2011). *Peaks and persistence: modeling the shape of microblog conversations.* Paper presented at the Proceedings of the ACM 2011 conference on Computer supported cooperative work.

Shao, M., Li, J., Chen, F., Huang, H., Zhang, S., & Chen, X. (2017). *An efficient approach to event detection and forecasting in dynamic multivariate social media networks.* Paper presented at the Proceedings of the 26th International Conference on World Wide Web.

Taghva, K., Elkhoury, R., & Coombs, J. (2005). *Arabic stemming without a root dictionary.* Paper presented at the Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on.

TweetSharp. (2017). TweetSharp. Retrieved from https://github.com/Yortw/tweetmoasharp

Twitter. (2017). Twitter. Retrieved from http://twitter.com/

Twitter. (2018). GET statuses/lookup — Twitter Developers. Retrieved from https://developer.twitter.com/en/docs/tweets/post-and-engage/api-reference/get-statuses-lookup

Wang, X., Tokarchuk, L., & Poslad, S. (2014). *Identifying relevant event content for real-time event detection.* Paper presented at the Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on.

Weng, J., & Lee, B.-S. (2011). Event detection in twitter. *Icwsm, 11*, 401-408.

Yılmaz, Y., & Hero, A. O. (2016). Multimodal Event Detection in Twitter Hashtag Networks. *Journal of Signal Processing Systems, 90*(2), 185-200.