QATAR UNIVERSITY

COLLEGE OF ENGINEERING

BUILDING A TEST COLLECTION FOR SIGNIFICANT-EVENT DETECTION IN ARABIC TWEETS

BY

HIND ALI AL-MEREKHI

A Thesis Submitted to the Faculty of

College of Engineering

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science

January 2016

COMMITTEE PAGE

The members of the Committee approve the thesis of Hind A. Al-Merekhidefended on the 22nd of February 2016.

Dr. Tamer Elsayed
Thesis/Dissertation Supervisor

Dr. Abdelkarim Erradi
Committee Chair

Dr. Joemon Jose
Committee Member

Dr. Somaya Al-Maadeed
Committee Member

Dr. Ali Jaoua
Committee Member

Approved:

Rashid Alammari, Dean, College of Engineering

Abstract

With the increasing popularity of microblogging services like Twitter, researchers discovered a rich medium for tackling real-life problems like event detection. However, event detection in Twitter is often obstructed by the lack of public evaluation mechanisms such as test collections (set of tweets, labels, and queries to measure the effectiveness of an information retrieval system). The problem is more evident when non-English languages, e.g., Arabic, are concerned. With the recent surge of significant events in the Arab world, news agencies and decision makers rely on Twitters microblogging service to obtain recent information on events. In this thesis, we address the problem of building a test collection of Arabic tweets (named EveTAR) for the task of event detection.

To build EveTAR, we first adopted an adequate definition of an event, which is a significant occurrence that takes place at a certain time. An occurrence is significant if there are news articles about it. We collected Arabic tweets using Twitter's streaming API. Then, we identified a set of events from the Arabic data collection using Wikipedias current events portal. Corresponding tweets were extracted by querying the Arabic data collection with a set of manually-constructed queries. To obtain relevance judgments for those tweets, we leveraged CrowdFlower's crowdsourcing platform.

Over a period of 4 weeks, we crawled over 590M tweets, from which we identified 66 events that cover 8 different categories and gathered more than 134k relevance judgments. Each event contains an average of 779 relevant tweets. Over all events, we got an average Kappa of 0.6, which is a substantially acceptable value. EveTAR was used to evaluate three state-of-the-art event detection algorithms. The best performing algorithms achieved 0.60 in F1 measure and 0.80 in both precision and recall. We plan to make our test collection available for research, including events description, manually-crafted queries to extract potentially-relevant tweets, and all judgments per tweet. EveTAR is the first Arabic test collection built from scratch for the task of event detection. Additionally, we show in our experiments that it supports other tasks like ad-hoc search.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Acknowledgements

My first words of gratitude goes towards my family for always keeping up with my antics. I thank my mother Aisha, for her unconditional love and support throughout my academic life. Her silent prayers and sleepless nights worrying about me have touched my heart deeply. I thank my elder sisters Noor and Maha, for being great role models in life. I thank my youngest sister Alanoud, for her assistance with my thesis completion. I thank my elder brother Mohammad, for sharing his knowledge in computing and providing assistance whenever I needed it. I would like to also thank my eldest brothers Hamad and Jumaa for supporting my decisions and encouraging me to pursue a Masters degree.

The completion of this thesis was made possible through the efforts of my supervisor Dr. Tamer Elsayed. His tremendous efforts during one of his famous "IR in a nutshell" seminars fueled my interest in the field of information retrieval. I'm extremely thankful to him for accepting to be my mentor. Over the past couple of years, I made a lot mistakes and learned from them thanks to Dr. Tamer's assistance. I'm grateful for all the hints and tips that he tends to offer whenever I need them. I'm forever thankful for all the time and effort that he dedicated to revise this work. The experience I had when working with Dr. Tamer really helped me improve my research skills and I'm truly honored to have him as my mentor.

I would like to also express my gratitude towards all the members of the Information Retrieval group at Qatar University. Dr. Tamer, Dr. Marwan, Dr. Mucahid, Maram, Mrs. Rana, Reem, Abeer, Nihal, and Fatima. Thank you for your great teamwork spirit and kindness. Being a part of this team helped me improve my skills as a researcher and a fellow member. I also thank my small group of friends for always being there when I needed them the most.

Lastly, I offer my deepest appreciation to the office of Education, Training and Development of Qatar Foundation for their continuous support. Thank you for giving me this opportunity to follow my dreams and continuing my education.

Dedication

To my dear mother. I cannot find the right words to express my gratitude and appreciation towards you. Your kind advice and encouraging words saved me from despair. So thank you from the bottom of my heart for always believing in me when I doubted myself. I couldn't have completed this thesis without you.

CHAPTER 1. INTRODUCTION

Over the past few years, online microblogging services like Twitter[1]gained an immense growth in popularity among users. Twitter's microblogging service allows users to communicate through short messages, commonly known as *tweets*. Unlike regular messages, tweets are very limited in length and cannot exceed 140 characters. The power of Twitter comes from the fact that tweets are shared in real-time. With the aid of Twitter's online web service and mobile application, users can post and retweet (i.e, repost) messages instantaneously [30]. This allows different kinds of messages to be shared (e.g, personal, opinion, earthquakes, disasters, sports, and political tweets) [61]. However, this leads Twitter's microblogging service to witness a surge in the volume of posted tweets. Recent statistics from April of 2015 show that the total number of tweets posted per day is about 500 million tweets[2]. This number might indicate that most of these tweets are either spam [24], repetitive, or uninteresting, which causes the problem of information overload [50].

Naturally, the overwhelming popularity of Twitter led several parts of the world such as the Arab region to use it as a medium for exchanging messages. To update people on critical events, news agencies like Al Jazeera and Al Arabiya often rely on Twitter's microblogging service to post their latest news. Furthermore, both news agencies are highly interested in tracking events that are not yet published. In this context, an *event* is an important happening that occurs at a particular point in time. For example, Al Jazeera journalists need to know the events associated with the revolutions in the Arab world, so they use Twitter as an information source to update their reports. Another scenario might consider using Twitter as a guide to save lives. As life-threatening disasters like hailstorms occur, it is important for rescue squads to get quick information on the situation and send help to save people. In this case, rescue squads could get their

[1] https://www.twitter.com

[2] http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/

information from live Twitter feeds that discuss the disaster. Examples of such events from the Arab world are given in Figure 1.1. The events are given on the left side of the figure, while some corresponding tweets that discuss those events are on the right. The first event in the figure discusses the death of two Saudi men near the Iraqi borders. The second event translates to Cristiano Ronaldo winning the FIFA Ballon d'Or 2014. The third event is about the resignation of Italy's president Giorgio Napolitano.

Figure 1.1. Examples of three events and event-related tweets in Arabic.

The increasing popularity of Twitter led several researchers to explore the problem of event detection in different ways. Event detection techniques can be either *online* or *offline*. Online detection deals with streams of tweets and detects events as tweets arrive, whereas offline detection finds events using variety of complex techniques that are not efficient for online settings. One of the most prominent event detection techniques is *clustering*. To detect events, clustering attempts to group several tweets that discuss an event together in groups (called clusters). Clusters are later classified as either *event-related* or *not event-related* [7, 48, 6, 39, 29]. The other technique used to detect events is based on *anomaly detection*. Anomaly stands for any witnessed abnormality

or difference in tweets. For instance, anomalies can be found when sentiment-related words are distributed in a different way than normal [40]. Furthermore, anomalies can be associated with terms that were trending in the consecutive or fixed time windows [22, 20, 62, 53, 56, 8]. Almost all anomaly-based event detection techniques depend on spatial analysis, which requires the usage of timestamps on tweets to detect spikes in the activity on Twitter.

Another popular event detection technique looks into the problem of *first story detection (FSD)*. In this case, systems are expected to identify the first event that occurs in a stream of tweets[45, 46]. By using the nearest neighbor distance to other documents as a similarity measure, FSD uses the distance with a certain threshold and determines if the tweet is new and novel to be considered a first story. The last event detection technique is about *predefined event types*. Unlike the previous event detection techniques that target all types of events, this technique looks into specific event categories. For example, earthquakes [51], criminal and disastrous events [30], disruptive events [6], sports [66], and brands [36].

Evidence from the Arab social media report[3] show that as of March 2014, an average of 17 million tweets are posted every day. Such tweets are extremely noisy, full of typos and redundancy. Hence, it is difficult to use Twitter's stream to manually identify events in the Arab world due to information overload [35]. It is evident that we need tools that can automatically identify events from Twitter streams. More importantly, we need an automatic way of evaluating the performance of such tools, which calls for a *test collection*. A test collection consists of a set of documents, topics, and relevance judgments (i.e, labels) that specify if documents are relevant to particular topics or not. Sometimes it is difficult to build a full set of relevance judgments due to time constraints, size of data collection, or lack of resources to judge documents [52]. The goal behind creating test collections is to help researchers with the evaluation of their event detection systems. With the aid of test collections, different event detection systems can be compared in

[3] http://www.arabsocialmediareport.com

performance and improved to enhance the quality of their output. The problem with the existing event detection test collections is that most of them focus on English [35]. Moreover, the only work that tackles the problem of event detection in Arabic tweets does not provide any data that could be used in further research [6]. Therefore, this study aims at building a test collection (named *EveTAR*) for the task of event detection. More specifically, the collection was designed for the problem of detecting *significant events*. A significant event is defined as an occurrence that happens at a particular time in a specific location and is discussed by the media. For instance, a significant event has a news article written about it on the web.

The adopted method to build *EveTAR* consists of five main steps, as illustrated in Figure 1.2:

1. Collect tweets: obtaining the tweet stream that will be used to build the collection. The stream was obtained for a period of one month (January 2015).

2. Identify events: finding events from the tweet stream that spanned a single month.

3. Gather event-related tweets: using the events to search the tweet stream for potentially event-related tweets.

4. Obtain relevance judgments: getting labels for the potentially event-relevant tweets that determine if they are actually relevant to each event or not.

5. Evaluate label quality: analyzing the obtained labels and determining their quality across events, then using the test collection by applying some state-of-the-art event detection systems on the collection [21].

The resulting test collection contains relevance judgments of more than 135,000 tweets that span 66 events during the month of January 2015. The built test collection can be used for evaluating existing event detection techniques. Moreover, it can be used to

support other information retrieval tasks such as *ad-hoc search*, *summarization*, *Tweet Timeline Generation (TTG)*, and *filtering* [4].

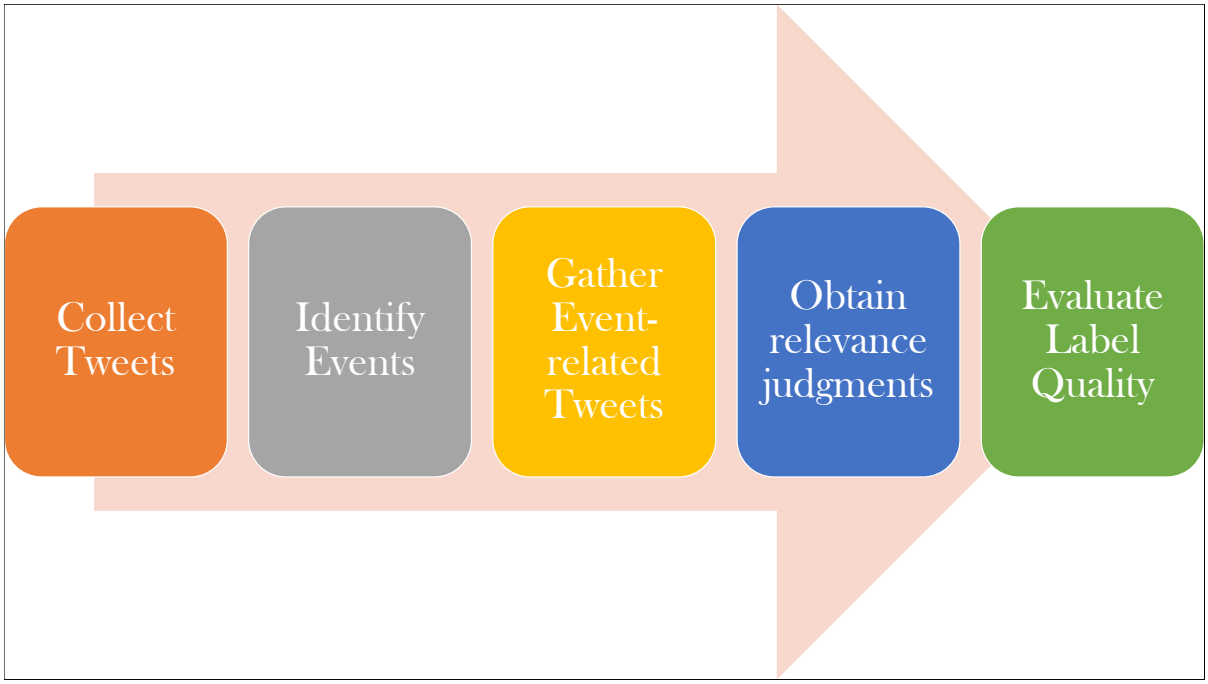


Figure 1.2. The five main steps to build *EveTAR* for the task of event detection.

1.1 Research Questions

In building a test collection for event detection in Arabic tweets, we address two main research questions:

- **RQ1:** How can we design a test collection that is reusable and supports multiple tasks?

  To support different information retrieval tasks, a test collection must take into consideration the requirements of each task. Here, we try to answer the following sub-questions:

  - Can the test collection be used to evaluate systems built for different tasks?
  - How can topics and judgments be collected to serve multiple tasks?

[4] https://sites.google.com/site/microblogtrack/

- **RQ2:** How can we use crowdsourcing in building the test collection?

  Using a crowdsourcing platform to obtain relevance judgments can save a lot time and effort. However, research shows that using such platforms can be a challenging task [35]. To get the best out of crowdsourcing, we believe that it is important to benefit from the platform resources to obtain relevance judgments of high quality. In this case, Learning how to use a crowdsourcing platform is not enough. We are also interested in answering a few improtant questions such as:

  - What makes a labeling job good? And what are the components of a good labeling job for the task of event detection?

    Since crowdsourcing deals with a heterogeneous network of workers (i.e, labelers), we think it is important to incorporate a good job design to boost the quality of the obtained labels.
  - Are the obtained labels through crowdsourcing reliable?

  Furthermore, we look into the quality of the obtained test collection by examining the labels and computing several statistics on the obtained results.

1.2 Contributions

We have three major contributions out of our study:

1. To our knowledge, *EveTAR* is the *first* test collection for the task of event detection in Arabic tweets. The collection supports other tasks like ad-hoc search. Additionally, it can be extended to support tasks like filtering, summarization, and Tweet Timeline Generation (TTG).

2. We make the full test collection available for research, including:

   - The list of 66 events that were identified by the study.
   - The detailed design of the crowdsourcing task.

- The relevance judgments and tweet ID's of 134,069 tweets.
- The queries for the 66 events for ad-hoc retrieval tasks.
- The full output of the crowdsourcing tasks, including the statistics of each event.
- Different representations of the test collection to support ad-hoc search and filtering.

3. We show that our test collection can be used to evaluate existing state-of-the-art automatic event detection systems.

1.3 Thesis Outline

The reminder of this thesis is organized as follows. First, in Chapter 2 we present some background information on event detection along with a detailed study on test collections and microblog event detection systems. The details given in Chapter 3 describe how candidate events were generated and selected to build the test collection, in addition to the pilot and full evaluations that were conducted using crowdsourcing. Information about the dataset and the evaluation results conducted on the test collection are given in Chapter 4. Final concluding remarks and future directions are given in Chapter 5.

CHAPTER 2. BACKGROUND AND RELATED WORK

To build a test collection for the task of event detection, it is essential to provide some background information on the problem of event detection. Hence, we provide a detailed survey of the most prominent work done in event detection. First, we focus on the definition of event detection in different scopes (including microblogs), then we look at the efforts done in building test collections for different information retrieval tasks. We also look at the research studies that focused on event detection in microblogs and we discuss some of the applications of event detection in microblogs. Finally, we analyze some of the few works that attempted to build test collections for the task of event detection in microblogs.

2.1 Background

2.1.1 Definitions of Event

Starting from the general definition of an event that was adopted by the TDT project, and ending with a very specific definition of what is known as a *sub-event*. The following definitions are meant to show how the definition of an event changes based on the context of usage.

- The Topic Detection and Tracking (TDT) task defines an event as something that happens at some specific time and place, and the unavoidable consequences, such as accidents, crimes, natural disasters, and presidential elections [4, 45, 26].

- The New Event Detection (NED) task states that NED is concerned with developing systems that can detect the first story on a topic of interest, where a topic is defined as a "seminal event or activity, along with directly related events and activities"[45]. Examples of an activity is the sinking of an oil tanker, first story is the article that discusses the sinking of the tank, and other stories with the same topic discuss environmental damage, the commercial effort and so on.

- An event in the context of Twitter [9] is defined as a real world occurrence with the following properties: (a) it is associated with a time period T_e and (b) a substantial stream of tweets that discuss the event that occurred at the time period T_e.

- Another definition of an event in Twitter [35] states that an event is a significant thing that happened at some specific time and place. As for significance, an event is significant if it is discussed in the media (i.e, there is a news report or article written about it).

- As for disastrous event detection in Twitter [51], an event is an arbitrary classification of a spacetime region. Moreover, an event might have actively participating agents, passive factors, products, and a location in space/time. Examples of disasters include earthquakes, typhoons, and traffic jams, which are visible through tweets.

- News event detection in Twitter has a similar definition to [51], however, it focuses on specific events rather than generic ones (in particular, fire-in-factory and labor-strike). Furthermore, [2] defines a structured set of event-objects that contain precise information about each event.

- Events in Twitter are defined in [25] as interest-driven activities that occur at a specific time and location. While sub-events are specific information about location or time, which are assumed to be co-located with the major event.

- Sub-events in crisis situations in Twitter are events during a disaster which are separated from other events w.r.t. time or location [47, 1]. For example, during an earthquake, in one place a bridge might collapse, while at the same time in another location some buildings might be critically damaged.

- Sub-events in soccer games in Twitter are the live tweets that occur during an event which describes those sub-events (e.g, the goals or penalties during a soccer game)[16].

Our definition of a significant event is similar to the many definitions in [35]. However, we emphasize on the significance of an event, which is often neglected in most of the event definitions in [6, 9, 45]

2.1.2 Event Detection

The problem of event detection is relatively old in the realm of information retrieval. In fact, it was one of the classical tasks that was supported by the U.S. Government's Defense Advanced Research Projects Agency (DARPA), under the project of Topic Detection and Tracking (TDT)[4].The TDT project involves five tasks that allow researchers to explore a variety of problems related to broadcasting news media. Each year, the National Institute of Standards and Technology (NIST)monitors the evaluation of the TDT project [18]. One of the five TDT tasks focuses on New Event Detection (NED) and event tracking, which was heavily discussed by Allan et al.[5]. The new event detection task aims at verifying if events exist in a stream of broadcasting documents. For this task, the document streams were constructed from newswire websites and speech recordings that were transcribed by humans from several CNN news shows.

Over the course of the TDT task, several researchers attempted to solve the problem of new event detection. One of the earliest attempts was by Allan et al.[42]. In this work, the researchers aimed at simulating an online setting to detect new events by looking at the first document that discusses the event in an incoming stream. The proposed approach relies on a single pass clustering algorithm and considers the characteristics of an event as a threshold model. Implementation of the proposed algorithm relied on a combination of the ranked-retrieval technique of Inquery, which consists of selection and feature extraction based on relevance feedback[28]. All the feature vectors were represented by Term Frequency Inverse Document Frequency (TFIDF) weights. To evaluate the performance of the single pass online clustering technique, a corpus of 15863 documents was used. The data was obtained between early July 1994 and late June 1995

from CNN news show transcriptions and Reuters newswire documents [42]. A total of 25 events were selected and used to evaluate the performance of the algorithm against previous work from the TDT task. The proposed evaluation method considered recall, precision, F1-measure, miss rate, false alarm rate, and distance from the origin as evaluation metrics. By running the algorithm on 11 passes, the results show that adding a time penalty on the data increases the overall performance.

In a similar work, Yang et al. [65, 64] attempted to solve the problem of online event detection through clustering. However, unlike the approach that was discussed previously, this work considered a hierarchical and non-hierarchical clustering technique to detect events in streams of documents. The hierarchical clustering is performed by first classifying documents into a set of comprehensive topics, then looking for new unseen topics in each topic. The feature computation relies on the TFIDF representation of documents. Moreover, the considered features were computed by removing topic-specific stop words and giving weights to named entities [65]. By leveraging the same corpus that was used by Allan et al.[42], the researchers in this work focused on both temporal and content based features to perform the clustering. Resulting hierarchies from the clustering approach showed that it was possible to identify unknown events that occurred in the past. Moreover, the underlying temporal characteristics of document clusters can show some interesting patterns that aid in detecting events in the past or online. To evaluate the performance of the hierarchical clustering technique, a set of 25 manually labeled events were used in a similar fashion to what was done in [42]. Results showed that the hierarchical technique scored 82% in past event detection. However, the technique's performance derogates in online clustering, achieving an F1 score of 42% [64].

To view the problem of NED from a different perspective, Brants et al. [10] introduced a technique that relies on an incremental TFIDF model. The methodology builds up on the previous work [42, 65] and extends the usage of the TFIDF model. The new additions to the model are: generalization of models specific to particular sources, normalization of similarity scores based on the averages of particular documents, segmentation of docu-

ments, normalization of similarity scores based on the averages of particular source pairs, and usage of inverse event frequencies to reweight terms. The authors of this work evaluated their technique on TDT3 and TDT4 test data. Furthermore, instead of using the well-known cosine distance, the proposed technique replaced it with Hellinger distance. Interestingly, the researchers in this work discussed two proposed additions to the TFIDF model that failed to improve it. The failed techniques are based on time information and the usage of the vocabulary in the look-ahead data in the TFIDF model. Evaluation results show that the new additions to the TFIDF model were able to collectively improve the detection by 18% [10].

Efforts to improve the task of NED were further exploited by Kurman and Allan [26]. This work introduced two main modifications to the classic NED approaches: the usage of text classification approaches and incorporating named entities in the classification process. Enhanced NED looked at different representations of documents in the vector-space model of NED systems. In this work, the classic NED system model was replaced by three representations of each document. The motivation behind this approach is to account for prominent terms and named entities that occur in new events. Moreover, weight scores were assigned after classifying stories into broad category types to improve the influence of individual terms. Hence, a document was represented by three vectors: a vector α that accounts for all terms (after removing stop words), a vector β that accounts for named entities only, and a vector γ that accounts for non-named entity terms. To discover named entities, this work relied on the BBN identifier [26]. Evaluation results show that the system contributed to significant improvement in event detection when compared with baselines. However, the authors emphasized that their rules for including named entities require further research to be utilized in NED tasks.

As for other media types, the Social Event Detection (SED) task [41] aims at identifying events from image metadata. The task was designed as part of the MediaEval 2011 benchmark to study the diffusion of multimedia content in social media platforms. By using social media streams from popular photo sharing platforms, like Instagram and

Flickr, systems are expected to identify associations between multimedia content and events. Video retrieval is also addressed by TRECVid evaluation benchmark initiatives [55]. In fact, two of the campaigns held by TRECVid tackle the problem of event detection in surveillance video footage. Using high-level feature extraction, events can be identified from different surveillance cameras. In this case, the targeted events are cases in which luggage is left behind by passengers in airports or subways.

2.1.3 Test Collections

The typical requirements for evaluating an information retrieval system include having a test collection, which consists of a collection of documents, a set of topics, and relevance judgments that specify if documents are relevant to particular topics. Creating a test collection for evaluating modern information retrieval systems is an expensive task because such systems require millions of documents to be evaluated. However, Cormack et al. [15] attempted to solve this problem by introducing two techniques to build large test collections efficiently. The first technique, *Interactive Searching and Judging*, was introduced to improve the quality of the produced judgments. Instead of providing annotators with a set of random documents to label, the first technique allows small research groups to use their limited resources to interactively select which documents to judge. The second technique, *Move-to-Front Pooling* aimed at improving on the classic pooling technique that was used to get relevance judgments. In a normal pooling scenario, the top k documents from each retrieval system are added to a pool for judgment. However, the Move-to-Front pooling technique improves on this approach by getting a different number of documents from each retrieval system based on it's performance. Hence, allowing the efficient creation of high quality test collections with minimum efforts.

When building test collections for information retrieval tasks, it is common to follow the well-established Cranfield evaluation paradigm, which assumes the completeness of the relevance judgments. However, Buckley et al. [11] argued that it is difficult to

strictly satisfy the completeness condition. Since pooling techniques that conferences like the Text REtrieval Conference (TREC) follow are difficult to achieve within short periods of time. Many users submit runs to the pool, which requires a lot assessor time. Therefore, the authors proposed an evaluation metric that shows robust performance when relevance judgments are incomplete. The new measure, named bpref, which stands for binary preference, looks at the fraction of non-relevant documents that were retrieved before the relevant documents. The evaluation metrics that were used in the study were mean average precision (MAP), precision at 10, and R-precision. The conducted experiments measured the change in systems rankings using Kendalls τcoefficient. Furthermore, experiments measured the effect of judgments completeness in all judgments, incomplete judgments, and imperfect judgments.

To further address the challenges associated with building large test collections, Carterette et al. [13] presented a new perspective on average precision (AV) to connect evaluation with test collection construction. The study shows that it is possible to gain high confidence when ranking a set of systems with a minimal set of relevance judgments. Building a test collection with average precision can be done through an algorithm that selects documents based on the available relevance judgments. This means that average precision is normally distributed across all the potential relevance judgments in the entire unjudged collection. The data used for the study was from the Aquaint corpus and TREC 4 & 5 disks. With the assistance of real annotators, the authors show that it took only six hours for the ranking confidence to reach 90%. This proves that the algorithm can work in different retrieval environments when small amounts of relevance judgments are available. To evaluate the algorithm, the authors conducted several tests to compare the mean average precision (MAP) with ϵMAP, the effect of the number of relevance judgments, how time affects confidence, and how reusable a test collection can be.

All of the work that was discussed earlier focused on minimizing the efforts to build test collections. However, none of them thought about reusing relevance judgments like

Carterette[12]. The idea behind this work is to help researchers with limited resources to construct low cost test collections to evaluate new retrieval systems. While the judgments produced by this technique might be useful for a single evaluation of a system, this might not be the case when reusing them with a new system. However, it is still useful when there is a few relevance judgments of a new system. Which means that this technique values the smallest number of judgments and uses it to build the test collection. A small number such as five judgments from two systems can be used to evaluate ten systems. The introduced model estimates the confidence in the set of few judgments based on a formal definition of reusability. The proposed algorithm combines techniques from Minimal Test Collection (MTC) construction and Robust Test Collection (RTC). Results show that the RTC confidence estimates are more accurate when compared to MTC [12].

What Carterette's work failed to address was the problem of evaluating a new unseen system at a low budget. Yet, Hosseini et al. [23] proposed a method to achieve this goal. The technique relies on two stages to build the test collection. The first stage starts with some queries and adopts a traditional pooling technique. However, only part of the budget is used to get the relevance judgments for some of the participating systems. The second stage involves refining the test collection by analyzing the available relevance judgments and adding priority to queries and documents. The aim of doing this is to improve the effectiveness of the test collection for comparative evaluations with other systems. Prioritizing of the query was formulated as a convex optimization problem, which allowed the authors to experiment with different constrains. The second part of the test collection construction budget was used to evaluate query-document pairs based on their priority score. Hence, reducing the cost of the test collection construction by only expending it to participating systems in the second phase. Results show that the proposed technique improved the reusability of the test collection and was cost efficient [23].

On a similar note, Rajput et al [49] investigated the reusability of test collections. The work relied on a small number of valuable information "nuggets" that get manually

extracted by assessors. In this work, the authors emphasized on the importance of importance of using nuggets in building high quality test collections. Issues like reusability, applicability, and scalability were addressed by this work. By using a TREC collection to test their methodology, the authors were able to build SampleAdHoc and SampleWeb collections in one sixth and half the time that TREC used to build the same collection respectively. Moreover, Kendall τ values of pilot studies were above 0.9 when compared to the relevance judgments of TREC systems. Such results proves the efficiency of the nugget based approach [49].

Test collection reliability is an important factor that affects the cost of building the collection. Urbano et al.[58] presented a work that aimed at filling the gap between the techniques used to measure the reliability of test collections. To build a reliable test collection, having a sufficient number of queries is essential. The Generalizability Theory (GT) that was used to provide statistical reliability indicators is too complex to interpret. To compare data-related reliability measures and GT measures, the authors looked at more than 40 TREC collections. Experimental results show that having 50 queries is not sufficient to achieve the desired reliability. Although GT reliability measures are powerful in assessing reliability, they have their own drawbacks. The first issue is that they are extremely sensitive to specific systems when assessing the reliability of other systems. The second drawback is that it requires a large number of systems and queries to assess system reliability. Therefore, the authors advice against using GT techniques to build test collections from scratch. The better reliability assessment approach is to look at interval estimates of stability indicators [58].

The research works discussed previously focused on techniques used for building test collections efficiently with limited resources. In fact, most of the work on building test collections tends to focus on classical information retrieval tasks, like ad-hoc search. However, we discovered that a few research efforts addressed the problem of building test collections for tasks like event detection. One of the earliest attempts was by Yang et al. [63], where they looked at issues pertaining the lack of labeled event-related relevance

judgments. In cases where the number of unlabeled data is more than the labeled data, or when events are too short, the work explored text categorization techniques based on k Nearest Neighbor (kNN) algorithm and Rocchio method. The goal behind using kNN is to improve the rate of tracking events.

2.2 Event Detection in Microblogs

The problem of event detection in microblogging services like Twitter is not new, as many researchers tackled this problem. Starting with the Topic Detection and Tracking (TDT) project that was discussed in the event detection section[26]. The TDT project focused on the problem of organizing newswire stories based on the events that were discussed in those stories. In microblogs, event detection is quite similar to the detection task from the TDT project. Because in both cases, a system is given a chronological stream of documents and asked to put each document into a proper cluster based on the events in that document. However, the difference between both tasks is evident in the huge volume of data that comes from Twitter compared to TDT task. This issue causes many challenges associated with event detection in microblogs.

The first issue with event detection in microblogs is that tweets tend to be short, noisy, and full of grammatical and spelling mistakes. This makes it important to handle these issues and consider them in the event detection system. The second major problem is the huge size of microblogging data, which exceeds orders of magnitudes the data used in the TDT task. Therefore, it is important for event detection systems in microblogs to cope with the increasing volume of data efficiently. The third issue is that most microblog posts are ordinal. Hence, an event detection system for microblogs must discover event-related posts and filter-out necessary posts. Furthermore, all of the event detection systems in the TDT task are not designed to cope with the real-time nature of Twitter. As tweets exhibit different characteristics when compared to regular lengthy documents, such TDT event detection systems would perform poorly and slowly on tweets.

The upcoming section describes some of the prominent event detection applications in microblogs. Followed by a section that surveys different event detection test collections in microblogs.

2.2.1 Event Detection Applications in Microblogs

The work done by Petrovic et al. [45] benefits from Locality Sensitive Hashing (LSH) to detect events in Twitter. The idea behind LSH is that similar documents are placed together in the same bucket of a hash table. The proposed method shows that with high probability, it is possible to reduce the size of the candidate set to a fixed number that consists of the nearest neighbors. By doing that, the clustering task performs in O(1) time, when a method is deployed to reduce the variance when no neighbors exist within a particular distance. The authors evaluate their system on tweets and show that it is one of the state-of-the-art approaches in event detection in microblogs.

Another similar work by Aggarwal and Subbian [3] shows that it is possible to detect events in microblogs through clustering. By selecting a fixed number to represent the total number of clusters, the authors rely on cluster summaries to reduce the number of comparisons needed to cluster documents. The contribution of this work is in a novel similarity score that leverages the graph-based structure of Twitter to create a metric that supports content-based similarity. By considering bursty clusters as events, the proposed technique detects events by following the growth rate of clusters.

The methodology of Weng et al. [62] views the statistics of terms, then transforms them into wavelets that can be used to compute the cross-correlation of each term. This technique considers the changes in the usage of a term over time by mapping those changes into the cross-correlations. Given a set of terms, the proposed technique constructs a graph of many correlation values to those terms. Then, the technique splits this graph to construct several clusters of terms that discuss similar events. The problems with this

approach is that any slight parameter tuning causes significant changes in the effectiveness of the approach. Which means that the technique is quite sensitive to parameter tuning.

In a similar manner, Becker et al. [9] proposed a clustering technique that was proposed in the TDT task on microblog data. The approach then deploys a manually trained classifier to discover features like hashtags and retweets. Such features could be later used in detecting event clusters.

One of the recent works by Parikh et al.[43] aimed at detecting events from tweets using a different approach. Instead of relying on clustering of full tweets, they look at event representative keywords that consist of bigrams. Such terms are selected based on content and pattern similarity scores, which are then used to cluster those terms. By applying a hierarchical clustering approach, the proposed technique favors content similarity of terms and gives it higher weights when compared to apperance similarity patterns. Then, the hierarchical clustering technique is applied with a threshold to get the final number of clusters. The output of the technique consists of a set of ordered clusters in decreasing order. Clusters that contain the largest number of keywords appear at the top of the final list[43].

The study of Alsaedi and Burnap [6] is the first on event detection in Arabic tweets, with focus on detecting events in Abu Dhabi. Around 1M Arabic tweets were collected and labeled by 3 annotators, however, the dataset is not made publicly available for research. Additionally, the dataset is restricted to events in Abu Dhabi, which introduces a bias towards types of events that happen in that location.

2.3 Test Collections for Event Detection in Microblogs

To build a test collection for event detection in Twitter, it is important to look at existing corpora that serve event detection and study their characteristics. Doing that will aid in identifying the usage of such corpora and suitability for the purpose of evaluating and analyzing event detection.

The first event detection collection in Twitter was produced by Becker et al. [9]. The problem with the collection is that it only contained tweets that were posted by users in New York. This issue causes a bias in the type of events that could be extracted from the collection due to geographical restrictions. Furthermore, the number of documents in the collection itself is small, as it contains around 2.6 million tweets, which may not be sufficient for event detection. The authors made their data collection publicly available for research purposes.

The collection that Petrovic et al. [45] focused on the task of First Story Detection. The constructed collection in this work is huge and consists of 50 million tweets that were collected between July 2011 and mid-September 2011. However, the authors identified 27 events only, which means that it is difficult to use the collection for conducting large-scale tests and comparisons. Furthermore, given a small number of events might cause misleading results when systems that reuse this collection fail to detect those events.

In a similar manner to [9], the authors made their data collection publicly available for research purposes. To address the issues with the test collection in [45], Petrovic et al.[46] tried to follow an approach similar to the one that NIST follows in TREC. The methodology relies on expert annotators that read descriptions about events and use event-related keywords to search for relevant documents. Although this method seems promising, it is still expensive because it requires expert annotators in event detection. Moreover, the methodology does not scale well when the size of the collection exceeds a particular limit. Hence, it is better to consider different inexpensive approaches like crowdsouring to get the relevance judgments.

Another relatively small test collection was constructed by Tsolmon and Lee [57]. The authors collected tweets from November 2010 until March 2011. The collection consists of a total of 683 K tweets that revolve around four major events. Apparent issues with this collection is that it is too small to actually perform event detection. Not to mention that the collected tweets were in Korean only, so this makes the chosen events biased to Korean tweets only. Furthermore, the authors did not release their test collection for

research, which further supports the fact that the collection is clearly not suitable for event detection in twitter.

On a larger scale, McMinn et al. [35] built a publicly available test collection for evaluating event detection. The authors crawled around 120M English tweets, covering more than 500 events identified using automatic and manual ways, and collected labels for over 150K tweets. We consider their approach to be a good starting point to our work as it was based on similar goals to ours. However, we followed a slightly different approach to construct *EveTAR* by using a manual method to identify events. On average,*EveTAR* has more tweets per event when compared to their collection. Additionally, we prepared our test collection to be generic enough to support additional tasks like ad-hoc search.

One evident trait that we found in all the literature that discusses event detection in microblogs was in the data. Most of the work that was surveyed reports information about the data that was used in the study. Some researchers go a step further and offer their data sets and relevance judgment labels for future research. Hence, we make the distinction between the notion of a *data collection* and a *test collection*. A test collection is a collection of tweets and relevance judgments that researchers made available to the public either free or in exchange for money. On the other hand, a data collection is a collection that was reported in the literature but was not made available to the public. Table 2.1 includes information about some of the reported data and test collections in the literature. The range of collection size that we identified ranges between thousands and millions of tweets. As for the availability part, we consider all available collections to be test collections, while all non-available collections to be data collections.

Table 2.1. Information on different data and test collections in microblogs (ED: Event Detection)

Collection size	**Number of events**	**Language**	**Usage**	**Availability**
50M tweets	27 events	English	Compare FSD systems[45]	No
120M tweets	796 events	English	Evaluate ED systems[35]	Yes
65M tweets	1000 events	Dutch	ED with term pivoting[27]	Yes
60M tweets	6 event categories	English	Unsupervised ED & categorization [67]	Yes
7.5M tweets	8 event categories	English	Sub ED [37]	No
135K tweets	28 events	English	ED using word similarity [43]	Yes
More than 1.1M tweets	7 event categories	Arabic	Disruptive ED [6]	No
35M tweets per month	73 K events	English	Large-scale ED [6]	No
683K tweets	4 topics	Korean	ED based on LDA [57]	No
345.1M tweets	883 events	Multilingual	Patterns of emerging events [14]	No
25K tweets	2 events	English	Temporal influence on hot topics[19]	Yes
51M tweets	27 events	English	Simulate scalable ED[34]	No
Over 10 datasets	20 events	English	Temporal event mining[31]	No
31K tweets	961 events	English	Attribute extraction of planned events[59]	No
1.4M tweets	5 events	Arabic	ED with location, time, and text[60]	No
341K tweets	57 events	English	Analyze events with Twitter network[48]	No
12K tweets	3 events	Chinese	Real time ED[17]	No
More than 53.4M tweets	2 events	English	ED using graphical model[68]	No
12K tweets	3 events	Chinese	Real time ED[17]	No
More than 60M tweets	1049 events	English	Large-scale credibility detection[38]	Yes

CHAPTER 3. BUILDING THE TEST COLLECTION

Recall that in Chapter 1.3, we briefly introduced the five main steps in the process of building *EveTAR* in Figure 1.2. In this Chapter, we explain the details of those steps in three main stages, which are given in Figure 3.1. The first stage is collecting Arabic tweets to build the Arabic test collection 3.1. The output of this stage is a stream of tweets that is used to identify events in the second stage 3.2. The result of identifying events from the stream of tweets is a list of events and potentially event-relevant tweets. Then, those events and tweets are used in the third stage to obtain relevance judgments through crowdsourcing. The final output of this stage is the relevance judgments for each event in the data collection.

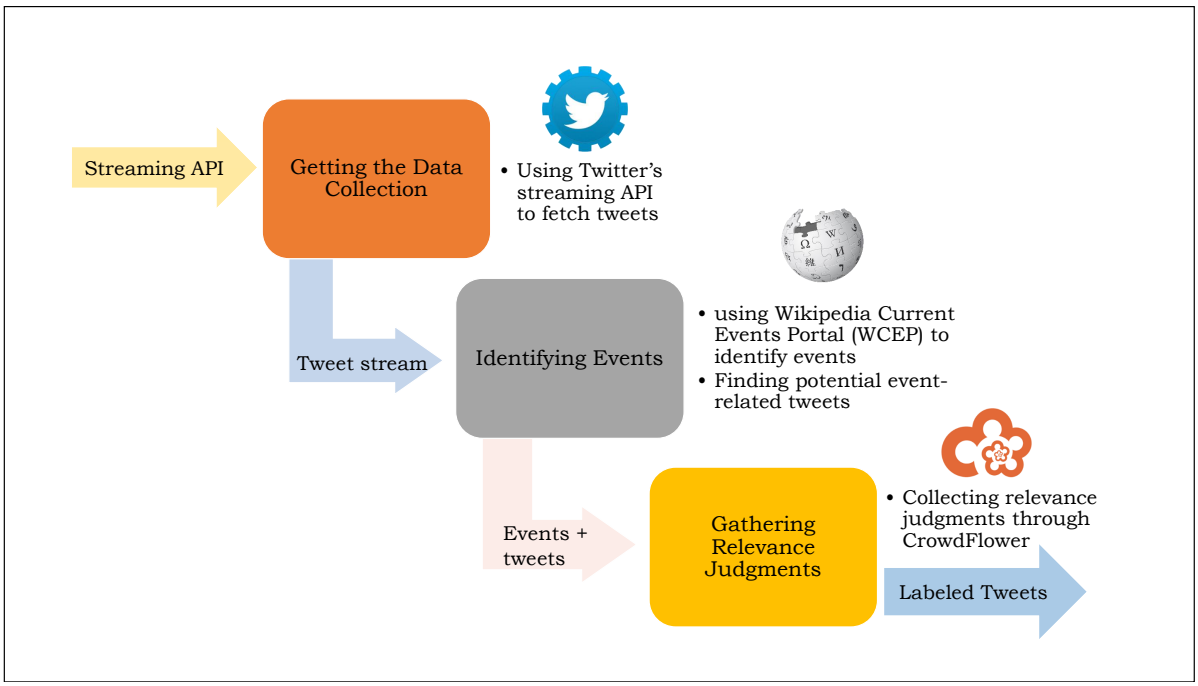


Figure 3.1. The three main stages of building a test collection for event detection.

3.1 Tweet Data Collection

The data collection that was used to construct *EveTAR* was obtained through Twitter's streaming API[1]. The tweets were collected using a pre-filtering sampling technique using the top N most frequent Arabic words. The most frequent Arabic words were extracted from a previously crawled collection by taking the most frequent non-processed tokens. The list of tokens was processed to remove any special characters like *,', and ".

The computed frequency of tokens is the document frequency (i.e, the total number of tweets that contain the token). In the tracking process, the maximum number of allowed terms to track is 400. For the full list of terms that were used in the sampling process, please refer to the Appendix A. To ensure the coverage of the collection for a full month, tweets were gathered three days before the month of January 2015 and two days after it. The reason behind including additional days on the intended period is to ensure that events were discussed beyond the days that they were reported on.

3.2 Identifying Events

Instead of randomly producing a list of events that might not be relevant to our data collection, we manually constructed a list of events after careful investigation of the circumstances that surround Arabic microblog posts. When McMinn et al. [35] built their test collection, they followed two approaches to create candidate events. The first approach relied on automatic event detection techniques like Locality Sensitive Hashing (LSH) and Cluster Summarization (CS). While the second approach leveraged Wikipedia's Current Events Portal (WCEP) to obtain a predefined list of candidate events. The goal behind following multiple approaches was to obtain a pool of candidate events for building the test collection. Our approach in identifying candidate events was adopted from McMinn et al. [35]. The difference is that instead of using automatic event detection techniques, we opted to use the WCEP approach with a few modifications to obtain a list of can-

[1] https://dev.twitter.com/streaming/overview

didate events. In the next section, we describe the details of using WCEP to identify events from our data collection. Next, we discuss the filtering process that was adopted to select candidate events with the aid of Twitter's advance search tool.

3.2.1 Wikipedia Current Events Portal (WCEP)

To help users gain access to current events from all around the world, Wikipedia established a page called the Wikipedia Current Events Portal[2]. The portal shows a list of current events in a particular month, along with a category type, a short description that explains the event and a reference hyperlink to a news article (or more) that discuss the event. The example in Figure 3.2 is an event from our list of events that will be used as a running example throughout this chapter. This event representation is sufficient to evaluate event detection systems that represent events by any combination of date/time, location, and set of keywords.

ID	E12
Title	Discovery of tomb of Egyptian queen Khentakawess III
Date	January 04, 2015
Location	Abusir, Egypt
Category	Arts and Culture
Reference	http://cnn.it/1O6grQK
Keywords	Khentakawess, Egyptian queen, archaeologist
Description	An archeological team from Czech discovered the tomb of an Egyptian queen named Khentakawess III who lived during the fifth dynasty.

Figure 3.2. The representation of the event of queen Khentakawess III as collected from WCEP.

[2] https://en.wikipedia.org/wiki/Portal:Current_events

Figure 3.3 shows a snippet from the news article in the reference field that discusses the event. The article in this example is in English but there is an Arabic version of it with fairly similar content.

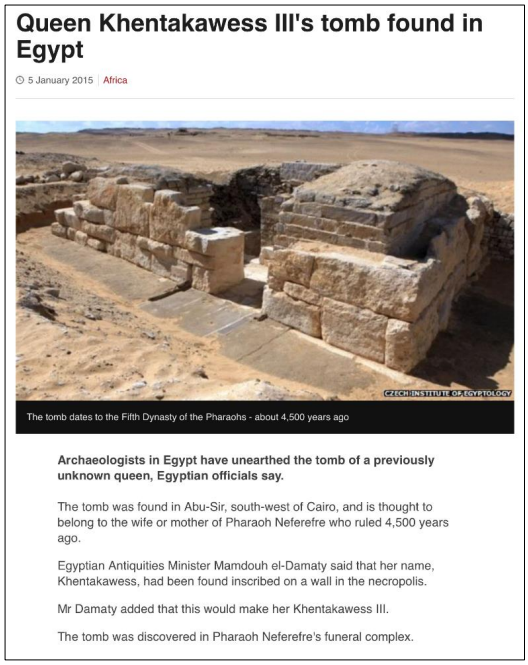


Figure 3.3. The news article of queen Khentakawess III tomb discovery from BBC news website.

Since we are working with Arabic tweets, we found a similar page to the English WCEP in Arabic[3]. Figure 3.4 shows the Arabic WCEP for our desired time period, which is the month of January 2015. Unfortunately, the Arabic WCEP is not as rich as the English counterpart. We believe that this might be due to the lack of dedicated editors for the Arabic WCEP. Figure 3.5 clearly illustrates the lack of events when compared with the Arabic version in Figure 3.4. For a single day in January, the English WCEP documents more events than the entire Arabic WCEP for a period of a full month. Thus, we could not rely on it as a main source for events. To solve this issue, we relied on both the English WCEP and the Arabic one to construct a list of events from both sources.

[3]https://ar.wikipedia.org/wiki/بوابة:أحداث جارية

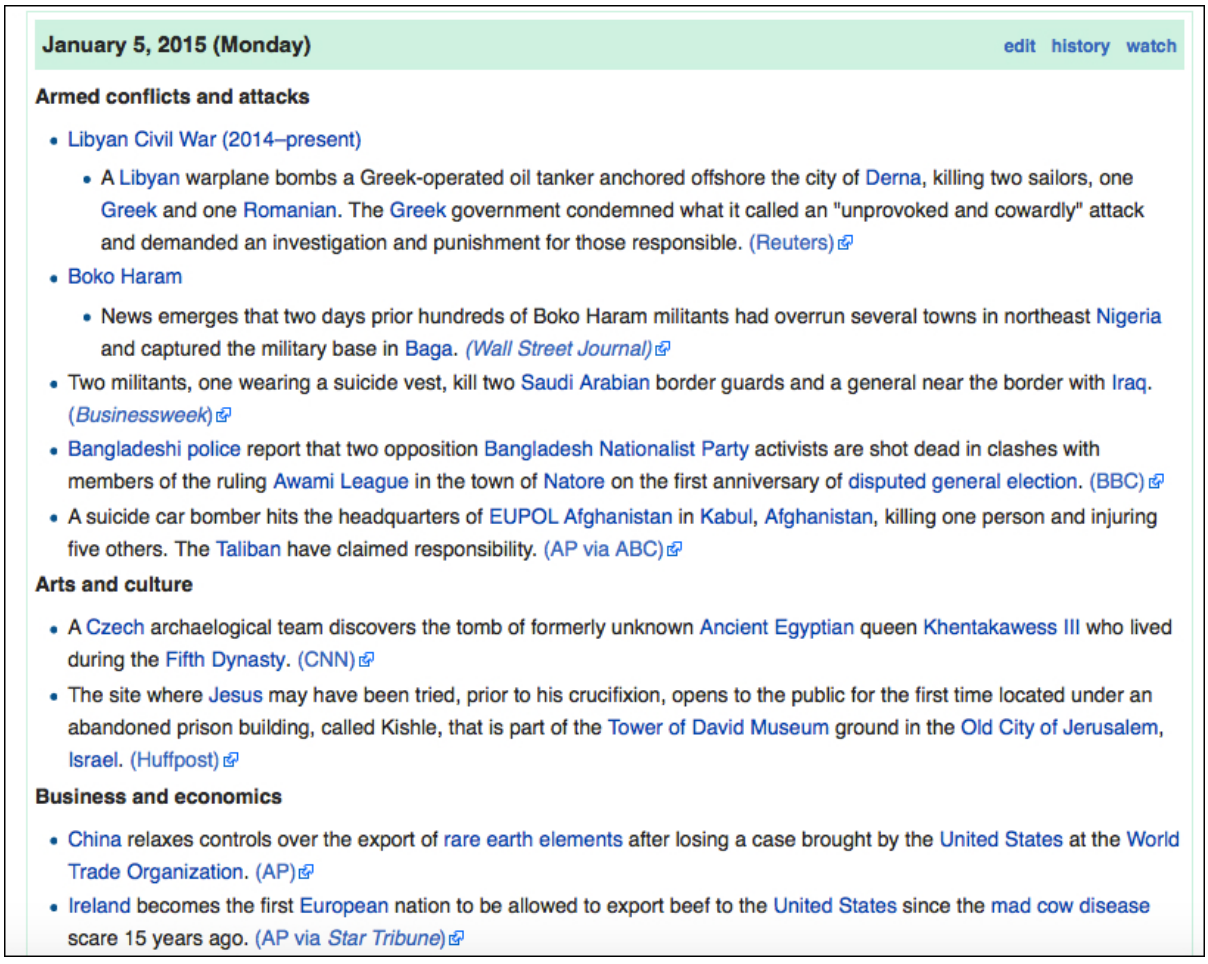
To do this, we had to translate the events from English to Arabic. This process was done manually and resulted in a list of 357 potential events. We then applied our significance criteria over two phases. In the first, we only kept events for which we found at least one online *Arabic* news article discussing the event; only 71 events satisfied that condition.

Figure 3.4. Wikipedia Current Events Portal page in Arabic for the month of January 2015.

3.2.2 Selecting Candidate Events

Combining the outcomes of both the English and Arabic WCEP was not a straight forward task. After examining both WCEP pages, we discovered that the events in the English WCEP are interpreted differently from the Arabic version. For instance, the event of discovering the ancient Egyptian tomb of queen Khentakawess III was reported on the 5th of January in Figure 3.5. However, the Arabic version of WCEP in Figure 3.4 reports the same event on the 4th of January. Hence, we had to revise our initial list

of events and improve it to compensate for those inconsistencies. As an initial step, we considered the Arabic WCEP our main source of events, so we reported the date of the event in the example as the 4th of January. The reason behind this choice is because we were dealing with Arabic tweets, so it was natural to rely on an Arabic source to obtain information about events. The second step in the selection process involved checking if there are sufficient Arabic tweets that discuss the events of the Arabic and English WCEP. Performing this checking process will help in filtering out insignificant events that might not be important for the test collection. For instance, the English WCEP reports an event on the 4th of January about the collapse of a building in Nairobi. While the event seems catastrophic, there were no actual Arabic tweets that discuss the incident. Hence, we considered this event *insignificant* (i.e, it was not discussed by a sufficient number of



January 5, 2015 (Monday) edit history watch

Armed conflicts and attacks

- Libyan Civil War (2014–present)
  - A Libyan warplane bombs a Greek-operated oil tanker anchored offshore the city of Derna, killing two sailors, one Greek and one Romanian. The Greek government condemned what it called an "unprovoked and cowardly" attack and demanded an investigation and punishment for those responsible. (Reuters)
- Boko Haram
  - News emerges that two days prior hundreds of Boko Haram militants had overrun several towns in northeast Nigeria and captured the military base in Baga. *(Wall Street Journal)*
- Two militants, one wearing a suicide vest, kill two Saudi Arabian border guards and a general near the border with Iraq. (*Businessweek*)
- Bangladeshi police report that two opposition Bangladesh Nationalist Party activists are shot dead in clashes with members of the ruling Awami League in the town of Natore on the first anniversary of disputed general election. (BBC)
- A suicide car bomber hits the headquarters of EUPOL Afghanistan in Kabul, Afghanistan, killing one person and injuring five others. The Taliban have claimed responsibility. (AP via ABC)

Arts and culture

- A Czech archaelogical team discovers the tomb of formerly unknown Ancient Egyptian queen Khentakawess III who lived during the Fifth Dynasty. (CNN)
- The site where Jesus may have been tried, prior to his crucifixion, opens to the public for the first time located under an abandoned prison building, called Kishle, that is part of the Tower of David Museum ground in the Old City of Jerusalem, Israel. (Huffpost)

Business and economics

- China relaxes controls over the export of rare earth elements after losing a case brought by the United States at the World Trade Organization. (AP)
- Ireland becomes the first European nation to be allowed to export beef to the United States since the mad cow disease scare 15 years ago. (AP via *Star Tribune*)

Figure 3.5. Wikipedia Current Events Portal page in English for the 5th of January 2015.

Arabic tweets on Twitter). More details about the process of filtering insignificant events are given in the upcoming section.

Twitter Advanced Search

The main idea behind selecting candidate events lies behind their significance. If many people discuss an event then it is significant. However, how many tweets is enough to say that a tweet is significant? Is it enough to find a single news article that discusses the event to say that it is significant? To answer these questions, we had to experiment with the events from WCEP and identify their importance. Before looking at our data collection, we decided to use Twitter's Advanced Search Tool [4] and test the initial list of candidate events. To do that, we manually constructed 6 simple queries per event and used them to search for tweets. In our running example about the discovery of the tomb of queen Khentakawess in Egypt, we constructed the following set of Arabic queries:

1. ملكة فرعونية
2. خنتكاوس الثالثة
3. منطقة أبو صير
4. الملكة خنتكاوس
5. ملكة أبو صير
6. علماء آثار

The queries listed above were used along with our own criteria to verify if an event is discussed by users on Twitter (i.e, the event is significant). The criteria for the importance of an event are given as follows:

- At least 20 different Arabic tweets discuss the event on Twitter. We chose this number because the work done by McMinn et al. [35] established a minimum of 30 tweets per event. Yet, we found that this number was good as a minimum for

[4] https://twitter.com/search-advanced?lang=en&lang=en

English tweets. Since our collection consists of Arabic tweets, we had to account for the difference in volume between English and Arabic tweets. So we chose a minimum of 20 tweets after several experiments to determine a suitable minimum value.

  - Duplicate tweets are not included in the 20 tweet count. This criteria was enforced mainly because we noticed a huge amount of duplicate tweets in Twitter. The importance of an event should not be solely based on duplicate tweets. Thus, if an event is mentioned by a very small number of non-duplicate tweets, then it is not considered significant.

- The search period on Twitter is two days before publishing the event on WCEP and 2 days after (including the day of publishing the event). McMinn et al. [35] chose a filtering period of one day before the event and one day after the event. In the case of Arabic events, we had to alter this choice to account for the fact that event propagation on the Arabic side of Twitter is different from the English side. Some events might be reported earlier or later depending on the nature of the event. Thus, the new filtering period was expanded to two days instead of one.

The process of applying the above filtering criteria on the event of queen Khentakawess using Twitter's Advanced Search tool is shown in Figure 3.6. The interface of Twitter's Advanced Search tool allows users to specify the filtering period as well. Since the event of queen Khentakawess happened on the 4th of January, the filtering period was set to the 2nd of January as a starting date and the 5th of January as an ending date. Due to space limitations, we chose to show a subset of the actual Twitter Advanced search fields in Figure 3.6. A sample of the output from the search process is given in Figure 3.7, where the tweet translates to "Khentakawess the third, a new pharaoh queen". The same process was applied to the 71 events that were initially identified from WCEP, which means that for each event, 6 queries were constructed and used to search Twitter for at

least 20 tweets that talk about the event. The result of this rigorous manual process was a list of 66 events for the month of January 2015.

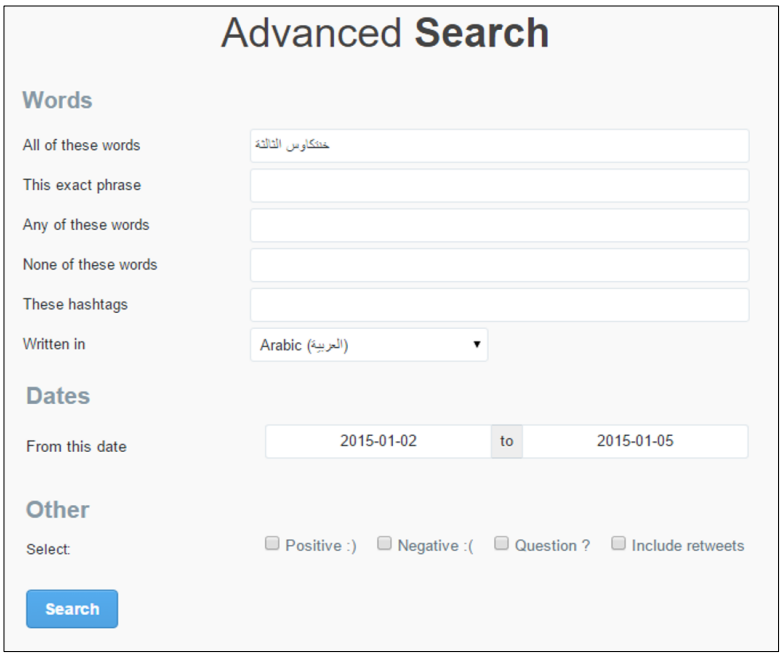


Figure 3.6. Twitter Advanced Search Tool interface using the second query from the event of queen Khentakawess III

In designing *EveTAR*, we elected to enrich the event representation in Figure 3.2 by adding a list of tweets related (or relevant) to each event. That serves two purposes; first, it helps evaluate several event detection systems that represent an event by a list of tweets, and second, it enables the evaluation of other types of retrieval systems such as ad-hoc search or filtering systems that rely on producing lists of tweets per topic.

We obtained those tweets over two main steps. We first extracted a list of potentially-relevant tweets for each event from our dataset, then used crowdsourcing to obtain relevance judgments on them; both are described in the following sections.

Figure 3.7. An example of a relevant tweet from the event of queen Khentakawess via Twitter's Advanced Search Tool

3.3 Gathering Relevance Judgments

Before obtaining relevance judgments, it is essential to obtain all the potential event-related tweets that can be later judged for their relevance. To achieve this goal, we used the list of 66 events obtained from WCEP and the local data collection. The idea is that events can be used to create keywords for searching the collection. The resulting tweets obtained from using such keywords could potentially be event-related, which qualifies them for the step of gathering relevance judgments. An integral part of any test collection is the *relevance judgments*, which is a set of labels that indicate if a data unit is relevant to a certain information retrieval topic or not. In the case of our collection, the relevance judgment labels should indicate if a tweet discusses a given event or not. Since we obtained about 626,247 tweets for all the events, it would be extremely difficult and time consuming to generate relevance judgments using conventional methods. A typical way of getting relevance judgments would be to ask a few volunteers or hire people to read about each event, then label all the tweets associated to that event as either relevant

or not. In this case, getting a single label from one user is not enough to judge the tweet accurately. In most studies, at least three labels are required to cast a judgment on a data item, which means that at least three users must read more than 626,247 tweets and label them accordingly. Such a requirement is impossible to achieve with the manual labor of a few volunteers, which is why we had to resort to crowdsourcing. The upcoming section explains how we gathered potential event-related tweets by searching the tweet data collection 3.3.1. Then, in section 3.3.2 we introduce CrowdFlower; the crowdsourcing platform of our choice, and explain in detail how we used it to obtain relevance judgments.

3.3.1 Tweet Collection Search

To obtain event-related tweets, we had to apply the event verification criteria that was introduced in section 3.2.2 on our local data collection to get tweets. However, searching our collection was not as straight forward as searching Twitter with the Advanced Search tool. The raw collection had to be preprocessed and prepared for searching. To achieve this, we used Lucene Java Library [33] to build an index from our collection to speed the search process. Then, we built an interface to facilitate querying and to simulate Twitter's Advanced Search tool. The major difference between the search approach on Twitter and on our local collection is in the queries. Previously, we constructed around 6 simple queries per event. Those queries were generally obtained from reading articles about the event or from WCEP event description. We discovered that querying our collection locally with the same queries is not sufficient to obtain reasonable results. Therefore, we used Lucene query syntax [33] to modify our queries and prepare them for search. The most useful syntax rules that we applied were the quotation marks (“ ”) for phrase queries, the distance sign ($\sim$) for proximity queries, and the minus sign ($-$) for term exclusion. An example of Lucene queries for the event of queen Khentakawess III is

given as follows: "ملكة أبو صير"4 ~ "الملكة ختكاوس" "منطقة أبو صير" "ختكاوس الثالثة" ملكة فرعونية "علماء آثار"

The proximity symbol in the query "ملكة أبو صير"4 ~ means that the terms must be within the distance of 4 words from each other. The search result of Lucene queries in the previous example are performed using the OR operation. In other words, when queries are written like the example above, they are always ORed together, unless otherwise specified. The interface used for searching our local collection is given in Figure 3.8. The additional criteria that we had to set in this search process was the maximum number of tweets. In Twitter Advanced Search tool, there was no need to specify a value for the maximum number of results, but in the case of the collection search interface, a maximum value must be specified. To ensure maximum coverage of tweets, we set the default value to be 10000. Some events might not be covered by 10000 tweets like the one in Figure 3.8, which is totally fine as long as the event is covered by more than 20 non-duplicate tweets. The result of this final search process was a list of 66 events, with a total of 626,247 tweets. The full list of events can be found in Appendix B.

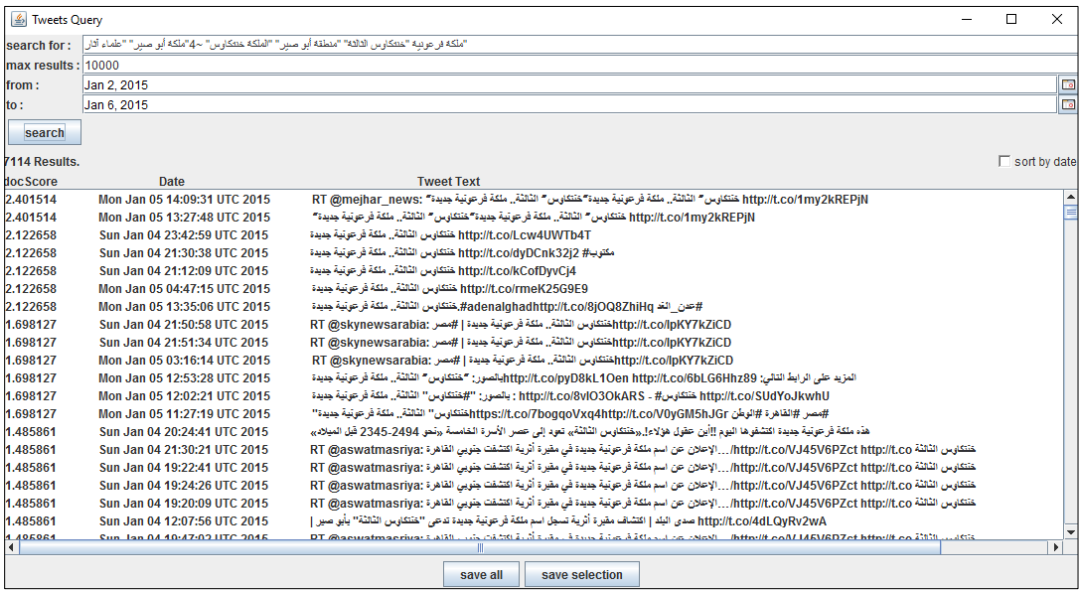


Figure 3.8. Collection Search interface using Lucene 4.0.7 text search library with the event of queen Khentakawess as an example

3.3.2 CrowdFlower Labeling Job

Our study is not the first one that used crowdsourcing efforts to generate relevance judgments. In fact, McMinn et al. [35] relied on Amazon Mechanical Turk (AMT) to obtain their relevance judgments. Since our data collection is in Arabic, we looked at what AMT had to offer in terms of Arabic support. We discovered that AMT has a limited amount of channels (countries) that labeling tasks could be published to. So, we couldn't specify target countries where the dominant language is Arabic (e.g, Gulf countries). Therefore, we decided to go with another crowdsourcing platform known as CrowdFlower[5]. Similar to AMT, CrowdFlower is a crowdsourcing platform that connects customers with workers and facilitates the process of obtaining relevance judgments from workers. To begin working with CrowdFlower, customers are asked to create a labeling job by uploading their data and designing the job to match their needs. The labeling job can be thought of as a task that can be customized to match a specific labeling requirement. Once the job is designed and launched, workers (i.e, users that label data) will be able to begin labeling the data. The platform estimates the costs needed for a labeling job based on the number of data items in the job and the effort required to finish labeling. However, the user can control the job payment to some extent and define a payment price for each worker per job page, which typically contains 10 rows (or 10 tweets). When the number of data items to be labeled increases, labeling costs increase as well, so we had to do something about our huge collection of 626,247 tweets. The simplest approach that we followed was the removal of exact duplicates. Upon closer inspection of our collection, we noticed that some events contained a large number of duplicate tweets. This increase might be caused by the huge amount of retweets. Moreover, we observed that some automatic news subscription accounts propagate duplicate tweets from different user accounts. Thus, we processed the collection to remove all the exact

[5] http://www.crowdflower.com/

duplicate tweets. Exact duplicate removal proved to be useful in reducing the number of tweets from 626,247 to 134,069, which were sent for labeling.

3.3.3 Pilot Study

Before launching a full labeling job, it was essential to understand how CrowdFlower can be used for this specific labeling task. Therefore, we conducted a set of pilot studies on a sample subset of our full collection. For the task of creating pilot studies, we launched a total of four labeling jobs on CrowdFlower. Each job corresponded to a particular event, so we used four events in the pilot studies. The setup of each pilot study was conducted as follows: First, we built a common job description for each of the four events. We tried to simplify the job description as much as possible to simplify the task on CrowdFlower workers. The job description consisted of the title of the event, the date of the event, a sentence describing the event, and hyperlink to an external Arabic news article that discusses the event. Second, we created a few test questions for each event to prevent spam workers from degrading the quality of the labeling task. So, each worker would be given a quiz (using the test questions) before beginning the labeling task. Once workers pass the quiz with an accuracy of 80% or above, they can proceed with the labeling task. A label is obtained if three workers agree on the label. For example, in Figure 3.11, if three annotators agree that the tweet is *not relevant* to the event of queen Khentakawess III, then the label will be set to not relevant. Due to our simplistic approach in the job design of the pilot studies, we were able to identify the following issues with the launched jobs:

- The choice of test questions affects the quality of the obtained labels. After trying different types of test questions, we discovered that it is best to keep them simple and straightforward. Complicating test questions in a particular pilot study caused a high error rate among workers and caused the job cost to increase due to contin-

uous failures. To avoid this problem, we decided to simplify the test questions in all the jobs that were launched for the full evaluation task.

- Most workers will not bother with reading news articles that describe events. Unfortunately, this fact is even true for the entire job description. Some workers join CrowdFlower to earn easy cash, so they do not take the time to read descriptions or provide accurate labels. Hence, instead of just providing a link to a news article for the workers, we designed our full evaluation jobs so that the actual article is included in the job description. This way, workers will not need to click on any external hyperlinks to read because the job description includes all the information that they need.

- Choice of news article that describes the event makes a difference. This issue was discovered while comparing different news articles to include in the job description. We observed that for the same event, some news articles are more informative than others. The richness of the news article contributes to the quality of the obtained labels; since workers are supposed to use them as resources to select relevant labels. Hence, in the full evaluations, we avoided brief news articles and focused on articles from respectable sources like Alarabiya or Aljazeera.

- The overall job design and description affects the quality of the resulting labels. Our simplistic approach towards a seamless and clean job design was useful for the most part. However, it was missing a comprehensive set of examples that illustrates to annotators the difference between relevant and non-relevant tweets. Therefore, in full evaluations, we provided a set of tweet examples that showcase what we consider non-relevant tweets per event.

3.3.4 Final Study

After resolving the issues that were encountered in the pilot studies, we launched a total of 66 jobs for all the events in our collection. The total number of tweets that were labeled without test questions is 134,069 tweets. In reference to our running example of queen Khentakawess, the job design for the event is given in Figure 3.9. The job description shows all the information that workers need to begin labeling in Arabic. The full inline news article that users can click on in the event description is given in Figure 3.10. The design of the job was done using CrowdFlower Markup Language (CLM) and a combination of CSS styling. Thus, the news article in Figure 3.10 was embedded between two collapsible containers. This allows users to show or hide the article whenever they desire.

Figure 3.9. The job description for the event of queen Khentakawess on CrowdFlower crowdsourcing platform

اكتشاف مقبرة أثرية لملكة فرعونية تدعى «خنتكاوس الثالثة»

أعلن الدكتور ممدوح الدماطي، وزير الآثار ، عن اكتشاف مقبرة أثرية تسجل اسم ملكة فرعونية جديدة لم تكن معروفة لعلماء الآثار من قبل، تدعى «خنتكاوس الثالثة»، تعود لعصر الأسرة الخامسة من الدولة القديمة، تم الكشف عنها أثناء أعمال الحفائر التي تجريها بعثة المعهد التشيكي للآثار المصرية بالتعاون مع وزارة الآثار بمنطقة أبو صير الأثرية

وأوضح الوزير أن البعثة نجحت أيضا في العثور على 24 إناءً مصنوع من الحجر الجيري و4 أدوات نحاسية الصنع تمثل جزء من الأثاث الجنائزي لصاحبة المقبرة، بالإضافة إلى الكشف عن كتابات للعمال سُجلت على الجدران الجانبية لحجرة الدفن تحمل بين سطورها اسم وألقاب صاحبة المقبرة وهو زوجة الملك وأم الملك

من جانبه قال ميروسلاف باركا، مدير البعثة التشيكية، أنه من خلال هذا الكشف تمكن فريق العمل من الوصول إلى جزء مجهول من تاريخ الأسرة الخامسة، والذي يكشف أهمية المرأة داخل البلاط الملكي في هذا العصر تحديدا، كما أنه يفتح المجال أمام المزيد من الدراسات المستقبلية للوصول إلى شجرة عائلة هذه الملكة الفرعونية المجهولة

وأضاف أن المقبرة المكتشفة تعد جزءًا من جبانة صغيرة تقع إلي الجنوب الشرقي من المجموعة الهرمية للملك رع نفر إف، والتي نجحت البعثة في الكشف عنها في التسعينيات من القرن الماضي، وهو ما دفع فريق العمل إلى التفكير في أن الملكة خنتكاوس الثالثة قد تكون زوجة رع نفر إف باعتبارها دفنت على مقربة من مجموعته الجنائزية

وقال ياروسير كريتشي، أحد أعضاء فريق العمل، أن لقب أم الملك المسجل داخل المقبرة يحمل درجة كبيرة من الأهمية التاريخية، وتابع: «إذا سلمنا بأن الملكة دفنت في عهد الملك نيوسررع كما هو مرجح من خلال العثور عن طبعة خاتم طيني يحمل اسمه داخل المقبرة، فهذا يدفعنا إلى القول بأن خنتكاوس الثالثة هي أم الملك منكاوحور خليفة نيوسررع على عرش مصر، الأمر الذي قد يكشف مستقبلا عن المزيد من المعلومات المتعلقة بهذا الملك، خاصة أن ما «لدينا عنه يعد نادرا إلى حد كبير

وأشار كمال وحيد، مدير عام آثار الجيزة، إلى أن المقبرة تتشابه في تخطيطها وقياساتها مع مقابر الجبانة بشكل عام، فالجزء العلوي منها مشيد من مصطبة ومقصورة قرابين صغيرة، أما حجرة الدفن فتقع في الجزء السفلي والذي يمكن الوصول إليه من خلال بئر عمودية

يذكر أن منطقة أبو صير الأثرية تعد من المواقع المهمة، حيث تقع على بعد 25 كيلومترا إلى الجنوب الغربي من القاهرة، وتضم المجموعات الهرمية للعديد من ملوك الأسرة الخامسة، كما تضم جبانات ومقابر لعدد من الموظفين والكهنة تؤرخ من عصر الأسرة الثالثة وحتى السادسة، بالإضافة إلى ما تحويه من مقابر يرجع تاريخها إلى العصرين الصاوي والفارسي

Figure 3.10. The news article for the event of queen Khentakawess in the job description

As for the validation test questions, workers were given a simple question in Arabic that asks if the tweet is relevant to the tweet or not. The sample test question in Figure 3.11 illustrates how test questions are given to workers. The answer to the test question is given as two radio buttons that users can click on either one of them (but not both). The first option is yes (i.e, the tweet is relevant to the event), while the second option is no (i.e, the tweet is not relevant to the event). Workers were given a total of 10 test questions as a quiz before labeling and a few additional test questions within each page of the labeling job. This ensures that workers do not randomly select answers without carefully reading the tweet or the description. To showcase the results obtained from the labeling task of each event, Table 3.1 shows examples of relevant tweets from three different events. For each tweet, three annotators agreed that it was relevant to the event after reading the event description and the news article related to the event.

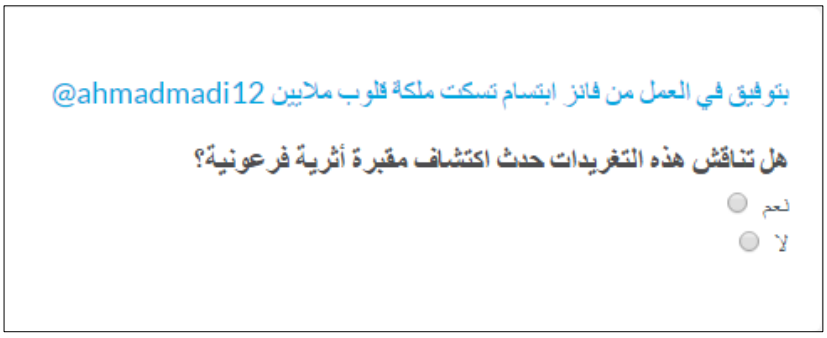


Figure 3.11. Sample test question for the event of queen Khentakawess in the job instructions page

Table 3.1. Examples of tweets from three different events that were labeled by CrowdFlower Workers

Event	Tweet
Tripoli Terrorist bombings in a cafe in Lebanon	لبنان: ٧ قتلى في تفجير انتحاري بمنطقة جبل محسن - جريدة المدينة
Bahrain protests for the detention of opposition leader	#البحرين #المنامة ائتلاف ١٤ فبراير يدعو لتجمّع جماهيري تضامناً مع الشيخ علي سلمان
Launch of Qatar Handball tournament for Men	غدا .. كاظم الساهر يغني في افتتاح كرة اليد ٢٠١٥ في قطر

CHAPTER 4. EVALUATION

To explain the process of analyzing and using the test collection, the first part of this chapter gives specific details on *EveTAR* itself in section 4.1 in terms of the gathered tweets, events, and annotations performed on event-related tweets. The second part of this chapter discusses the usage of the test collection in section 4.2 in terms of applying some existing event detection systems on the collection and evaluating the performance of those systems.

4.1 Test Collection

In Chapter 3, we dedicated section 3.1 for describing the process of gathering the data collection. Here, we provide the statistics associated with the obtained data collection. The information depicted in Table 4.1 shows the total number of tweets in the entire test collection, the duration of the collection, and the disk space. The duration period was extended to 3 days before January 2015 and 2 days after to ensure event coverage. As for the tweets, Table 4.2 provides statistics that show the maximum and minimum number of tweets per event in the entire collection. The first column indicates the original number of tweets before removing duplicates, while the second column shows the decrease after removing exact duplicates. The third column shows the time spent during labeling as it was reported by CrowdFlower in hours. As for the last two rows in Table 4.2, the values show the total and average numbers of all tweets in all events.

Table 4.1. Statistics about the collected tweets that were used to build *EveTAR*

Tweets	**Duration (from - to)**	**Disk Space(GB)**
590,066,789	29/12/2014 - 02/02/2015	240

Table 4.2. Statistics about the number of tweets in our collection before and after removing exact duplicates.

Statistic	Tweets (before processing)	Tweets (after processing)	Labeling time(hours)
Maximum	10000	5767	371
Minimum	400	83	3
Total	626247	135887	3916
Average	9489	2059	59

4.1.1 Events

In section 3.2, we discussed our approach at acquiring candidate events. However, we left out some important information about the nature of the events and the number of tweets found in each event. To get a general overview about the overall tweet distribution across events, Figure 4.2 shows the total number of labeled tweets and events in the collection. Each bar in the figure stands for a single event, which leads to a total of 66 bars for all the events. The number of tweets was sorted to illustrate the distribution across all events. Further inspection of the events that we had lead us to believe that they actually fall into different categories. Using Wikipedia's current events portal, which was discussed in section 3.2.1, we were able to identify 8 different event categories. Actually, the portal includes a wide selection of categories to cover all types of events. However, we identified 8 categories only due to the limited number of events that we had, which is only 66 events. Examples of events from each category are given in Table 4.3. For the full list of events per category and the events English translation, please refer to Appendix B. In Figure 4.1, we show the distribution of tweets across the 8 different categories that we got from Wikipedia's current events portal. The pie-chart shows that most of our events fall into the category of "armed conflicts and attacks", with an staggering 68%. On the other hand, the category with the lowest number of events is the "business and economy" category, with only 1% of the events falling under it. Such observations show that our collection is skewed to a single category, which might be related to the time period of the

collection. We specifically chose the time period beforehand because we knew from news outlets that the month of January was full of interesting events. However, the events were not planned to fall under a certain category, so we believe that this distribution is limited to our collection and the time period that it fell under.

Table 4.3. Examples of events from each category identified in *EveTAR*

Event Category	**Event title**
Armed conflicts and attacks	مقتل جنديين في هجوم لحزب الله على رتل للجيش الاسرائيلي في مزارع شبعا
Business and economy	ليتوانيا تتخلى عن الليتاس وتنضم لليورو
International relations	أمريكا تفرض عقوبات إضافية على كوريا الشمالية بعد الهجوم على سوني
Law and crime	تنفيذ عقوبة الجلد علناً بحق مدون سعودي أدين بتهمة «إهانة الإسلام»
Politics and elections	اشتباك محتجين في البحرين بسبب مواصلة إعتقال زعيم معارض
Sports	فوز استراليا على الكويت ٤ ١ في افتتاح نهائيات آسيا
Disasters and accidents	انتشال مسجل قمرة القيادة من حطام طائرة «ايرآسيا»
Arts and culture	بناء أول كنيسة في إسطنبول منذ قرن

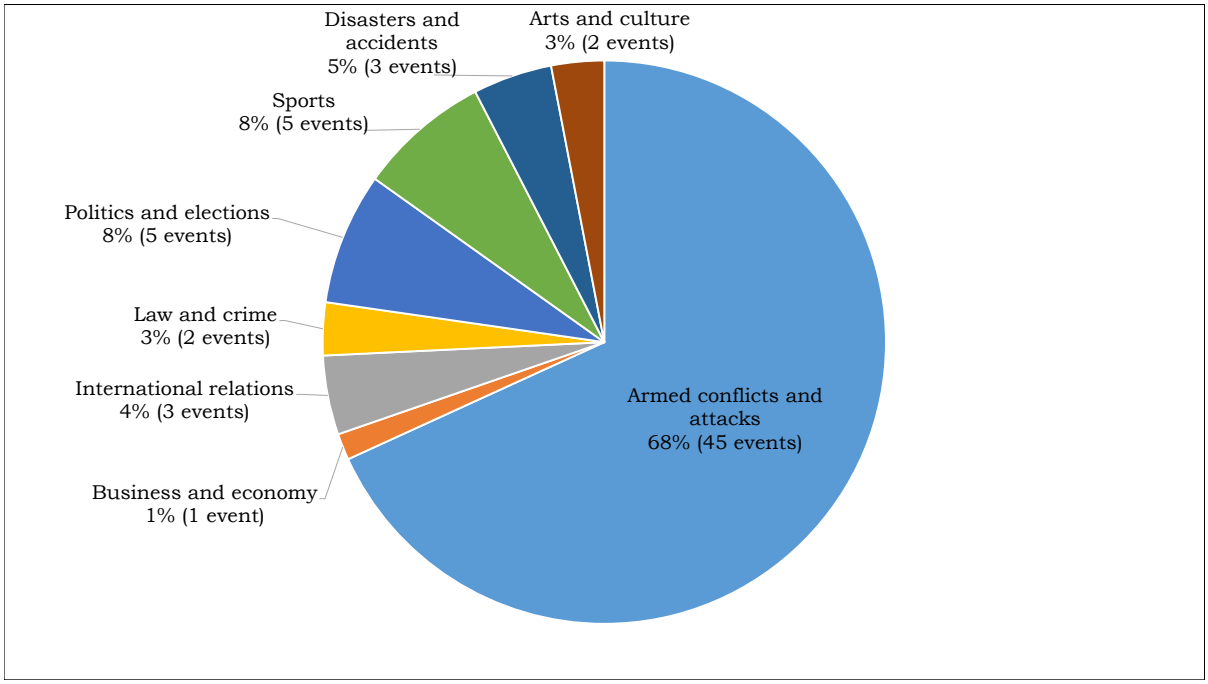


Figure 4.1. The overall event distribution across the 8 categories that were identified from WCEP

4.1.2 Annotations

An integral part of the test collection is the annotations, which are basically the labels that were obtained from CrowdFlower workers for each event in the collection. Recall that in section 3.3.4, annotators were asked to label a tweet as either relevant to a given event or non- relevant. The results that we obtained from the annotators after this process were quite interesting. Initially, we thought that when events had a large number of tweets, then such events would produce a high number of relevant tweets and vice versa. This assumption seemed logical at the time of launching the jobs for labeling. Yet, the results that we obtained were different. The results depicted in Figure 4.2 show a deviation in the total number of tweets and relevant tweets for some events. The most obvious cases are shown in the right-most-event, which is event 62. Notice that the total number of judged tweets is much more than the actual relevant tweets. This means that many of the tweets in that particular event were labeled as "non-relevant". This is mainly caused by the queries that were used to obtain the tweets for this event. In some cases, if queries are not carefully chosen, they produce a huge number of noisy tweets.

The uneven distribution of relevance judgments in events is given in Figure 4.2. The stacked view of the columns that represent events shows a different side of the issue. Notice that the deviations at the top of the stack represent the total number of non-relevant tweets, while the bottom part of the column stack is the relevant tweets. The event with the least amount of relevant tweets is shown in the last column from the right side. The event number 62 on the Figure, which is about Houthi's control over military camp in south Sana'a in Yemen, had 12 relevant tweets only and a total of 3,346 non-relevant tweets. For annotators, this particular event was tricky because of the different noisy discussions that revolve around Yemen but not necessarily about the event itself. On the other hand, event 66 in Figure 4.2 has the highest number of relevant tweets. With a total of 3,619 relevant tweets and 517 non-relevant tweets; the event was discussing the death of the second Japanese hostage by ISIS. Upon further examination of

the annotated event, we noticed that most of tweets were actually discussing the incident. This is perhaps due to the popularity of events that discuss ISIS in the Arab world. The majority of the events shown in Figure 4.2 tend to have a higher number of non-relevant tweets. For instance, event 11 talks about the death of several Houthi's because of a bombing in Dhamar Governorate in Yemen. The total number of relevant tweets is 622, while the total number of non-relevant tweets is 1,227. While the difference between the number of relevant and non-relevant tweets is not large, we noticed a similar pattern in most of the events. The reason behind the increased number of non-relevant tweets is the amount of noise that was present in the tweets. Although exact duplicate tweets were removed from tweets, we did not attempt any further processing to remove tweets that do not belong to events. Regardless of such results, we obtained a total of 51,424 relevant tweets across all events and 82,645 non-relevant tweets.

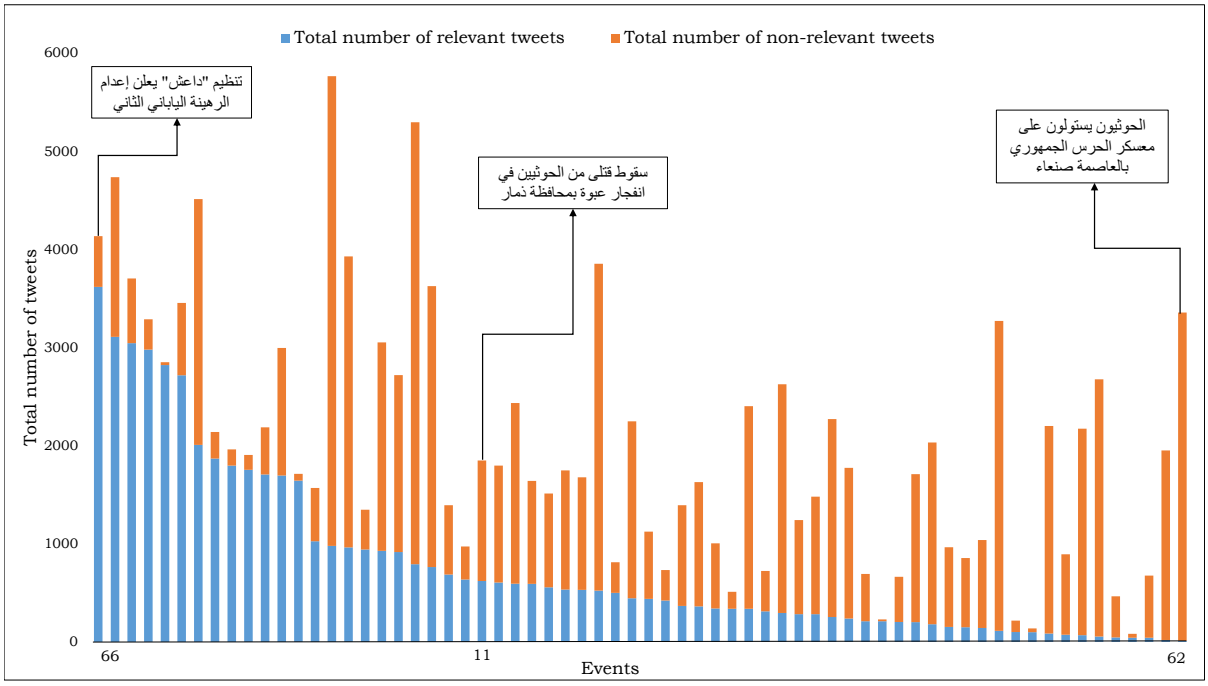


Figure 4.2. The ratio of relevant to non-relevant tweets per event across all events

As a final remark on the distribution of relevance judgments across all events, we show the stacked view distribution in Figure 4.3. The overall number of relevant and non-relevant tweets are given per column in the figure. The top part of the column stack represents the % of non-relevant tweets, while the bottom half depicts the relevant tweets. Figure 4.3 shows that at 50%, the majority of tweets for 22 events are relevant. This indicates that more than half of the judgments obtained for 33% of the events are relevant.

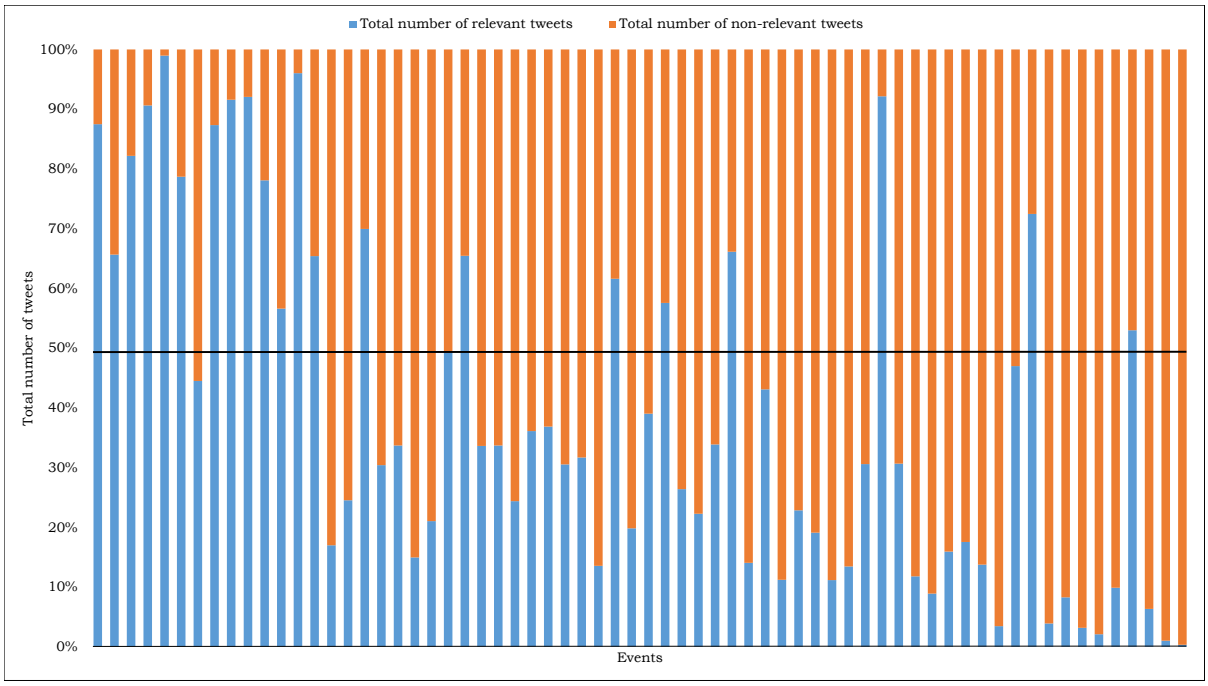


Figure 4.3. Stacked view of the ratio of relevant to non-relevant tweets per event across all events

The relationship between the number of judgments per event and the total time spent judging each event are depicted in Figure 4.4. Outliers in the plot are identified by red circles around the event. The Figure shows that the time spent during a labeling job of an event is not solely related to the number of tweets in that event. Actually, there is a somewhat linear trend showing in Figure 4.4, where the labeling time increases with the number of relevance judgments. However, this is not the case for all events. For instance, the outlier with the highest labeling time is an event took 371 hours to finish labeling. The event is about the recapturing of at least 90% of Koban, Syria by Kurdish

fighters. The total number of judgments in this event is 2,720, which is much lower than the first outlier above the line that has 3,704 judgments. Moreover, the first outlier above the line took 202 hours to finish labeling, while the outlier below the line took 34 hours only to finish labeling 3,854 tweets. This particular event is about a bombing in a Shiite mosque in Shikarpur District of Pakistan. The reason behind such numbers is that the event about Koban was more challenging to CrowdFlower workers when compared to the event of Shikarpur bombings. Workers reported several issues with the test questions of the event about Koban, so more test questions were supplied to speedup the labeling process. CrowdFlower reported that 54 annotators gave this particular event a rating of 3.3 out of 5 for the ease of the job, while the Shikarpur bombings event got a rating of 3.8 out of 5 by 35 annotators. This shows that event difficulty plays a critical role in the labeling time of an event, where difficulty is mostly associated to the clarity of the tweets associated with the event.

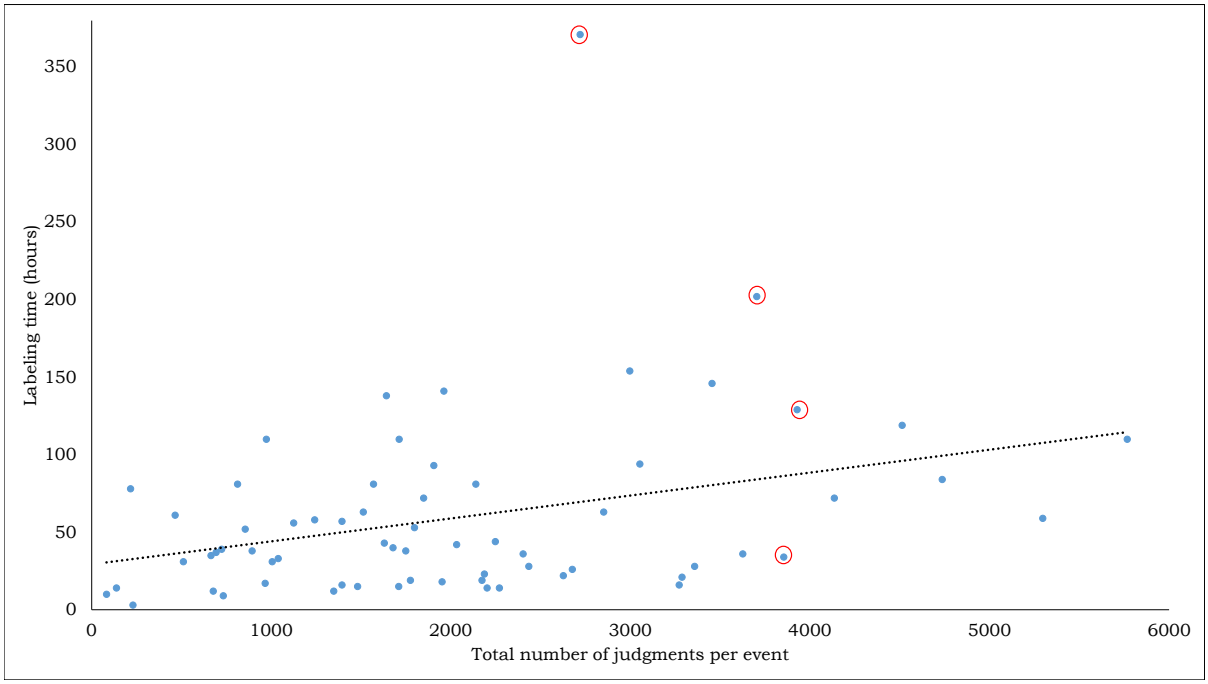


Figure 4.4. Relationship between the total number of judgments submitted to CrowdFlower and labeling time

4.1.3 Qualitative Analysis

Based on section 3.3.3, the annotations that we obtained through CrowdFlower were computed based on the majority vote of three annotators. To get the label for a particular tweet, CrowdFlower allows at most 3 annotators to agree whether a tweet is relevant to an event or not. For a single event, several annotators might completely agree on a label or completely disagree. In some cases, annotators might get confused about the label and raise issues with the event or job design. Hence, we decided to measure the quality of the labels obtained through CrowdFlower by computing the inter-annotator agreement. Since we fixed the number of annotators that agree on a label to be 3, the most appropriate reliability measure is Fleiss' Kappa [54]. Kappa is a statistical measure that computes the degree of reliability between the labels obtained from annotators. The formula to compute Kappa is given in Equation 4.1.

$$K = \frac{P_\circ - P_c}{1 - P_c} \tag{4.1}$$

$P_\circ$ in the equation stands for the proportion of agreements that were observed during the labeling phase, while P_c stands for the proportion of agreements that were obtained by chance [54]. To compute Kappa, we used the Real Statistics tool for Microsoft Excel [1]. According to [54], each value of Kappa tends to fall within a certain category. There are six well-known Kappa categories based on their values, which are given in Table 4.4.

Table 4.4. The six Kappa categories according to the range of Kappa values

Agreement	*Almost perfect*	*Almost perfect*	*Moderate*	*Fair*	*Slight*	*Poor*
Range	1 - 0.81	0.8 - 0.61	0.6 - 0.41	0.4 - 0.21	0.2 - 0.01	≤ 0

[1] http://www.real-statistics.com/reliability/fleiss-kappa/

In Figure 4.6, we show the distribution of events based on the Fleiss' Kappa categories that they belong to. Our collection covers five of the six different Kappa categories, since there are no events that belong to the poor category. Based on the computed Kappa values, 13 events belong to the almost perfect category, 23 events belong to the substantial category, 16 events fall under the moderate category, 11 events are in the fair category, and 3 events belong to the slight category. This categorization shows that across all the events that we have, only 14 events (which belong to the fair and slight category) have poor Kappa values. By examining the event with the highest Kappa value (0.96), we found that it is about the bombing of a Shiite mosque in Shikarpur. The total number of non-relevant tweets for this event is 3,331, while the relevant tweets are 523. Moreover, we found that for this particular event, annotators did not fully agree on the labels of 404 tweets. For example, the following tweet was judged by three annotators for the event of Shikarpur bombings:التكفيريين اليوم فجروا مسجد للشيعة في باكستان و حافلة للشيعة في دمشق ،،، حزب الله فجر سيارتين للصهاينة مع حبة مسك. Two out of the three annotators labeled the tweet as relevant, so there was no full agreement within annotators. We also looked at the event with the lowest Kappa value (0.04), which contains 2,852 tweets. The event is about a strike launched by an Israeli helicopter near Syria. Although the total number of non-relevant tweets in this event is only 29tweets.

The trend line in Figure 4.5 shows that in general, Kappa values tend to decrease when the total number of judged tweets increases. Yet, there are several outliers that defy this trend. This is evident when the trend line slope decreases while events remain above the line. We believe that this is due to the amount of disagreement (or full agreement) present within the labels of each event. The event that we discussed earlier about Shikarpur bombings is shown at the far top of the plot (close to 1). Whereas, the event of the Israeli helicopter strike is shown at the bottom (close to 0). In fact, the event of Israeli helicopter strike is considered an outlier because it has a large number of relevance judgments, yet a small Kappa value.

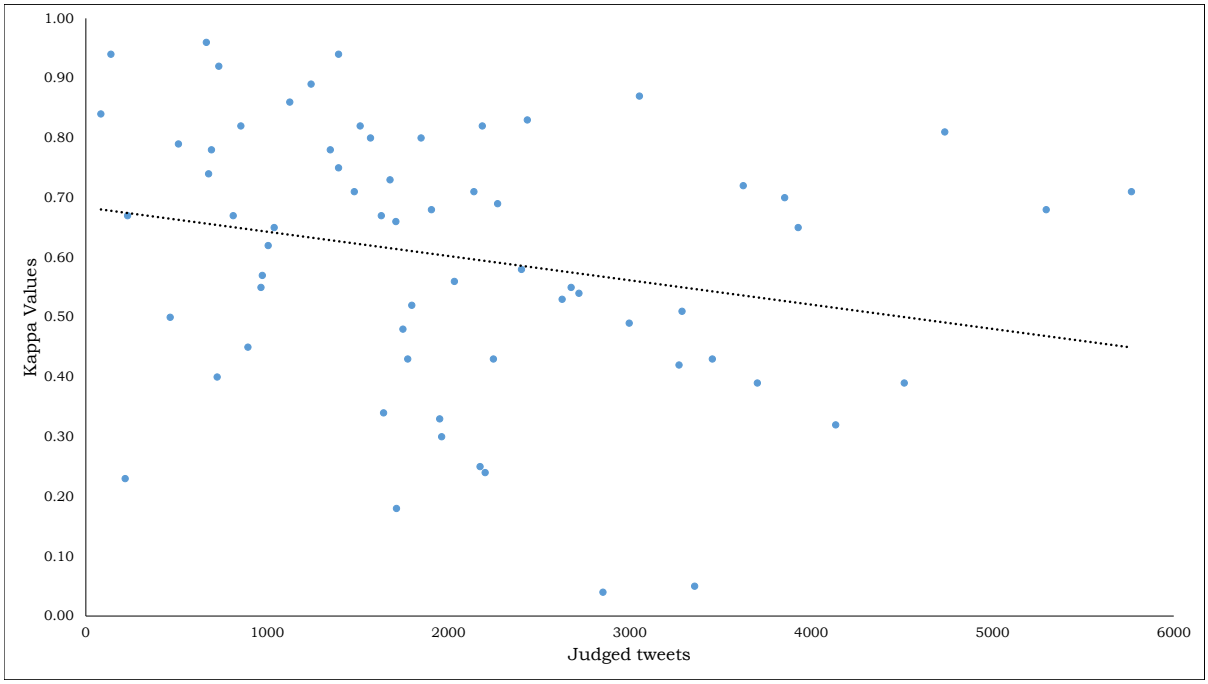


Figure 4.5. The relationship between Kappa values and the total number of judged tweets per event

In addition to Kappa, we computed the overall annotator agreement per tweet using Equation 4.2. Since we have two possible labels (either a tweet is relevant to an event or not relevant) and a total of three annotators. The equation represents the agreement per tweet for a single event. Averaging all of the agreement values per tweet will give us the agreement per event.

$$\text{Annotator\%agreement(per tweet)} = \frac{\text{Max(Number R and NR tweets)}}{\text{Total number of annotators}} \tag{4.2}$$

The last statistic that we computed on our labels is the confidence value. Since we relied on CrowdFlower to get the annotations, we used the results report that the platform generates to obtain the trust value per annotator. According to CrowdFlower, each annotator has a certain trust value that lies between 1 and 0. The platform does not declare how this trust value is computed, but we know that based on a labeling job setting, we can choose to allow annotators with a certain trust to participate in labeling. Since CrowdFlower offers 3 levels to control the speed-quality ratio, we adjusted the

settings of each labeling job at a level 2. We discovered that this setting allows jobs to finish at a moderate speed with relatively good quality. Level 1 disregards quality and favors speed, while level 3 favors quality at the price of very slow speed. If we were working with English tweets, level 3 would be a reasonable choice. However, Arabic tweets require annotators to be familiar with the language. Since we cannot guarantee that CrowdFlower's top rated annotators will be familiar with Arabic, we decided to go with level 2. To compute the confidence in a certain label given by an annotator, we use Equation[2] 4.3.

$$\text{Tweet Trust Score} = \frac{\max(\sum_{i=1}^{r} trust_i, \sum_{i=1}^{n} trust_i)}{\sum_{i=1}^{n+r} trust_i} \tag{4.3}$$

where r and n are the number of annotators labeling the tweet as relevant or non-relevant respectively, and $trust_i$ is the trust score for an annotator. Averaging this overall trust score over all tweets and all events results in an average quality score of 0.94 out of 1. Getting the average of all tweet trust score values gives us the trust score per event.

The results shown in Table 4.5 summarize the findings that we obtained from the annotations. We only report the maximum, minimum, and average values for all the events. The average Kappa value for all the events is 0.6, which falls in the substantial category. Moreover, more than half of the events got a substantial to an almost perfect agreement. Additionally, eliminating the 3 events with slight agreement among annotators results in an average agreement of 0.62, which is considered *substantial*. Those 3 events had large annotators disagreement due to confusion with other similar events, resulting in very low Kappa. The figure also shows the values of the average trust score per event, which indicates a slight general drop with decreasing Kappa values. Coincidentally, we discovered that computing the tweet trust score for both relevant and non-relevant was similar to the agreement. Hence, we refrained from including duplicate results and presented the agreement, which in our case is equal to the overall tweet trust score of both

[2]The equation is stemming from the one used by CrowdFlower to report confidence in the aggregated label given for a data item, see: http://bit.ly/20NmFkU

relevant and non-relevant tweets. The maximum and minimum values in Table 4.5 are used to illustrate the extreme values that were identified in the 66 events. For a full list of confidence and Kappa statistics per event, please refer to Appendix C at the end of this thesis.

Table 4.5. Kappa and confidence values across all events in the collection

Statistic	Agreement	Overall Confidence	Kappa
Maximum	0.99	0.99	0.96
Minimum	0.81	0.81	0.04
Average	0.94	0.94	0.60

The result combining Fleiss' Kappa categories and confidence values (or tweet trust score) is shown in Figure 4.6. The Figure shows that at the beginning of the substantial category, the confidence values begin to fluctuate and differ from the semi-steadiness in the almost perfect category. By tracing the columns that represent Kappa values and the confidence lines, we noticed that the overall confidence resembles the progress of Kappa values more than the relevance confidence. This is due to the fact that the overall confidence considers both relevant and non-relevant tweets. Therefore, it is more comparable to the Kappa values that were computed from both relevant and non-relevant tweets.

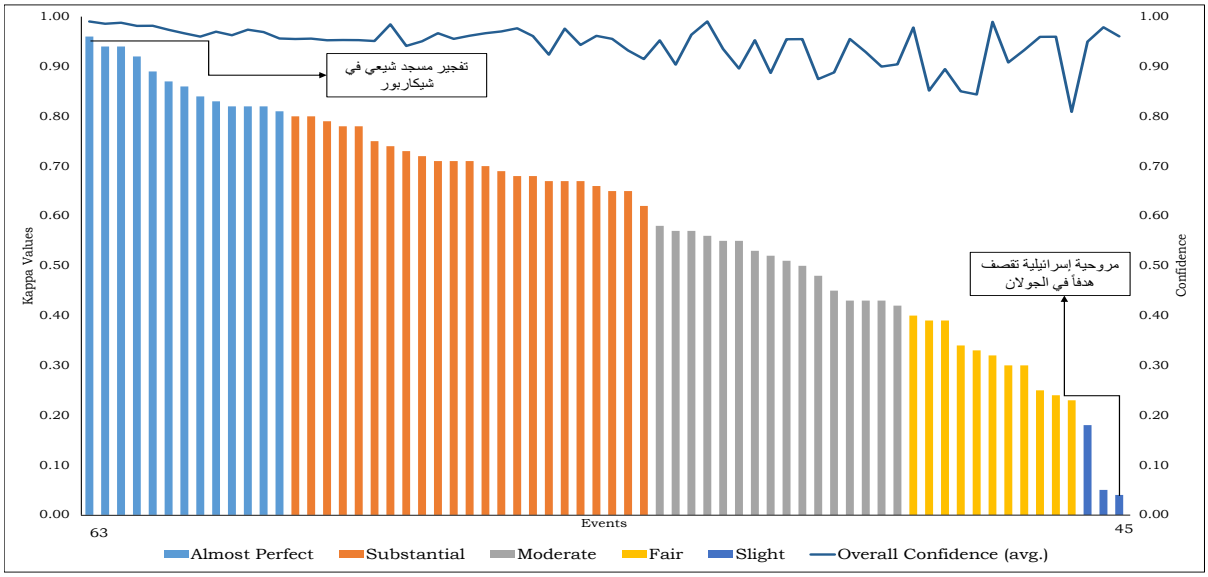


Figure 4.6. Categories of Fleiss' kappa Vs. overall confidence per event across all events

4.1.4 Comparison With Other Test Collections

To compare between our test collection and the existing test collections that we discussed Chapter 2, we present a new version of Table 2.1 that focuses on the available test collections. This time, instead of highlighting the availability, we look at the average number of tweets per event (if it was reported), as shown in Table 4.6. Our test collection is situated at the last row in Table 4.6. On average, our test collection has more tweets per event when compared with [35].

Table 4.6. Information on different data and test collections in microblogs (ED: Event Detection)

Collection size	Number of events	Avg. Number of tweets	Language	Usage
120M tweets	361 events	259 tweets	English	Evaluate ED systems[35]
65M tweets	1000 events	NA	Dutch	ED with term pivoting[27]
60M tweets	6 event categories	2.8 K tweets	English	Unsupervised event extraction and categorization [67]
135K tweets	28 events	NA	English	ED using word similarity [43]
25K tweets	2 events	22K and 3K	English	Temporal influence on hot topics[19]
60M tweets	1049 events	NA	English	Large-scale credibility detection[38]
590M tweets	66 events	2K tweets	Arabic	Event detection & ad-hoc search

4.2 Using *EveTAR*

Since Chapter 3 discussed the process of building a test collection from Arabic tweets for event detection; now, it is relevant to showcase how such a collection can be used to evaluate event detection systems. While the main goal of this research is to build the collection, we believe that it is essential to demonstrate the usage of the collection through different event detection algorithms. The following section demonstrates four different event detection algorithms and their performance when applied to our test collection.

4.2.1 SONDY's Social Analysis Tool

Developing event detection algorithms from scratch to evaluate the test collection is tedious. Therefore, we looked for an open source tool that integrates some state-of-the-art event detection algorithms. Recently, a work by Guille et al.[21] discussed a tool named

SOcial Networks DYnamics (SONDY). The platform is an open source application that allows end-users and researchers to explore and manipulate Twitter's social messages in different ways. Users can benefit from SONDY through four main services: the *data manipulation service*, *topic detection and tracking service*, *network analysis and visualization service*, and the service for *importing algorithms* to the tool. Initially, we were interested in SONDY because it encompasses some event detection algorithms, which are available through the topic detection and tracking service. In brief, the two services that we used allowed us to achieve the following:

- *Data manipulation service*: Through this service, we were able to import our data collection with the data import part to prepare it for event detection algorithms. The preprocessing part offers stemming, lemmatization, tokenization, and time-slice partitioning. In the case of Arabic tweets, we discovered that applying stemming and lemmatization makes the resulting events unreadable. Hence, we only applied tokenization and time-slice partitioning [21]. The service allows the data collection to be filtered by removing stop words, which are configurable. We added a list of common Arabic stop words that were identified from previous studies. The full list of Arabic stop words is given in Appendix A. In addition to stop word removal, the filtering part allows the tweet stream size to be adjusted and re-sized to a specific time window. Both stemming and lemmatization were disabled, while tokenization was set to 1-gram and partitioning was set to 30 minutes. The filtering that was applied to the collection is English, Arabic, and Twitter-related stop word removal. We applied English stop word removal to ensure that no English stop words slip into the event detection algorithms. As for Twitter stop words, they include terms like RT (retweet), http, and follow. Such words do not contribute to the event detection algorithms, thus, they were removed from the message stream.

- *Topic detection and tracking service*: With the aid of this service, different topic and event detection algorithms can be applied to the tweet stream. By selecting

the prepared messages from the data manipulation service, we were able to choose between 7 different algorithms. Some of the offered algorithms were for topic detection, like on-line LDA (Latent Dirichlet Allocation). Therefore, we focused on the event detection algorithms, which are MABED [22], EDCoW [62], Peaky Topics and Persistent Conversations[53]. A detailed overview of the chosen algorithms will be given in the upcoming section. In addition to event detection, SONDY provides an event visualization tool. Each detected event can be viewed on a time line that illustrates the event peaks. The demonstration in Figure 4.7 depicts the application of MABED algorithm on the Arabic data stream. The highlighted event in the Figure matches the running example that we introduced in Chapter 3. The time line in the Figure matches the days in which the highlighted event occurs. Moreover, the tweets associated with each detected event are accessible through the messages view. For the highlighted event of queen Khentakawess in Figure 4.7, the messages view is given in Figure 4.8. The columns in the messages view show the tweet ID, timestamp, and text of all the tweets that belong to the event. As for the event time interval, the view shows a range between the 4th and 6th of January, which is the same period that we reported in our test collection. Moreover, Figure 4.8 shows that MABED identified 671 relevant tweets in the event. In fact, some of those tweets belong to the 284 tweets that crowdsourcing identified as relevant for the same event. Some of the tweets might be non-relevant, yet the algorithm managed to identify the actual event, the exact date, and a portion of the relevant tweets.

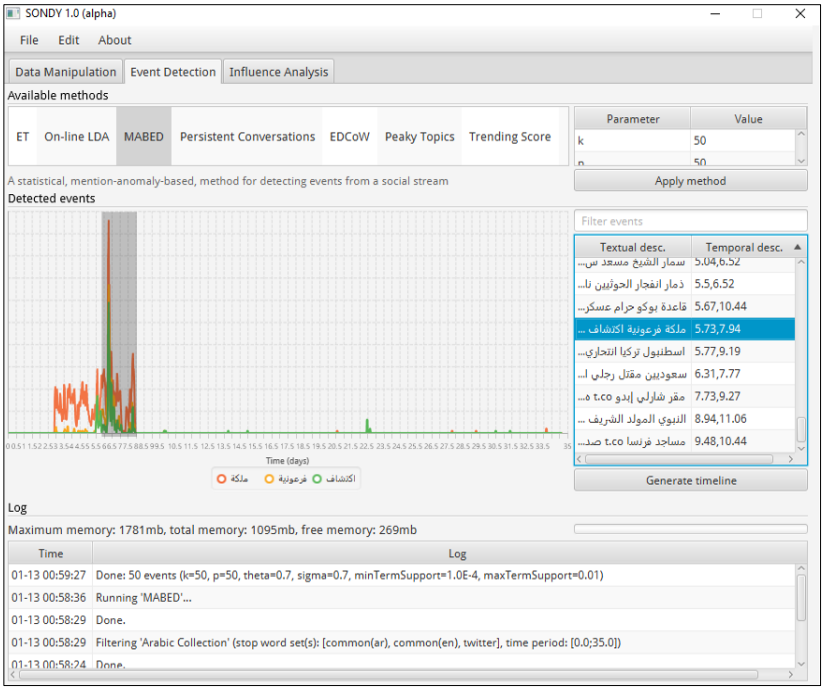


Figure 4.7. SONDY's event detection interface, where MABED is applied to *EveTAR*

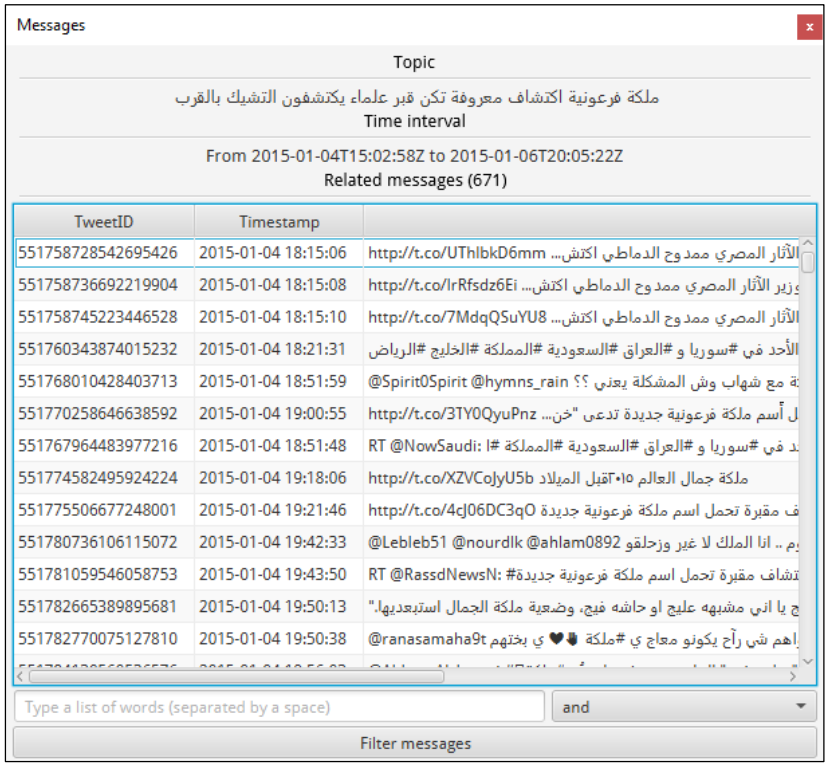


Figure 4.8. Messages view in SONDY that shows all the tweets associated with the detected event

4.2.2 Performance of Event Detection Algorithms

In the upcoming sections, we describe the details of the algorithms used in the evaluation, then we discuss the experiments conducted on our Arabic test collection. Additionally, we explain the inter-annotator agreement that was computed for *EveTAR*, followed by a section about the conducted experiments on other English test collections.

Algorithms

Previously, we mentioned that SONDY offers several algorithms for topic and event detection. For the task of event detection, we explored four different algorithms from SONDY's topic detection and exploration service. The details of each algorithm along with their parameters are given as follows:

1. *EDCoW:* Event Detection with Clustering of Wavelet-based signals (EDCoW) aims at identifying events from a stream of tweets [62]. The technique relies on analyzing wavelet signals of individual words to identify events. For each word, the algorithm builds a signal based on the wavelet analysis of the word's raw frequency. Then, autocorrelation is used to compute each word's bursty energy. Cross-correlation is then computed between pairs of bursty words. By doing that, insignificant words are filtered out from the computations. By the end of this, the remaining bursty words form a cross-correlation table, which is later used to build a modularity-based graph [62]. To detect events, graph-partitioning techniques are applied on the constructed graph of bursty words. Final events are constructed by clustering (grouping) words with similar bursty patterns. To apply EDCoW, the wavelet signal must be built for each individual word. Therefore, the difference between each set of consecutive sample points is set by default to 8 minutes and $\delta = 48$. Such setting allows the algorithm to compute the final signals of individual words as they change within 384 hours (16 days). The maximum term support parameter

was set to 0.01, while the minimum term support was left at 0.0001 [21]. As for the tunable parameter γ, it was set to 5.

2. *Peaky Topics:* To discover temporal patterns from a stream of tweets, peaky topics leverages a normalized version of term frequency to detect momentary terms of interest. The technique works by identifying a list of highly-localized terms or "Peaky Topics" [53]. In other words, peaky topics identifies terms that appear in a fixed time window but do not appear in any other time window. So term frequency is computed over a specific time window and used as a score for all terms within that window. The technique is solely focused on temporal features of terms. Thus, the only parameters that the algorithm requires are the maximum term support at 0.01, and minimum term support, which was set to 0.0001.

3. *Persistent Conversations:* The implementation of persistent conversations is quite similar to peaky topics. In fact, both algorithms were introduced by Shamma et al. [53]. Unlike peaky topics, persistent conversations looks for terms that are popular throughout the entire stream of tweets. The algorithm aims at finding terms that remain important and highly discussed during a longer time period than peaky topics. To score terms, the algorithm uses the scores obtained from peaky topics. A term is ranked based on how longer it remains interesting, so the score is computed by averaging the normalized pre-peak and post-peak term frequency. Similar to peaky topics, the only parameters used in this algorithm are the maximum and minimum term support, which were set to 0.01 and 0.001.

4. *MABED:* Mention-Anomaly-Based Event Detection is a unique technique that has an advantage over all the previously mentioned techniques. Instead of using the temporal characteristics of a tweet stream to identify events, the algorithm uses tweets text only to detect events. By using the power of mentions (dynamic links) that users often use in their tweets, MABED is able to detect important events. Another advantage that the algorithm holds when compared to the previous ones

is that it can dynamically estimate the time period in which the event occurs [22]. This means that there is no need to specify a fixed time period to detect events, as the algorithm adjusts this period automatically. The algorithm computes the anomaly of mention creation of a certain word at a time slice by taking into account the expected frequency of words that contain at least one mention in the same time slice. The parameters that MABED needs were set to their default values, where θ = 0.7 and σ = 0.7. Another parameter is p, which stands for the number of words in a tweet. Given that the average number of words per tweet is 10.7 [22], p was set to 10. Users can also set the desired number of detected events in the parameter k. The last parameters are the maximum and minimum term support. Just like the previous studies, the parameters were set to 0.01 and 0.0001.

Evaluating Event Detection

To measure the performance of each event detection algorithm, we used the same collection that was labeled by CrowdFlower workers, which consists of 135,887 tweets. To conduct comprehensive comparisons, we devised several scenarios in which we try different preprocessing settings. The first experiment focused on applying different tokenization techniques to the data collection. In this context, tokenization stands for the splitting of terms into smaller pieces, often called *tokens*. SONDY's data manipulation service offers three types of tokenization: 1-gram, 2-gram, and 3-gram. We decided to experiment with the three different settings in all experiments. The second setting that we looked into was the message time-slice partitioning option. By default, SONDY recommends a time-slice of 30 minutes, which we found reasonable for our experiments. However, we decided to experiment with a 60 minute time-slice as well. The evaluation measures that we adopted were the standard precision 4.4, recall 4.5 and harmonic mean (F_1 measure) 4.6. The equations for each of the applied evaluation measures are:

$$\text{Precision} = \frac{\text{relevant retrieved}}{\text{(relevant retrieved+non-relevant retrieved)}} \tag{4.4}$$

$$\text{Recall} = \frac{\text{relevant retrieved}}{\text{(relevant retrieved+relevant not retrieved)}} \tag{4.5}$$

$$F_1 \text{ measure} = 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}} \tag{4.6}$$

To compute precision,recall, and F_1 measure, first, we manually examined the output of each algorithm and compared it to the 66 events that were identified in *EveTAR*. If an algorithm repeats the same event in different wordings, it does not count as a different event. Any event that an algorithm detects but does not belong to the 66 events is considered non-relevant. Given such criteria, we used the number of relevant events from each algorithm, and used it along with the number of retrieved events in the calculations. Second, we used an automated technique to compute precision, recall and F_1 measure. Further details about the automatic evaluation technique will be presented later. The results of the first experiment are shown in Table 4.7, where the time slice was fixed to 30 minutes. The Table shows the evaluation measures for 1-gram, 2-gram, and 3-gram options. As for MABED, the parameter k was tested with the following values: 25, 50, 75, and 100. The results with the label [A] stand for the automatic evaluation approach, while the reminder of the results without the label were done manually. The remaining algorithm parameters were not changed from their default configurations that were mentioned previously.

Table 4.7. Precision, Recall, and F_1 measure for the 30 minute time slice setting in *EveTAR*

	1-gram			2-gram			3-gram		
Algorithm	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1
EDCoW	0.09	0.15	0.11	0.18	0.24	0.21	0.25	0.27	0.26
Peaky Topics	0.11	0.80	0.19	0.12	**0.88**	0.21	0.10	**0.94**	0.18
Peaky Topics[A]	0.10	0.71	0.20	0.12	**0.88**	0.21	0.09	0.83	0.16
MABED(25)	**0.80**	0.30	0.44	**0.64**	0.24	0.35	0.56	0.21	0.31
MABED(25)[A]	0.56	0.21	0.31	0.60	0.23	0.33	**0.68**	0.25	0.37
MABED(50)	0.60	0.45	0.52	0.50	0.38	0.43	0.50	0.38	0.43
MABED(50)[A]	0.58	0.44	0.50	0.60	0.45	0.52	0.64	0.48	0.55
MABED(75)	0.56	0.64	0.60	0.47	0.53	0.50	0.41	0.47	0.44
MABED(75)[A]	0.61	0.70	0.65	**0.64**	0.73	**0.68**	0.63	0.71	0.67
MABED(100)	0.56	0.64	0.60	0.40	0.61	0.48	0.41	0.62	0.49
MABED(100)[A]	0.61	**0.92**	**0.73**	0.56	0.85	0.67	0.58	0.88	**0.70**

The second experiment focused on the 60 minute time slice. In Table 4.8, the configurations from the previous experiment were maintained except for the time slice option. The results in the Table are given in a similar manner to the Table 4.7, where precision, recall, and F_1 measure are computed for all algorithms. Notice that in both tables, we did not include the results from the persistent conversations algorithm. With experiments, we noticed that both algorithms produce similar results. In fact, both algorithms tend to produce the same number of events with an extremely similar textual description. Hence, we refrained from duplicating the results of both algorithms in the tables and used our findings from experimenting with peaky topics.

Table 4.8. Precision, Recall, and F_1 measure for the 60 minute time slice setting in *EveTAR*

	1-gram			2-gram			3-gram		
Algorithm	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1
EDCoW	0.14	0.12	0.13	0.29	0.18	0.22	0.38	0.18	0.24
Peaky Topics	0.17	0.76	0.28	0.17	0.79	0.28	0.15	0.82	0.25
Peaky Topics[A]	0.17	0.76	0.28	0.19	**0.89**	0.31	0.15	0.85	0.26
MABED(25)	**0.64**	0.24	0.35	**0.72**	0.27	0.40	0.60	0.23	0.33
MABED(25)[A]	0.48	0.18	0.26	0.60	0.23	0.33	**0.68**	0.26	0.37
MABED(50)	0.60	0.45	0.52	0.48	0.36	0.41	0.50	0.38	0.43
MABED(50)[A]	0.54	0.41	0.47	0.60	0.45	0.52	0.62	0.47	0.53
MABED(75)	0.47	0.53	0.50	0.49	0.56	0.52	0.47	0.53	0.50
MABED(75)[A]	0.59	0.67	0.62	0.61	0.70	0.65	0.56	0.64	0.60
MABED(100)	0.51	0.77	0.61	0.49	0.74	0.59	0.50	0.76	0.60
MABED(100)[A]	0.59	**0.89**	**0.71**	0.57	0.86	**0.69**	0.58	**0.88**	**0.70**

Inter-annotator Agreement

To ensure that our labels for the output of each algorithm are accurate, we asked another annotator to label the resulting events as either relevant or non-relevant. First, we provided the new annotator with a list of the 66 events in the collection. Then, we asked the annotator to label the output of each algorithm as either relevant or non-relevant. We considered an event to be relevant if it was included in the list of 66 events of our collection. Any duplicate instances of a relevant event were considered non-relevant. To measure the inter-annotator agreement, we used Cohen's Kappa [54]. This Kappa is different from Fleiss' Kappa that we computed in section 4.1.3 because it is used to compute the agreement between two annotators. The computed Kappa tries to eliminate any labels obtained by chance; thus reducing the amount of uncertainty in the labels obtained manually. The equation to compute Cohen's Kappa is a slightly modified version of 4.1, which is given as follows:

$$K = 1 - \frac{1 - P_\circ}{1 - P_c} \tag{4.7}$$

In equation 4.7, when annotators agree completely, K=1. The value of P_c determines if annotations were obtained by chance; in this case K might be ≤ 0. By computing Cohen's Kappa for the labels of the three algorithms, we had an average agreement of 0.68. We noticed that the average Kappa value was higher in the labels obtained for MABED, which was 0.73. This might be due to clarity of the event descriptions produced by MABED when compared to EDCoW and Peaky Topics. To improve the average Kappa for all the algorithms, the annotators resolved the labels of the events where they disagreed on. This process improved the Kappa slightly from 0.68 to 0.69. We believe that this slight improvement is due to the fact that one of the annotators is not familiar with the events in our Arabic collection. Although the annotator was provided with a list of all the events in the collection; some of the algorithms produced specific event-related details that were not clear to the annotator. To grasp such details, the annotator had to at least be familiar with the event details, which were not provided during annotation.

Automatic Evaluation

In addition to manual evaluation, we *automatically* evaluated event detection using the approach of Petrović [44]. Automatic evaluation was done by computing the proportion (P) of tweets produced from each algorithm that are covered by the events in *EveTAR*. Figure 4.9 illustrates the process of automatic evaluation. For this particular type of evaluation, we view each event as a cluster of tweets. Hence, algorithms that don't produce tweets like EDCoW could not be evaluated automatically. To get the proportion of covered tweets, we take the intersection of the tweets produced by a particular algorithm's events with the tweets from the events in *EveTAR*. Then, we set a condition on the proportion to be more than the threshold value θ, which was set to 0.5. This means

that we consider an event to be valid (i.e, covered by the events in *EveTAR*) if more than half of the tweets in this event match the tweets in *EveTAR*. The process shown in Figure 4.9 is repeated for all the events produced for each algorithm to get number of covered events. The final number of covered events for each algorithm was used to compute recall, precision, and F_1 using the equations that were given previously. Here, we consider the number of covered events to be the number of relevant retrieved events that were mentioned in equation 4.4 and 4.5.

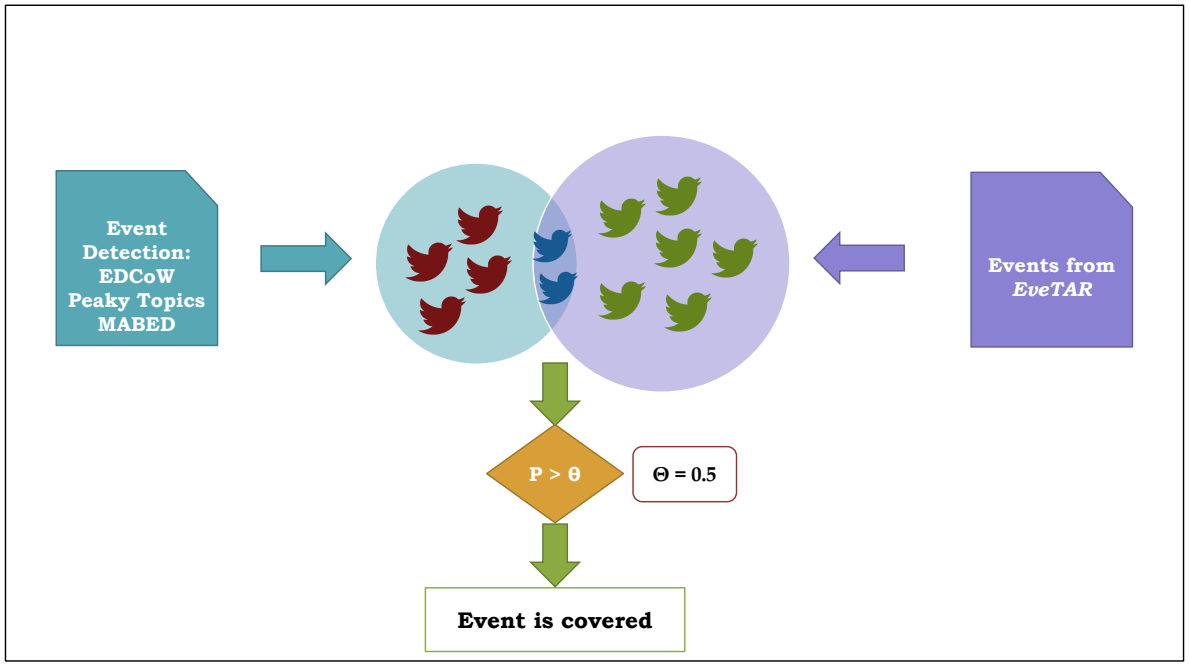


Figure 4.9. Process of automatically evaluating the event detection algorithms using *EveTAR*

Evaluation using other English test collections

In addition to the two conducted experiments, we compare the output of each algorithm using our collection with the numbers reported in the literature. Starting with EDCoW, the authors stated that the value of γ controls the precision of the algorithm. We noticed that changing γ did not affect our output that much. When γ is 10, EDCoW has a

precision of 0.14 [62]. At the same γ setting, we got a precision of 0.06 in the first experiment (30 minutes) and 0.14 at the second experiment (60 minutes). In general, we think that EDCoW did not perform better using our test collection, however; it produced comparable results. Moreover, we did not notice any change in the algorithm performance when γ was 5, hence the output when γ is 5 is the same as the output when γ is 10. In MABED, the authors reported the performance of the algorithm in terms of precision and F_1 measure. In an English corpus, MABED has a precision of 0.78 and F_1 of 0.68. While the algorithm performed better on a French corpus, obtaining a precision and F_1 of 0.83 [22]. Since we used MABED in four different settings, we will compare the outputs obtained from the first experiment that we conducted at 30 minutes. We noticed that the precision of MABED (25) is higher than the precision of the English corpus. However, the F_1 is much lower due to the low recall. In the French corpus, the outputs reported by MABED slightly outperform the precision of MABED (25) but the F_1 is still much better in the French corpus [22]. As for Peaky Topics, the authors did not report any numbers that we could use to compare with our evaluation measures.

To compare our test collection with one of the available test collections, we used the English test collection of McMinn et al. [35]. In terms of annotator agreement, the Wikipedia approach of [35] was 0.72, which is higher than the average agreement that we had of 0.6. This might be due to the difference in the number of tweets that were annotated in both collections. As we mentioned in Table 4.2 that obtained annotations for 135,887 tweets. However, in the English test collection, a total of 39,980 tweets were annotated. This difference in the tweet number coupled with the nature of the events and event-related tweets might cause the annotator agreement to be higher in the English test collection. Additionally, we wanted to compare the output of EDCoW, MABED, and Peaky Topics when applied on the English test collection. Thus, we crawled a subset of the English test collection and obtained a total of 22,814 tweets that span 361 events. Then, we used the same parameters that were used in the two previous experiments on

our collection with the English test collection. The results of experimenting with the 30 minute time slice are depicted in Table 4.9.

Table 4.9. Precision, Recall, and F_1 measure for the 30 minute time slice setting in the English Test collection

	1-gram			2-gram			3-gram		
Algorithm	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1
EDCoW	0.17	0.05	0.07	0.15	0.04	0.06	0.36	0.06	0.10
Peaky Topics	0.17	**0.58**	0.26	0.16	**0.56**	0.25	0.21	**0.76**	0.33
Peaky Topics [A]	0.08	0.29	0.13	0.08	0.29	0.13	0.08	0.29	0.13
MABED(25)	0.92	0.06	0.12	0.84	0.06	0.11	0.92	0.06	0.12
MABED(25) [A]	**1.00**	0.07	0.13	**1.00**	0.07	0.13	**1.00**	0.07	0.13
MABED(50)	0.88	0.12	0.21	0.84	0.12	0.20	0.88	0.12	0.21
MABED(50) [A]	**1.00**	0.14	0.25	**1.00**	0.14	0.25	**1.00**	0.14	0.24
MABED(75)	0.83	0.17	0.28	0.79	0.16	0.27	0.91	0.19	0.31
MABED(75) [A]	**1.00**	0.22	0.36	**1.00**	0.21	0.36	0.99	0.20	0.34
MABED(100)	0.89	0.24	0.38	0.86	0.24	0.37	0.91	0.25	0.39
MABED(100) [A]	**1.00**	0.28	**0.43**	**1.00**	0.29	**0.45**	0.99	0.27	**0.43**

We conducted an additional experiment on the English test collection using the 60 minute time slice. The results of each algorithm are given in Table 4.10.

Table 4.10. Precision, Recall, and F_1 measure for the 60 minute time slice setting in the English test collection

	1-gram			2-gram			3-gram		
Algorithm	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1	**Precision**	**Recall**	F_1
EDCoW	0.39	0.05	0.09	0.47	0.04	0.07	0.36	0.03	0.05
Peaky Topics	0.31	**0.54**	0.39	0.36	**0.65**	**0.46**	0.35	**0.63**	**0.45**
Peaky Topics [A]	0.16	0.28	0.21	0.16	0.29	0.21	0.16	0.28	0.20
MABED(25)	0.92	0.06	0.12	0.80	0.06	0.10	0.84	0.06	0.11
MABED(25) [A]	**1.00**	0.07	0.13	**1.00**	0.07	0.13	**1.00**	0.07	0.13
MABED(50)	0.80	0.11	0.19	0.86	0.12	0.21	0.82	0.11	0.20
MABED(50) [A]	**1.00**	0.14	0.25	**1.00**	0.14	0.25	**1.00**	0.14	0.24
MABED(75)	0.81	0.17	0.28	0.89	0.19	0.31	0.84	0.17	0.29
MABED(75) [A]	**1.00**	0.22	0.36	**1.00**	0.21	0.35	0.99	0.20	0.34
MABED(100)	0.88	0.24	0.38	0.87	0.24	0.38	0.90	0.25	0.39
MABED(100) [A]	**1.00**	0.28	**0.44**	**1.00**	0.29	0.45	0.99	0.27	0.43

After conducting the four experiments with different time slices, we came with the following conclusions:

- Algorithms like EDCoW were difficult to evaluate due to the implementation of it. The wavelet algorithm depends on clustering of correlated bursty words. This clustering does not necessarily grantee that terms from the same event will be clustered together. For example, if e_1 and e_2 are two different events that happened at the same time. According to EDCoW, if they happen to have the same wavelet burst, they will be clustered together. This observation was evident with many events from our collection, which caused events to be meaninglessly clustered together. This explains the low precision that EDCoW received in the the first experiment, as shown in Table 4.7. However, we discovered that the result improved slightly in the second experiment.

- Manually evaluating the outcome of algorithms like peaky topics and persistent conversations was extremely difficult. The output of each algorithm in the 1-gram

setting is a list of words that may or may not represent events. Moreover, sometimes those algorithms produce more than 600 events, which makes it even more difficult to keep track of the relevant event count. Actually, the algorithms do not distinguish between events and discussions. As we noticed that any "hot topic" that gets mentioned a lot is considered an event. This explains the huge number of events that such algorithms produce, which mostly consists of noise and insignificant events. However, the advantage of these algorithms is that due to their sensitivity to time, almost all of the 66 events are always included in their output. The high recall values in Table 4.7 and 4.8 clearly depict this observation.

- As we initially predicted, MABED outperforms all the algorithms in terms of precision, recall, and F_1 measure. We noticed that the output produced from MABED is well formatted, readable, and closely resembles the 66 events that we have in our test collection. Additionally, setting k to 25 appears to be the best setting for MABED, as we noticed a decline in precision when k is more than 25. We believe that setting k to 25 is the ideal setting because increasing k tends to produce duplicate events.

- Automatic evaluation is more accurate than manual evaluation. This is especially true for the English test collection. The annotators that were assigned with the labeling process tend to lose focus when the number of events is large. Thus; performing automatic evaluation on the tweet level ensures that all events will be taken into consideration when computing precision, recall, and F_1 measure.

- The results obtained from automatic evaluation are comparable to the ones produced by manual evaluation in *EveTAR*. This observation shows that regardless of the slight human errors produced from manual labels, we can still trust the precision, recall, and F_1 numbers when they are computed automatically.

- The experiments conducted on the English test collection show higher precision values when compared to *EveTAR*. There are many factors that might contribute to those results, the first one might be the huge difference in the number of events in each test collection. It is extremely difficult to compare between 66 and 361 events. Moreover, the way each data collection was obtained contributes to the quality of the test collection.

Evaluating Ad-hoc Search

In addition to event detection, *EveTAR* is designed to support evaluation of other tasks like ad-hoc search as it provides a short query per event in addition to a list of relevant tweets. We epxeriment with running and evaluating two ad-hoc systems that we refer to as baseline (BL) and query expansion (QE) with *EveTAR*. We re-implemented these models based on those of one of the good teams participating in the ad-hoc search task in TREC-2013 microblog track [32]. We ran the search systems over the Lucene index of the 590M tweets. We then evaluate ad-hoc search using MAP and P@30 which are the two main measures used in evaluation of tweet ad-hoc search in TREC-2013 [32]. Results are summarized in Table 4.11.

Table 4.11. Ad-hoc search performance with *EveTAR*

Model	MAP	P@30
BL	0.1283	0.3783
QE	0.1207	0.3384

We observe that the values of P@30 for both models are in range of the P@30 values reported in [32], however, value of MAP is much lower than expected. The difference in ad-hoc search performance between *EveTAR* and the English collection in [32] might be due to the very big difference in size, as the English set is much smaller than *EveTAR*. Additionally, we ran the models using parameter values reported in the original paper (tuned on the English collection) which might not be good for ad-hoc search over Arabic

tweets. Further experiments are needed to understand the performance of ad-hoc search with *EveTAR*.

CHAPTER 5. CONCLUSION AND FUTURE WORK

By the end of this *first* thorough study that tackles the problem of building a test collection for significant-event detection in Arabic tweets, we conclude this work with some final remarks and further guidelines for future work in the upcoming section.

5.1 Conclusion

In this work, we present our work on constructing *EveTAR*, then we followed a pipeline that consists of four main stages to build the test collection. The first stage consists of using Wikipedia's current events portal to get a list of candidate events from our January 2015 data collection. The list of events was refined and filtered to include significant events only. The second stage involves developing queries to retrieve event-related tweets from our data collection. For each event, we developed 6 queries, that were later formatted to follow Lucene's query syntax. The third stage focuses on obtaining relevance judgments for the event-related tweets through crowdsourcing. We used CrowdFlower to launch the 66 events as labeling jobs. The total number of labeled tweets was 135K, which consists of 51K relevant tweets from all events. The fourth and final stage addresses the evaluation of *EveTAR* by using it to evaluate the performance of three state-of-the-art event detection algorithms.

In this thesis, we answered all the raised research questions and showcased how a test collection can be collected to serve other information retrieval tasks, such as ad-hoc search.

The outcome of the construction and evaluation of the test collection is summarized as follows:

- The quality of crowdsourcing labels depend on the quality of the results obtained from collection search. Therefore, the queries must be accurate to aid in the retrieval of relevant tweets.

- Annotators need clear and concise instructions to label tweets accordingly. When the job description lacks necessary information and examples, this will affect the quality of the labeling job in a negative way.

- State-of-the-art event detection algorithms are able to successfully detect events from our Arabic test collection. Some of which report high precision, recall, and F1 measures.

- Manipulating some of the algorithm's parameters yields different kinds of results. For instance, choosing k to be 25 in MABED gives better results when compared to higher and lower values.

- Our developed test collection is comparable to some of the existing test collections. However, we distinguish our work as the *first* Arabic test collection for the task of event detection.

5.2 Future Work

With the aid of this built test collection, we aim to conduct further studies that leverage the obtained labels for other information retrieval tasks, like tweet filtering. Moreover, we would like to extend the test collection for summarization and Tweet Timeline Generation (TTG) tasks. To conduct extensive evaluations on the test collection, we plan to use other algorithms like ET [43] and tune their parameters according to the requirements of Arabic tweets. Additionally, the output of the event detection algorithms can be examined by experts or through crowdsourcing to obtain accurate evaluations. Due to restrictions on full tweet distribution by Twitter, we plan to provide our own API through which users of our collection can get easy access to the full tweets while still abiding by Twitter's terms of service. To enrich the test collection, the output of automatic event detection algorithms can be used to refine the initial list of events and judgments, in a fashion similar to [35]. This process will compensate for the lack of completeness that *EveTAR*

currently suffers from. Finally, it would be interesting to see how new state-of-the-art event detection algorithms can benefit from *EveTAR* to evaluate their systems.

Bibliography

[1] Dhekar Abhik and Durga Toshniwal. Sub-event detection during natural hazards using features of social media data. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 783–788. International World Wide Web Conferences Steering Committee, 2013.

[2] Puneet Agarwal, Rajgopal Vaithiyanathan, Saurabh Sharma, and Gautam Shroff. Catching the long-tail: Extracting local news events from twitter. In *ICWSM*, 2012.

[3] Charu C Aggarwal and Karthik Subbian. Event detection in social streams. In *SDM*, volume 12, pages 624–635. SIAM, 2012.

[4] James Allan. Introduction to topic detection and tracking. In *Topic detection and tracking*, pages 1–16. Springer, 2002.

[5] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45. ACM, 1998.

[6] Nasser Alsaedi and Pete Burnap. Arabic event detection in social media. In *Computational Linguistics and Intelligent Text Processing*, pages 384–401. Springer, 2015.

[7] Nasser Alsaedi, Peter Burnap, and Omer Farooq Rana. A combined classification-clustering framework for identifying disruptive events. 2014.

[8] Foteini Alvanaki, Sebastian Michel, Krithi Ramamritham, and Gerhard Weikum. See what's enblogue: real-time emergent topic identification in social media. In *Proceedings of the 15th International Conference on Extending Database Technology*, pages 336–347. ACM, 2012.

[9] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 11:438–441, 2011.

[10] Thorsten Brants, Francine Chen, and Ayman Farahat. A system for new event detection. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 330–337. ACM, 2003.

[11] Chris Buckley and Ellen M Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32. ACM, 2004.

[12] Ben Carterette. Robust test collections for retrieval evaluation. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 55–62. ACM, 2007.

[13] Ben Carterette, James Allan, and Ramesh Sitaraman. Minimal test collections for retrieval evaluation. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 268–275. ACM, 2006.

[14] Flavio Chierichetti, Jon Kleinberg, Ravi Kumar, Mohammad Mahdian, and Sandeep Pandey. Event detection via communication pattern analysis. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[15] Gordon V Cormack, Christopher R Palmer, and Charles LA Clarke. Efficient construction of large test collections. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 282–289. ACM, 1998.

[16] Ahmed AA Esmin, Rômulo SC Júnior, Wagner S Santos, Cássio O Botaro, and Thiago P Nobre. Real-time summarization of scheduled soccer games from twitter stream. In *Natural Language Processing and Information Systems*, pages 220–223. Springer, 2014.

[17] Xiao Feng, Shuwu Zhang, Wei Liang, and Zhe Tu. Real-time event detection based on geo extraction and temporal analysis. In *Advanced Data Mining and Applications*, pages 137–150. Springer, 2014.

[18] Jonathan G Fiscus and George R Doddington. Topic detection and tracking evaluation overview. *Topic detection and tracking*, pages 17–31, 2002.

[19] Thomas Gottron, Olaf Radcke, and Rene Pickhardt. On the temporal dynamics of influence on the social semantic web. In *Semantic Web and Web Science*, pages 75–87. Springer, 2013.

[20] Adrien Guille and Cécile Favre. Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach. *Social Network Analysis and Mining*, 5(1):1–18, 2015.

[21] Adrien Guille, Cécile Favre, Hakim Hacid, and Djamel A Zighed. Sondy: An open source platform for social dynamics mining and analysis. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1005–1008. ACM, 2013.

[22] Antoine Guille and Cécile Favre. Mention-anomaly-based event detection and tracking in twitter. In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, pages 375–382. IEEE, 2014.

[23] Mehdi Hosseini, Ingemar J Cox, Natasa Milic-Frayling, Trevor Sweeting, and Vishwa Vinay. Prioritizing relevance judgments to improve the construction of ir test collections. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 641–646. ACM, 2011.

[24] R Kaushik, S Apoorva Chandra, Dilip Mallya, JNVK Chaitanya, and S Sowmya Kamath. Sociopedia: An interactive system for event detection and trend analy-

sis for twitter data. In *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, pages 63–70. Springer, 2016.

[25] Arpit Khurdiya, Lipika Dey, Diwakar Mahajan, and Ishan Verma. Extraction and compilation of events and sub-events from twitter. In *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 504–508. IEEE Computer Society, 2012.

[26] Giridhar Kumaran and James Allan. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304. ACM, 2004.

[27] FA Kunneman and APJ van den Bosch. Event detection in twitter: A machine-learning approach based on term pivoting. 2014.

[28] Victor Lavrenko, James Allan, Edward DeGuzman, Daniel LaFlamme, Veera Pollard, and Stephen Thomas. Relevance models for topic detection and tracking. In *Proceedings of the second international conference on Human Language Technology Research*, pages 115–121. Morgan Kaufmann Publishers Inc., 2002.

[29] Chenliang Li, Aixin Sun, and Anwitaman Datta. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 155–164. ACM, 2012.

[30] Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. Tedas: A twitter-based event detection and analysis system. In *Data engineering (icde), 2012 ieee 28th international conference on*, pages 1273–1276. IEEE, 2012.

[31] Yuan Liang, James Caverlee, and Cheng Cao. A noise-filtering approach for spatio-temporal event detection in social media. In *Advances in Information Retrieval*, pages 233–244. Springer, 2015.

[32] Jimmy Lin and Miles Efron. Overview of the TREC-2013 Microblog Track. In *TREC-2013*, 2013.

[33] Michael McCandless, Erik Hatcher, and Otis Gospodnetic. *Lucene in Action: Covers Apache Lucene 3.0.* Manning Publications Co., 2010.

[34] Richard McCreadie, Craig Macdonald, Iadh Ounis, Miles Osborne, and Slobodan Petrovic. Scalable distributed event detection for twitter. In *Big Data, 2013 IEEE International Conference on*, pages 543–549. IEEE, 2013.

[35] Andrew J McMinn, Yashar Moshfeghi, and Joemon M Jose. Building a large-scale corpus for evaluating event detection on twitter. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 409–418. ACM, 2013.

[36] Eric Medvet and Alberto Bartoli. Brand-related events detection, classification and summarization on twitter. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, volume 1, pages 297–302. IEEE, 2012.

[37] Polykarpos Meladianos, Giannis Nikolentzos, François Rousseau, Yannis Stavrakas, and Michalis Vazirgiannis. Degeneracy-based real-time sub-event detection in twitter stream. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[38] Tanushree Mitra and Eric Gilbert. Credbank: A large-scale social media corpus with associated credibility annotations. In *Proceedings of the 9th International Conference on Web and Social Media, Oxford, UK*, 2015.

[39] Miles Osborne, Sean Moran, Richard McCreadie, Alexander Von Lunen, Martin D Sykora, Elizabeth Cano, Neil Ireson, Craig Macdonald, Iadh Ounis, Yulan He, et al. Real-time detection, tracking, and monitoring of automatically discovered events in social media. 2014.

[40] Georgios Paltoglou. Sentiment-based event detection in twitter. *Journal of the Association for Information Science and Technology*, 2015.

[41] Symeon Papadopoulos, Raphael Troncy, Vasileios Mezaris, Benoit Huet, and Ioannis Kompatsiaris. Social event detection at mediaeval 2011: Challenges, dataset and evaluation. In *MediaEval*, 2011.

[42] Ron Papka, James Allan, et al. On-line new event detection using single pass clustering. *UMass Computer Science*, 1998.

[43] Ruchi Parikh and Kamalakar Karlapalem. Et: events from tweets. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 613–620. International World Wide Web Conferences Steering Committee, 2013.

[44] Sasa Petrović. *Real-time event detection in massive streams.* PhD thesis, School of Informatics, University of Edinburgh, 2013.

[45] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189. Association for Computational Linguistics, 2010.

[46] Saša Petrović, Miles Osborne, and Victor Lavrenko. Using paraphrases for improving first story detection in news and twitter. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–346. Association for Computational Linguistics, 2012.

[47] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. Automatic sub-event detection in emergency management using social media. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 683–686. ACM, 2012.

[48] Narumol Prangnawarat, Ioana Hulpus, and Conor Hayes. Event analysis in social media using clustering of heterogeneous information networks. In *The Twenty-Eighth International Flairs Conference*, 2015.

[49] Shahzad Rajput, Virgil Pavlu, Peter B Golbus, and Javed A Aslam. A nugget-based test collection construction paradigm. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1945–1948. ACM, 2011.

[50] Alan Ritter, Oren Etzioni, Sam Clark, et al. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012.

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

[52] Mark Sanderson. *Test collection based evaluation of information retrieval systems*. Now Publishers Inc, 2010.

[53] David A Shamma, Lyndon Kennedy, and Elizabeth F Churchill. Peaks and persistence: modeling the shape of microblog conversations. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, pages 355–358. ACM, 2011.

[54] Julius Sim and Chris C Wright. The kappa statistic in reliability studies: use, interpretation, and sample size requirements. *Physical therapy*, 85(3):257–268, 2005.

[55] Alan F Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 321–330. ACM, 2006.

[56] Giovanni Stilo and Paola Velardi. Efficient temporal mining of micro-blog texts and its application to event discovery. *Data Mining and Knowledge Discovery*, pages 1–31, 2015.

[57] Bayar Tsolmon and Kyung-Soon Lee. An event extraction model based on timeline and user analysis in latent dirichlet allocation. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1187–1190. ACM, 2014.

[58] Julián Urbano, Mónica Marrero, and Diego Martín. On the measurement of test collection reliability. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 393–402. ACM, 2013.

[59] Yu Wang, David Fink, and Eugene Agichtein. Seeft: Planned social event discovery and attribute extraction by fusing twitter and web content. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[60] Wei Wei, Kenneth Joseph, Wei Lo, and Kathleen M Carley. A bayesian graphical model to discover latent events from twitter. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[61] Andreas Weiler, Michael Grossniklaus, and Marc H Scholl. Evaluation measures for event detection techniques on twitter data streams. In *Data Science*, pages 108–119. Springer, 2015.

[62] Jianshu Weng and Bu-Sung Lee. Event detection in twitter. *ICWSM*, 11:401–408, 2011.

[63] Yiming Yang, Tom Ault, Thomas Pierce, and Charles W Lattimer. Improving text categorization methods for event tracking. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 65–72. ACM, 2000.

[64] Yiming Yang, Jaime G Carbonell, Ralf D Brown, Thomas Pierce, Brian T Archibald, and Xin Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, (4):32–43, 1999.

[65] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 28–36. ACM, 1998.

[66] Siqi Zhao, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. Human as real-time sensors of social and physical events: A case study of twitter and sports games. *arXiv preprint arXiv:1106.4300*, 2011.

[67] Deyu Zhou, Liangyu Chen, and Yulan He. An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[68] Xiangmin Zhou and Lei Chen. Event detection over twitter social media streams. *The VLDB JournalThe International Journal on Very Large Data Bases*, 23(3):381–400, 2014.

APPENDIX A. COLLECTION KEYWORDS

The list of keywords in Table A.1 and A.2 are the 400 tokens that were used to crawl the data collection. For further details about the process of using the keywords to collect the tweets, please refer to section 3.1.

Table A.1. The tokens used for crawling the data collection through the streaming API(1 of 2)

عبدالله	وان	النار	#الاتحاد	خمس	بالله	رتويت	من
ل	وأنا	نفسه	قلبك	بها	مثل	يارب	في
صح	هذي	شر	كم	كما	يكون	بـ	الله
مرة	وأتوب	ربنا	استغفر	فى	الى	قلبي	على
معي	#شعر	للتسجيل	يضيفك	بما	ف	بين	لا
مما	آخر	نحن	الصورة	صحيح	رب	هي	و
لأن	كثير	#تاملات_ايمانية	النبي	ولكن	الخير	ربي	ما
الإنسان	احب	منهم	ابن	نفسك	هذه	بك	اللهم
ايه	ربك	وين	إليه	مش	الحياة	ومن	كل
قل	لاتفوتك	النصر	أجمل	نفسي	الهلال	#الهلال	أن
عندك	وعلى	إنك	وقت	الملك	واحد	محمد	عن
#Hadith	النوم	المتابعين	الشيخ	ناس	#الرياض	عليك	اللي
فـ	بأن	مساء	#قطر	منك	#رتويت	أو	ولا
صار	صلاة	حب	لكل	لمن	م	غير	يا
مباراة	فِي	اله	جميع	الحب	إله	أنا	مع
كلها	كلمة	الحمدلله	متابع	اكثر	صلى	الي	ان
أنك	المسلمين	فوق	الأرض	سورة	لكن	فقط	إلا
معنا	#بر_الوالدين	منها	عندي	ذلك	رسول	ي	انا
مجاني	عذاب	ليش	عليها	كانت	لزيادة	#السعودية	لك

Table A.2. The tokens used for crawling the data collection through the streaming API(2 of 2)

جديد	مِن	اجعل	راح	بيع	متابعينك	الآن	هو
فرصـتك	فـ	وحده	تم	فيك	وانا	كنت	لو
تبي	#الشعب ـيقول ـكلمته	شاء	الليل	الاف	احد	شخص	بس
يَ	كلام	مره	كن	لكم	بن	التي	لي
#فلورينا	الكلام	الاتحاد	اول	لأ	#النصر	قد	يوم
عادي	عبر	#غرد ـبصورة	#Quran	بل	#تطبيق ـقرآني	وش	هذا
قوة	السماوات	فيني	اغفر	القلب	انك	عند	عليه
أفضل	الحين	أكبر	قلوبنا	أسألك	هم	خير	قال
اخر	مَ	غيرك	فإن	انه	عندما	أنت	بعد
نور	ولم	عيني	عليكم	ممكن	صباح	وما	كان
حسابك	حياتك	ليه	عني	بدون	لهم	الان	له
بشكل	مسلم	آمنوا	عنه	بي	متابعين	به	الناس
عام	إليك	كله	قلب	الصلاة	ليس	بعض	إن
الجمعة	سجل	السلام	وبحمده	جميل	ثم	عمل	حتى
حيث	حياتي	الوقت	كذا	بكل	تويتر	عدد	إلى
اكبر	الجنه	لقد	تقول	منه	هل	اني	لم
اشترك	فلورينا	تحب	عليهم	حين	هنا	وسلم	والله
العرب	أنه	القلوب	جديدة	ترى	أكثر	لنا	شيء
يبي	اضافة	مني	كثر	العالم	لن	فيها	الذي
خلاص	ذا	قلت	انتي	عنك	لـ	وهو	فيه
برنامج	القرآن	ابي	مهما	هناك	يعني	ريتويت	إذا
البشر	حول	صل	معك	وإن	يقول	عشان	الا
حاجه	الموت	السماء	الف	طيب	تكون	او	اليوم
الفيس	عنا	لازم	نفس	دون	ريال	الذين	اذا
السعادة	حد	وكل	حساب	وفي	اي	مو	شي
مين	فقد	احبك	تابعني	أعوذ	أحد	لله	علي
والأرض	سبحانك	متى	أول	علينا	لها	العظيم	قبل
أحب	حق	ترا	يمكن	وَ	فلا	كيف	انت
أيها	بلا	اجمل	ولو	#الكويت	الحمد	ب	إني
فَ	وأنت	تحت	للمتواجدين	وانت	الجنة	سبحان	الدنيا
ألف	جداً	جدا	#دبي	أي	صورة	ع	لما

APPENDIX B. EVENTS

The list of 66 events that were used in this study are given in this appendix, along with their English translation and corresponding category. Both Table B.1 and B.2 include all the chronologically-ordered events of the month of January 2015.

Table B.1. List of Arabic and English event titles and categories (1 of 2)

Event	Arabic title	English title	Event Category
1	حادثة التفجير الإنتحاري بمدينة إب في اليمن	Suicide bombing in Ibb, Yemen	Armed conflicts and attacks
2	ليتوانيا تتخلى عن الليتاس وتنضم لليورو	Lithuania adopts the euro instead of Litas	Business and economy
3	انضمام فلسطين للمحكمة الجنائية	Palestine joins the International Criminal Court	International relations
4	وفاة أبو أنس الليبي قبل أيام من محاكمته في نيويورك	Death of Abu Anas al-Libi	Armed conflicts and attacks
5	عقوبات كوريا الشمالية بعد الهجوم على سوني	North Korea sanctions after Sony hack	International relations
6	بناء أول كنيسة في إسطنبول منذ قرن	Turkey permits the building of a church in Istanbul	Arts and culture
7	إنشاء حزب جديد قبل إنتخابات اليونان	Formation of new party before Greek elections	Politics and elections
8	بوكو حرام تختطف ٤٠ شابا	Boko Haram had kidnapped around 40 boys	Armed conflicts and attacks
9	بوكو حرام تسيطر على قاعدة عسكرية	Boko Haram controls military base	Armed conflicts and attacks
10	غارات جوية على طالبان في باكستان	Pakistan Air Force strikes Pakistani Taliban	Armed conflicts and attacks
11	قتلى من الحوثيين في انفجار عبوة بمحافظة ذمار	Bombing in Dhamar kills Houthis	Armed conflicts and attacks
12	اكتشاف مقبرة أثرية لملكة فرعونية	Discovery of tomb of ancient Egyptian queen	Arts and culture
13	قصف ليبيا ناقلة نفط يونانية	Libya bombs Greek-operated oil tanker	Armed conflicts and attacks
14	مقتل رجلي أمن سعوديين قرب حدود العراق	Death of two Saudi guards near Iraq	Armed conflicts and attacks
15	مئات المسافرين علقوا داخل طائرة في أبوظبي	Passengers stuck in airplane in Abu Dhabi	Disasters and accidents
16	لبنان يقيد دخول السوريين بعد تنفيذ قواعد جديدة	Lebanon implements strict immigration rules	International relations
17	مقتل جنود القوات العراقية باشتباكات الأنبار	Death of Iraqi soldiers in Anbar clashes	Armed conflicts and attacks
18	شن قوات التحالف غارات على داعش	Combined Joint Task Force strikes on ISIS	Armed conflicts and attacks
19	عملية انتحارية في مركز للشرطة في اسطنبول	Suicide attack in Turkish police station	Armed conflicts and attacks
20	قتلى بهجوم مسلح على صحيفة شارلي إيبدو	Deaths at armed attack on Charlie Hebdo	Armed conflicts and attacks
21	تحديد هوية منفذي هجوم شارلي إبدو	Identification of Charlie Hebdo suspects	Armed conflicts and attacks
22	تفجير سيارة مفخخة بصنعاء	Car bomb explodes in Sana'a	Armed conflicts and attacks
23	استسلام مشتبه ومطاردة الأخوين كواشي	Surrender of suspect and chase of brothers	Armed conflicts and attacks
24	اعتداءات على مساجد باريس	Attacks on Paris Mosques	Armed conflicts and attacks
25	سقوط قتلى بتفجيرين في بغداد	Suicide bomber kills people in Baghdad	Armed conflicts and attacks
26	بوكو حرام تحرق بلدة باغا النيجيرية	Boko Haram burns Baga in Nigeria	Armed conflicts and attacks
27	مقتل منفذي هجوم شارلي إبدو	Death of Charlie Hebdo suspects	Armed conflicts and attacks
28	احتجاز رهائن في سوق يهودي بفينسين	Captive hostages in Jewish Market in Vincennes	Armed conflicts and attacks
29	تنفيذ عقوبة الجلد علناً بحق رائف البدوي	Raif Badawi receives 50 lashes for insulting Islam	Law and crime
30	فوز استراليا على الكويت في افتتاح نهائيات آسيا	Australia wins AFC cup first match against Kuwait	Sports

Table B.2. List of Arabic and English event titles and categories (2 of 2)

Event	Arabic title	English title	Event Category
31	تفجير مسجد شيعي في باكستان	Suicide bomb attack on Shiite mosque in Pakistan	Armed conflicts and attacks
32	هجوم انتحاري استهدف مقهى بطرابلس	Suicide attack at a cafe in Tripoli, Lebanon	Armed conflicts and attacks
33	اشتباك محتجين في البحرين مع الشرطة	Clash between Bahraini protestors and police	Politics and elections
34	زعماء العالم يشاركون في مسيرة تضامن في باريس	World leaders participate in Paris unit rally	Armed conflicts and attacks
35	هجوم على صحيفة نشرت رسوما مسيئة للرسول	German newspaper arson attack	Armed conflicts and attacks
36	انتحار فتاة يقتل ١٩ شخصاً في نيجيريا	Girl suicide kills 19 person in Nigeria	Armed conflicts and attacks
37	انتشال جزء من حطام الطائرة الاندونيسية	Divers retrieve part of the crashed Indonesian jet	Disasters and accidents
38	اختراق داعش حساب القيادة المركزية الأمريكية	ISIL hacks U.S Central Command Twitter and YouTube feeds	Armed conflicts and attacks
39	كريستيانو رونالدو يفوز بجائزة الكرة الذهبية لأفضل لاعب	Cristiano Ronaldo wins the FIFA Ballon d'Or for 2014	Sports
40	انتشال مسجل قمرة القيادة من حطام طائرة اير آسيا	Divers recover cockpit recorder of Air Asia	Disasters and accidents
41	إعادة محاكمة مبارك في قضية قصور الرئاسة	Egypt's court initiates a retrial of Hosni Mubarak	Law and crime
42	استقالة الرئيس الإيطالي جورجو نابوليتانو	Resignation of Italy's president Giorgio Napolitano	Politics and elections
43	افتتاح مونديال كرة اليد في قطر	Launch of men's handball world championship in Qatar	Sports
44	ختطاف أحمد عوض بن مبارك	Abduction of the chief of staff to Yemen's president	Armed conflicts and attacks
45	مروحية إسرائيلية تقصف هدف في الجولان	Israeli helicopter strike near the border with Syria	Armed conflicts and attacks
46	الجيش الليبي يعلن وقف إطلاق النار	Libyan army declares a ceasefire	Armed conflicts and attacks
47	إستضافة غينيا الإستوائية كأس أمم أفريقيا	Equatorial Guinea to host the Africa Cup of Nations	Sports
48	قناة اليمن ووكالة سبأ تحت سيطرة الحوثي	Houthi rebels seize the official Saba News Agency	Armed conflicts and attacks
49	انقلاب الحوثي والإستيلاء على دار الرئاسة	Houthi rebels take over the residence of the President	Armed conflicts and attacks
50	تهديد داعش اليابان بالرهينتين المحتجزين	ISIS threatens to kill two Japanese citizens	Armed conflicts and attacks
51	استقالة الرئيس اليمني وحكومته	Yemeni President, Prime Minister, and Yemeni cabinet resign	Armed conflicts and attacks
52	وفاة خادم الحرمين الشريفين	Death of Custodian of the Two Holy Mosques King Abdullah	Politics and elections
53	تعطيل الأمم المتحدة إعمار غزة	United nations debilitate the reconstruction of Gazza	Politics and elections
54	مقتل جنود لبنانيين في مواجهات مع داعش	Death of Lebanese soldiers in clash with ISIS	Armed conflicts and attacks
55	إعدام الرهينة الياباني هارونا يوكاوا	ISIS kills the first Japanese hostage	Armed conflicts and attacks
56	الأكراد يستعيدون أجزاء كبيرة من كوباني	Kurdish fighters recapture most of Koban	Armed conflicts and attacks
57	العراق يخسر أمام كوريا الجنوبية في أمم آسيا	Iraq loses against South Korea in AFC Asian cup	Sports
58	هجوم على فندق كورنثيا بطرابلس في ليبيا	Attack on Libyan Corinthia Hotel in Tripoli	Armed conflicts and attacks
59	مقتل جنديين في هجوم لحزب الله على رتل للجيش الاسرائيلي	Death of soldiers in attack on Israeli military convoy	Armed conflicts and attacks
60	قتلى بسلسلة تفجيرات ببغداد وسامراء والفلوجة	Deaths caused by several attacks around Baghdad	Armed conflicts and attacks
61	أنصار بيت المقدس تقتل ٣٠ شخصا في ليلة دامية بسيناء	Deaths caused by terrorist attacks in Sinai Peninsula	Armed conflicts and attacks
62	الحوثيون يستولون على معسكر الحرس الجمهوري	Shiite Houthi rebels seize a Yemeni military base	Armed conflicts and attacks
63	تفجير مسجد شيعي في شيركابور بباكستان	Bombing of Shiite mosque in Shikarpur	Armed conflicts and attacks
64	هجوم داعش على كركوك وسقوط قتلى من البيشمركة	ISIS attack on Kirkuk and peshmerga deaths	Armed conflicts and attacks
65	ملك السعودية يعيد تشكيل مجلس الوزراء	Saudi king reconstructs the council of ministers	Politics and elections
66	داعش يعلن أعدام الرهينة الياباني الثاني	ISIS announces the death of the second Japanese hostage	Armed conflicts and attacks

APPENDIX C. EVENT STATISTICS

This appendix is dedicated to the statistics associated with the identified events. For each event, we present the exact number of relevant, non-relevant, and total judged tweets. Then, we include the labeling time in hours, the % agreement, the trust-based confidence, and Kappa statistic. Events are organized in a similar manner to the previous tables, were event 1 in table C.1 is the same as event 1 in table B.1.

Table C.1. List of event relevance judgment details and statistics (1 of 2)

Event	Relevant	Non-relevant	Total Judgments	Labeling time (hrs)	Agreement(%)	Overall Confidence (avg.)	Kappa
1	637	336	973	110	0.90	0.90	0.57
2	44	39	83	10	0.96	0.96	0.84
3	338	173	511	31	0.95	0.95	0.79
4	1755	151	1906	93	0.98	0.98	0.68
5	558	955	1513	63	0.96	0.96	0.82
6	687	707	1394	57	0.99	0.99	0.94
7	43	634	677	12	0.98	0.98	0.74
8	204	461	665	35	0.99	0.99	0.96
9	422	311	733	9	0.98	0.98	0.92
10	212	481	693	37	0.95	0.95	0.78
11	622	1227	1849	72	0.95	0.96	0.80
12	284	958	1242	58	0.98	0.98	0.89
13	150	705	855	52	0.97	0.97	0.82
14	1027	543	1570	81	0.96	0.96	0.80
15	100	38	138	14	0.99	0.99	0.94
16	1708	479	2187	23	0.97	0.97	0.82
17	102	115	217	78	0.81	0.81	0.23
18	46	419	465	61	0.95	0.95	0.50
19	439	686	1125	56	0.97	0.97	0.86
20	606	1192	1798	53	0.89	0.89	0.52
21	239	1536	1775	19	0.93	0.93	0.43
22	1869	271	2140	81	0.97	0.97	0.71
23	446	1802	2248	44	0.90	0.90	0.43
24	929	2124	3053	94	0.97	0.97	0.87
25	74	820	894	38	0.96	0.96	0.45
26	501	312	813	81	0.92	0.92	0.67
27	338	2065	2403	36	0.95	0.95	0.58
28	2719	736	3455	146	0.90	0.90	0.43
29	1797	165	1962	141	0.93	0.93	0.30
30	534	1215	1749	38	0.89	0.89	0.48

Table C.2. List of event relevance judgment details and statistics (2 of 2)

Event	Relevant	Non-relevant	Total Judgments	Labeling time (hrs)	Agreement(%)	Overall Confidence (avg.)	Kappa
31	20	1932	1952	18	0.99	0.99	0.33
32	2980	308	3288	21	0.95	0.95	0.51
33	143	896	1039	33	0.96	0.96	0.65
34	593	1049	1642	138	0.84	0.84	0.34
35	56	2621	2677	26	0.99	0.99	0.55
36	181	1852	2033	42	0.96	0.96	0.56
37	312	412	724	39	0.85	0.85	0.40
38	594	1841	2435	28	0.97	0.97	0.83
39	202	1508	1710	15	0.96	0.96	0.66
40	283	1198	1481	15	0.96	0.96	0.71
41	943	405	1348	12	0.95	0.95	0.78
42	212	18	230	3	0.98	0.98	0.67
43	154	812	966	17	0.93	0.94	0.55
44	368	1026	1394	16	0.95	0.95	0.75
45	2823	29	2852	63	0.96	0.96	0.04
46	69	2105	2174	19	0.96	0.96	0.25
47	341	665	1006	31	0.91	0.91	0.62
48	363	1267	1630	43	0.94	0.94	0.67
49	3045	659	3704	202	0.89	0.89	0.39
50	295	2332	2627	22	0.95	0.95	0.53
51	763	2863	3626	36	0.95	0.95	0.72
52	1645	68	1713	110	0.95	0.95	0.18
53	86	2116	2202	14	0.96	0.96	0.24
54	254	2017	2271	14	0.97	0.97	0.69
55	113	3159	3272	16	0.98	0.98	0.42
56	917	1803	2720	371	0.90	0.90	0.54
57	964	2965	3929	129	0.93	0.93	0.65
58	980	4787	5767	110	0.96	0.96	0.71
59	3110	1627	4737	84	0.96	0.96	0.81
60	793	4504	5297	59	0.96	0.96	0.68
61	2009	2505	4514	119	0.85	0.85	0.39
62	12	3346	3358	28	0.98	0.98	0.05
63	523	3331	3854	34	0.97	0.97	0.70
64	1697	1300	2997	154	0.87	0.87	0.49
65	532	1146	1678	40	0.94	0.94	0.73
66	3619	517	4136	72	0.91	0.91	0.32