NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

BSc THESIS

Spatial Event Detection in Twitter: A Comparison of State-of-the-Art Techniques

Vasileios D. Papavasileiou

Supervisor: **Dimitrios Gunopulos,** Professor

ATHENS

JULY 2019

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Χωρική Ανίχνευση Γεγονότων στο Twitter: Μια Σύγκριση Τεχνικών Τελευταίας Τεχνολογίας

Βασίλειος Δ. Παπαβασιλείου

Επιβλέπων: **Δημήτριος Γουνόπουλος**, Καθηγητής

ΑΘΗΝΑ

ΙΟΥΛΙΟΣ 2019

BSc THESIS

Spatial Event Detection in Twitter: A Comparison of State-of-the-Art Techniques

Vasileios D. Papavasileiou

S.N.: 1115201400139

SUPERVISOR: **Dimitrios Gunopulos,** Professor

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Χωρική Ανίχνευση Γεγονότων στο Twitter: Μια Σύγκριση Τεχνικών Τελευταίας Τεχνολογίας

Βασίλειος Δ. Παπαβασιλείου
Α.Μ.: 1115201400139

ΕΠΙΒΛΕΠΩΝ: **Δημήτριος Γουνόπουλος,** Καθηγητής

ABSTRACT

During the last decade, social media platforms such as Twitter have grown and matured to a point where enormous amounts of data are being generated in real-time fashion. Users of these networks are constantly uploading information about the current state of their surroundings. This wealth of information can be exploited in order to provide meaningful real-time feedback about ongoing events for a wide range of applications.

In this thesis, we test two targeted domain, event detection systems; one that is supervised and one that is not. These event detection systems, process a collection of time-indexed data and perform the detection procedure as a pipeline of three discrete steps, a filtering phase, a spatial clustering phase, and a scoring phase. In the filtering phase, the tweets get processed, and they are divided into two categories, those that are related to the targeted domain and those that are not. The next step of the process is the spatial clustering, which aggregates the related tweets into areas of interest. As the last part of the process, the regions that were extracted by the clustering phase, are sorted, by ranking higher those regions that were mostly affected by the event. More specifically, the supervised system, requires human labor and a simple algorithm for filtering the tweets. By contrast, the unsupervised system employs an unsupervised algorithm for this process. Both systems cluster the related tweets, either with the k-Means algorithm, or with a modularity based graph clustering algorithm. Finally these systems rank the resulting clusters with the help of several ranking schemes. During our experiments, it became clear that the unsupervised system provides worse results than the supervised approach, but it does not require time for labeling the data, and thus it provides a good trade-off between human labor and accuracy of the results.

SUBJECT AREA: Data Mining

KEYWORDS: event, detection, spatial, clustering, k-Means, modularity, supervised, unsupervised

ΠΕΡΙΛΗΨΗ

Κατά την διάρκεια της τελευταίας δεκαετίας, οι πλατφόρμες κοινωνικής δικτύωσης όπως το Twitter έχουν αναπτυχθεί και ωριμάσει, με αποτέλεσμα τεράστιο πλήθος δεδομένων να δημιουργείται σε πραγματικό χρόνο. Οι χρήστες αυτών των δικτύων, μεταφορτώνουν συνεχώς δεδομένα σχετικά με την κατάσταση του περιβάλλον τους. Αυτή η πλούσια συλλογή δεδομένων μπορεί να χρησιμοποιηθεί για να προσφέρει ανάδραση πραγματικού χρόνου για ενεργά γεγονότα, η οποία μπορεί να αξιοποιηθεί με ποικίλους τρόπους.

Σε αυτήν την πτυχιακή εργασία, δοκιμάζουμε δύο συστήματα, ανίχνευσης γεγονότων στοχευμένου τομέα, ένα εκ των οποίων είναι με επίβλεψη, και το άλλο είναι χωρίς. Αυτά τα συστήματα ανίχνευσης γεγονότων, επεξεργάζονται μια συλλογή από χρονικά ταξινομημένα δεδομένα και εκτελούν την διαδικασία της ανίχνευσης γεγονότων ως μια σωλήνωση τριών διακριτών βημάτων, αυτό του φιλτραρίσματος, της χωρικής συσταδοποίησης και της βαθμολόγησης. Στο βήμα του φιλτραρίσματος, τα tweets, έπειτα από επεξεργασία, χωρίζονται σε δύο κατηγορίες, αυτά που είναι συναφή με τον τομέα, και αυτά που δεν είναι. Το επόμενο βήμα της διαδικασίας είναι η χωρική συσταδοποίηση των συναφών tweets σε συναφής περιοχές. Στο τελικό στάδιο της διαδικασίας, οι περιοχές οι οποίες είχαν εξορυχθεί κατά την διαδικασία της χωρικής συσταδοποίησης, ταξινομούνται με τέτοιο τρόπο, ώστε οι περιοχές οι οποίες είχαν επηρεαστεί περισσότερο από το γεγονός, να βαθμολογούνται υψηλότερα. Πιο συγκεκριμένα, το σύστημα με επίβλεψη, απαιτεί ανθρώπινη εργασία και έναν απλό αλγόριθμο για το φιλτράρισμα των tweets. Σε αντίθεση, το σύστημα χωρίς επίβλεψη υιοθετεί έναν αλγόριθμο χωρίς επίβλεψη για αυτήν την διαδικασία. Και τα δύο συστήματα συσταδοποιούν τα συναφή tweets, είτε με τον αλγόριθμο k-Means, είτε με έναν αλγόριθμο συσταδοποίησης γράφων με την χρήση της μετρικής modularity. Τέλος, αυτά τα συστήματα ταξινομούν τις συστάδες με βάση κάποιες στρατηγικές βαθμολόγησης. Κατά την διάρκεια των πειραμάτων μας, κατέστη σαφές πως το σύστημα χωρίς επίβλεψη, είχε χειρότερα αποτελέσματα από το σύστημα με επίβλεψη, αλλά δεν απαιτεί χρόνο για τον χαρακτηρισμό των δεδομένων, και με αυτόν τον τρόπο προσφέρει έναν καλό συμβιβασμό μεταξύ ανθρώπινης εργασίας και ακρίβειας των αποτελεσμάτων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εξόρυξη Δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: γεγονότα, ανίχνευση, χωρική, συσταδοποίηση, k-Means, modularity, επίβλεψη

To my family

ACKNOWLEDGEMENTS

I would like to express my very great appreciation to my supervisor and professor Dimitrios Gunopulos, for his valuable and constructive suggestions during the planning and development of this bachelor thesis. His willingness to give his time so generously has been very much appreciated.

I would also like to thank the members of the KDD lab, and especially Antonia Saravanou, for their time and valuable suggestions.

CONTENTS

1 INTRODUCTION 13

1.1 Background 13

1.2 Related Work 14

1.3 Our Objective 15

2 SYSTEMS OVERVIEW 16

2.1 Supervised Approach 16

2.1.1 Filtering 16

2.1.2 Clustering 17

2.1.3 Identifying Affected Areas 17

2.2 Unsupervised Approach 17

2.2.1 Filtering 17

2.2.2 Clustering 19

2.2.3 Identifying Affected Areas 21

3 EXPERIMENTAL RESULTS 23

3.1 Dataset Description 23

3.2 Evaluation of the Parameters 24

3.2.1 Supervised Approach 24

3.2.2 Unsupervised Approach 25

3.3 Comparing the two Systems 28

4 CONCLUSIONS AND FUTURE WORK 32

TABLE OF TERMINOLOGY 33

ABBREVIATIONS - ACRONYMS 34

REFERENCES 36

LIST OF FIGURES

1.1 Example of an event (floods) as described by BBC 13

2.1 Overview of the filtering process of the supervised approach 16

2.2 The Dynamic Query Expansion algorithm in pseudocode 18

2.3 Example of a clustering structure that a modularity algorithm would generate 20

2.4 The Spatial Modularity Clustering algorithm in pseudocode 20

2.5 Graph representation of the tweets before the clustering 21

2.6 Graph representation of the tweets after the clustering 21

3.1 One day heatmap 23

3.2 Spatial clusters generated by k-Means clustering 25

3.3 F-measure score in relation to the number of seed terms............ 25

3.4 One cluster with 100 km radius 27

3.5 Running average of the normalized Likert scores, supervised system........ 30

3.6 Running average of the normalized Likert scores, unsupervised system..... 31

LIST OF TABLES

1.1 Example of flood related tweets 13

3.1 Dataset Statistics 23

3.2 Lexicon Keywords 24

3.3 Results of the DQE algorithm, January 14 26

3.4 Example of similar tweets in the DQE algorithm 26

3.5 DQE runtimes in seconds for various parameters, January 13 27

3.6 SMC statistics for various parameters, January 13 27

PREFACE

The thesis at hand was undertaken as part of the course of study for the undergraduate degree at the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens. The relevant work was conducted from May 2018 to July 2019 in Athens under the supervision of professor Dimitrios Gunopulos.

1. INTRODUCTION

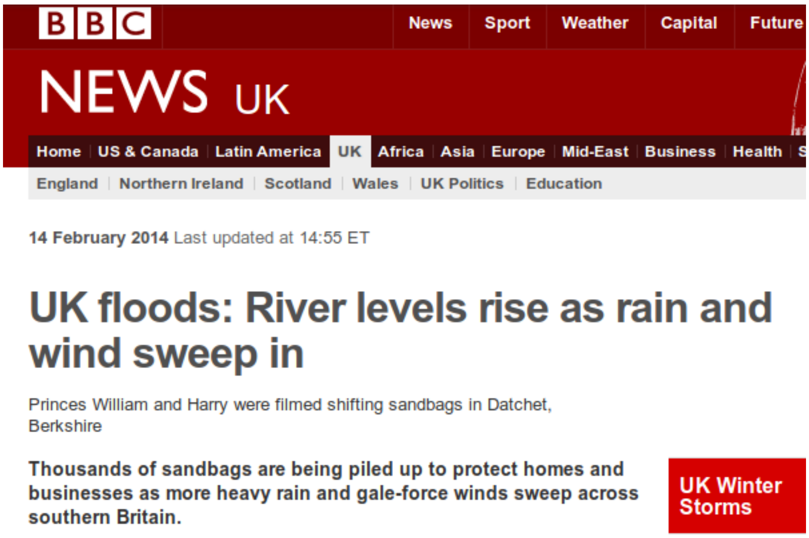
1.1 Background

Microblogging platforms like Twitter have experienced a significant growth in usage in the past years and they have accumulated millions of users. These users are constantly uploading short messages called “tweets” which are often enriched with links, media files, and tagging of other users, with content coming from a wide range of topics. The topics discussed in Twitter due to its more informal and personal tone, are similar to the ones discussed from person to person and thus they can vary greatly in significance to the material world. For example they can range from things that are purely abstract (e.g. discussing about movies and music) to more concrete events, either extreme (e.g. earthquakes, floods) or less so (e.g. strikes, elections). This wealth of real-time information can be exploited in order to provide meaningful real-time feedback about ongoing events for a wide range of applications.

Table 1.1: Example of flood related tweets

And its still raining. Dinghy to get us home. http://t.co/ma3uE6AIiT
@GSpellchecker @humanistdalek @AtheistBlobfish Help! It’s been raining for what seems like 40 days and nights! Should I build a big boat?
Walking home in the rain is so not fun
I hate the rain -_-. Had to cycle in the bloody rain. I really do hate the rain -_-
I’m sick of this cold weather! Will summer hurry up

The extraction of such events falls under the more general field of event detection. In general, the event detection domain varies greatly, in the way the problems are described, which are their goals-objectives and eventually their proposed solutions. For the purpose of this thesis we will define an event **e** as a real-world phenomenon that occurred at some specific time **t** and tied to a location **l**.



14 February 2014 Last updated at 14:55 ET

UK floods: River levels rise as rain and wind sweep in

Princes William and Harry were filmed shifting sandbags in Datchet, Berkshire

Thousands of sandbags are being piled up to protect homes and businesses as more heavy rain and gale-force winds sweep across southern Britain.

Figure 1.1: Example of an event (floods) as described by BBC

Unfortunately, automatically analyzing tweets to identify real-life events is a quite difficult task. This is due to the fact that:

1. The enormous amount of data that need to be processed in real-time fashion.
2. Not all tweets have their precise location tied to them, so techniques for extracting location indirectly are needed [20].
3. Tweets are often written in a very informal way, have a lot typos, unstructured language, slangs and acronyms. They also have a very high amount of spam content [7], and in general they can prove to be a very noisy source of information [5].

1.2 Related Work

In general most of the existing work done in event detection on microblogging platforms can be classified into two categories.

In the first category, the aim is to detect general-interest events as they appear without having any prior knowledge about their existence. Usually methods under this category apply unsupervised techniques such as topic modeling [22], burst detection [8, 9] and online clustering [1, 2, 13]. For example, Zhijun Yin et al. [22] created a framework which combines geographical clustering and topic modeling for geographical topic comparison. Jon Kleinberg [8] presented an approach for modeling the stream using an infinite-state automaton, in which, bursts appear naturally as state transitions. By following a different approach, Lappas et al. [9] explored several methods for detecting spatiotemporal term burstiness which can be used for trend identification and document extraction. Alvanaki et al. [1] proposed an online system which detects unusual shifts in correlations of various statistics, specific to tags and tag-pairs of Twitter data, caused by real-world events. In addition, Becker et al. [2], analyzed and compared various document similarity metrics which take in account both textual and non-textual features, to enable online clustering of media to events. Finally, Mathioudakis et al. [13] developed an online system which identifies and groups bursty keywords on Twitter, in order to identify trends. Of course other novel approaches exist, like relying on the emotional classification of the tweets combined with spatiotemporal information to identify events [21].

In the second category, the aim is to detect events within a particular domain (e.g. earthquakes [16], floods [17], civil unrest [23], diseases [19], crimes [10]). This kind of approach, usually requires supervised techniques in order to filter the relevant tweets to that particular domain and then apply clustering techniques to identify the locations of the events. It becomes immediately clear, that for the supervised methods to work successfully, a lot of human effort is required to label the training data. For example, Sakaki et al. [16], proposed a system which focuses on earthquakes and its main objective is to accurately extract the location of an earthquake by extracting earthquake related tweets with the help of a manually crafted lexicon, and by using a model that incorporates Kalman and particle filtering. In a similar way, Saravanou et al. [17] proposed a system which, with the help of a manually crafted lexicon and clustering techniques, identifies the areas that were mostly affected by floods. Contrary to the most approaches, Zhao et al. [23] developed a system which adopted an unsupervised algorithm to filter relevant tweets, and then applies a clustering algorithm which takes in account both semantic and geographical data, for detecting civil unrest events. Signori et al. [19] deployed an SVM classifier, along with other models, for tracking disease activity, while Li et al. [10] proposed an online system consisting of a trained classifier to extract crime related tweets and a ranking model for sorting the tweets by their importance in order to detect crime events.

The two approaches discussed above have different goals in mind, that serve different needs. On the one hand, general-interest event detection provides a way to detect events that we can't use descriptive terms like *#hashtags* (i.e., user generated topic labels) or keywords (e.g. ”protest”, ”demonstration”) to identify. On the other hand targeted-domain event detection provides a way to monitor only specific events that are of our interest. In addition, the latter approach allows for greater understanding of the event and better ways to quantify and interpret the data in hand.

1.3 Our Objective

The main objective of this thesis is to analyze and compare approaches that fall under the targeted domain event detection. More specifically, we will focus on two different approaches, one that is supervised [17] and one that isn't [23]. These two methods share a common structure. At first a bag of keywords is used in order to filter the relevant tweets. Then the relevant tweets are clustered into geospatial regions for further analysis. The main difference in these two approaches is how the bag of words is extracted. On the supervised approach the bag of keywords was extracted manually by the authors. In contrast, the unsupervised approach, applies an algorithm that is responsible for extracting the bag of words, given a very small seed query.

This key difference allows the unsupervised method to work on various domains with very little effort, since the only thing that needs to be altered is the seed query. This can be proved to be especially useful on extreme situations, where the timing is critical. Our intuition is that, this approach will result in worse results than the hand picked keyword approach since there is no human oversight. Therefore, we want to test the unsupervised approach on the same dataset (floods in UK [17]) as the supervised one, and compare the results. The above comparison will reveal, how well the unsupervised approach will perform against an optimal approach for the given dataset, and what is the trade off between performance and human intervention.

2. SYSTEMS OVERVIEW

In the following section we will provide an overview of the two different systems that were discussed above.

2.1 Supervised Approach

The supervised approach consists of several sequential steps. At first the data (tweets) is filtered based on the lexicon that was compiled by the authors, in order to keep the tweets that are relevant to that particular domain (extreme weather - floods in the UK in early 2014). After that, the whole collection of tweets, including those that are irrelevant to the domain, are used to create sub-regions in the geographical area of study with the help of clustering. Finally, for each of these sub-regions, various metrics are applied in order to quantify which of these regions endure the most extreme effects and therefore need the most assistance.

2.1.1 Filtering

In the filtering phase, the first step is to compile a custom lexicon. A small seed set of related tokens (13 in total) is created (e.g. rain, flood, weather). These related tokens are used in order to search and store keywords in tweets that contain them as sub-strings (excluding mentions, i.e. @username). As a result, a new set of words is extracted. Unfortunately, this approach yields a lot of false positives, and thus, careful review of the tokens is needed. After the review of the tokens, the custom lexicon is complete.

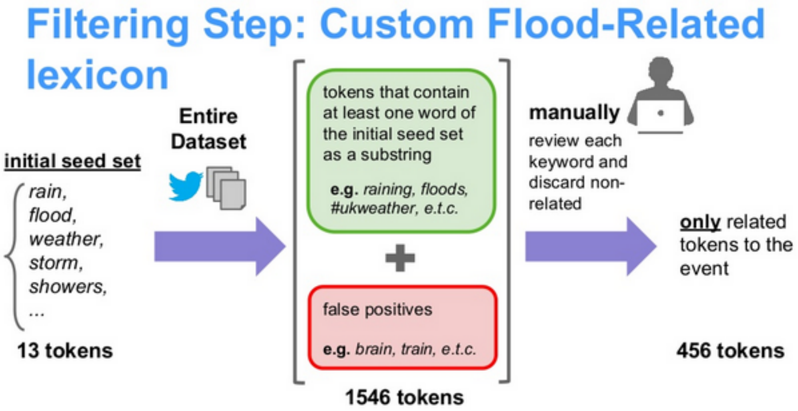

Filtering Step: Custom Flood-Related lexicon
initial seed set
rain,
flood,
weather,
storm,
showers,
...
13 tokens
Entire Dataset
tokens that contain at least one word of the initial seed set as a substring
e.g. raining, floods, #ukweather, e.t.c.
+
false positives
e.g. brain, train, e.t.c.
1546 tokens
manually review each keyword and discard non-related
only related tokens to the event
456 tokens

Figure 2.1: Overview of the filtering process of the supervised approach

As a second step, each tweet is processed again, and converted into an unordered set of words-tokens. Then, if any of these words-tokens match the custom lexicon, the tweet is classified as relevant and ready to be used for further analysis. It should also be noted that due to limitations and challenges in location extraction, all the tweets that don't have a geographical signature are omitted.

2.1.2 Clustering

As a result of the filtering process, each tweet has a location tied to it. Given that the goal is to identify regions that are affected, a need for aggregating the GPS coordinates of the tweets, becomes apparent. For this reason all the tweets of the dataset, including the unrelated ones, are clustered by their geographic location with the help of the k-Means clustering. The use of the whole dataset for geographical clustering, instead of using only the related ones, provides a better representation of the underlying population, since there are a lot more tweets. Before the clustering takes place, the GPS coordinates are first converted to Cartesian ones, using Mercatorian map projections. This step is necessary, because the k-Means clustering uses the Euclidean distance which works only on Cartesian systems.

2.1.3 Identifying Affected Areas

The next challenge after clustering, is to identify which regions were mostly affected, by the event. Depending on the number of clusters that is chosen, a high number of subregions may arise. For this reason the authors of the paper proposed several prioritization schemes, which sort the areas, by returning the most affected regions first, and the less affected regions afterwards. The prioritization schemes that were proposed and tested, are the following:

1. **The number of tweets**: This metric serves as a baseline. Essentially the regions are sorted by the number of tweets they have in a descending order, regardless if the tweets are related to the event or not.
2. **The number of related tweets**: The areas are sorted in a descending order, by the number of related tweets they have.
3. **SNR**: The areas are sorted in a descending order, by the ratio of related tweets that each region has. For example, in location r the ratio is:

$$\frac{\text{number of related tweets in } r}{\text{number of tweets in } r}$$

2.2 Unsupervised Approach

The unsupervised system retains a similar structure as the supervised one. The main difference is that, at first, an algorithm is responsible for transforming a small seed query to one that is more extensive, without any supervision. Then every tweet that matches the extended query is regarded as a related one, and is stored for the next step of the process. At last, a clustering phase takes place, which in contrast to the unsupervised implementation, it takes in consideration both the geographical locations and the semantic similarities of the tweets.

2.2.1 Filtering

As part of the initialization of the filtering process a graph representation of the data is built. This is in contrast to the bag of words representation of the other implementation.

More specifically, an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W}, \mathcal{S})$ is created, where $\mathcal{V} = \mathcal{T} \cup \mathcal{F}$. $\mathcal{T}$ refers to a set of tweet nodes, and $\mathcal{F}$ to a set of feature nodes. The tweet nodes are essentially identifiers of the tweets, while the feature nodes are nodes that represent other attributes of the tweets such as the words that the tweets are made of, hashtags, hyperlinks, users etc. The edges $\mathcal{E}$ refer to the various relationships between the nodes. These relationships reveal information about the author of a tweet (edge between a user node and a tweet node), the keywords a tweet contains (edge between a term node and a tweet node), the replying relationship between tweets (edge between two tweet nodes) and more. The $\mathcal{W}$ denotes the weights of the nodes of the graph. Finally $\mathcal{S}$ refers to the set of GPS locations of the tweets. At the same time as the graph generation takes place, the tweets get sanitized by removing stop words [12] (i.e. words that are common and don't offer any semantic value, e.g. ”The”, ”is”) and lemmatization [12] (i.e. a process for removing inflectional endings and returning the base or dictionary form of a word, e.g. ”cars” ⇒ ”car”). Moreover, a method for detecting near duplicate documents efficiently, known as simhash [11] is used for identifying and removing tweets that are very similar, as a preprocessing step.

The next task of the filtering process is to get a small seed query $\mathcal{Q}_0$ and generate an expanded one $\mathcal{Q}_p$. This task is handled by the Dynamic Query Expansion (DQE) algorithm. The DQE algorithm takes advantage of the heterogeneous relationships between the various entities of the tweets in order to calculate the weights of the nodes. The higher the weight a node has, the more relevant is to the targeted domain. For example terms like ”#UCLfinal” or ”Liverpool” are surely more related to the sport domain than a keyword like ”#EUElections2019”. In a similar way, a replying tweet is more likely to share the same domain as the tweet that is replying to. As a result the more relevant nodes get a higher weight since they are connected with other highly weighted nodes. By formalizing the above observation we have the following:

```
Algorithm 1: Dynamic Query Expansion (DQE)
Data: Seed Query $\mathcal{Q}_0 = \{(v_i, w(v_i)^{(0)})_{i=1}^{M}\}$
Result: Expanded Query $\mathcal{Q}_p$
Initialize T, F, $T_r^0$ and $w(T)$;
$k = 0$;
repeat
    repeat
        Swap(min($w(T_r^k)^{(k)}$), max($w(T - T_r^k)^{(k)}$));
        $\sigma$ = min($w(T_r^k)^{(k)}$) − max($w(T - T_r^k)^{(k)}$);
    until $\sigma \geq 0$;
    $w(F)^{(k)} = D_F \cdot A_{F,T} \cdot w(T)^{(k-1)}$;
    $w(T)^{(k)} = \Phi(A'_{F,T} \cdot w(F)^{(k)} + \beta A_T \cdot w(T)^{(k-1)})$;
    $\sigma$ = max($w(T - T_r^k)^{(k)}$) − min($w(T_r^k)^{(k)}$);
    $k = k + 1$;
until $\sigma \leq 0$;
$w(F_r) = \{(w(v_i)^{(k)}) \in w(F)^{(k)} \mid v_i \in F_r \subseteq F\}$;
$\mathcal{Q}_p = \{(v_i, w(v_i)) \mid v_i \in F_r, w(v_i) \in w(F_r)\}$;
```

Figure 2.2: The Dynamic Query Expansion algorithm in pseudocode

The DQE Algorithm: Given a seed query $\mathcal{Q}_0$ with its appropriate weight (1.0 for every word), all the tweets that match it are marked as related T_r^0. All the feature nodes $F \subseteq \mathcal{F}$

that match the related tweets T_r^0 are stored. In a similar way the tweet nodes $T \subseteq \mathcal{T}$ that match the F are kept for possibly being related to the domain (Line 1). After that a repetitive process begins. The tweets that are marked to be related T_r^k in the k-th repetition are compared to the ones that are not related $T - T_r^k$. If any of the tweets that belong to the set $T - T_r^k$ have a higher weight than tweets in the related set T_r^k, then they get swapped until all the related tweets have a higher weight (Lines 4-7). The following step is to update the weights of the features F and the tweets T. This is done via the equation on the Line 8 of the algorithm $w(F)^{(k)} = D_F \cdot A_{F,T} \cdot w(T)^{(k-1)}$, where D_F is the inverse document frequency (IDF) matrix of F and $\mathrm{A}_{F,T}$ is the adjacency matrix between F and T. At a similar way Line's 9 equation $w(T)^{(k)} = \Phi(A'_{F,T} \cdot w(F)^{(k)} + \beta A_T \cdot w(T)^{(k-1)})$ updates the weights of the features F with the help of the recently calculated weights of $w(F)^{(k)}$. $\mathrm{A'}_{F,T}$ is the transpose of the adjacency matrix between F and T. A_T is the adjacency matrix between tweets and it represents the replying relationship between them. β is a constant for choosing the balance between the influences of the various features and the replying relationships of the tweets. Φ is a function that normalizes the resulting matrix by column. Afterwards, the algorithm checks if there is a need for swapping the tweets between the related set and the non-related set as described above, and if the need arises, the loop begins again (Lines 3-12). Otherwise the algorithm comes to an end. As a result, a set of feature nodes that match the related tweet nodes are returned. This set of words is the resulting lexicon $\mathcal{Q}_p$ that will be used to extract the related tweets $T_{\mathcal{Q}_p}$.

2.2.2 Clustering

In a similar way as the supervised approach, a clustering method is in place for aggregating the various tweets locations into areas. For this reason a clustering algorithm that is based on the work of Liang Zhao et al. [23] is used. Unfortunately since the implementation of the algorithm could not be disclosed by the authors, due to intelligence property issues, we had to improvise and come up with a simpler algorithm, that we call: Spatial Modularity Clustering Algorithm (SMC).

Before exploring the SMC algorithm any further, we will define modularity for graphs based on the work of Newman et al. [14]. Following is a intuitive and simple definition of modularity provided by Shiokawa et al. [18]. The main idea of modularity algorithms is to find groups of vertices that have a lot of inner-group edges and few inter-group edges . Modularity Q is defined as follows:

Let e_{uv} be the total number of edges between cluster u and v; a_u be the total number of edges that are attached to vertices in cluster u; and m be the total number of edges in the whole graph. The following equation gives the modularity score of the clustering result.

$$Q = \sum_u \{ \frac{e_{uu}}{2m} - (\frac{a_u}{2m})^2 \}$$

In the definition above, $\frac{a_u}{2m}$ is the expected fraction of edges of u, which can be obtained when we assume the graph to be a random graph. Therefore, well clustered graphs will obtain high modularity scores, since the value of e_{uu} is highly different from the random graph.

The SMC Algorithm: Before using the algorithm we need to construct a new undirected graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0, \mathcal{W}_0, \mathcal{S}_0)$. In this case $\mathcal{V}_0 = T_{\mathcal{Q}_p}$ which are the related tweets that were calculated with the help of the DQE algorithm. $\mathcal{E}_0$ represents a set of undirected edges between all the related tweets. $\mathcal{W}_0$ is essentially the weight set w($\mathcal{E}_0$), and it represents

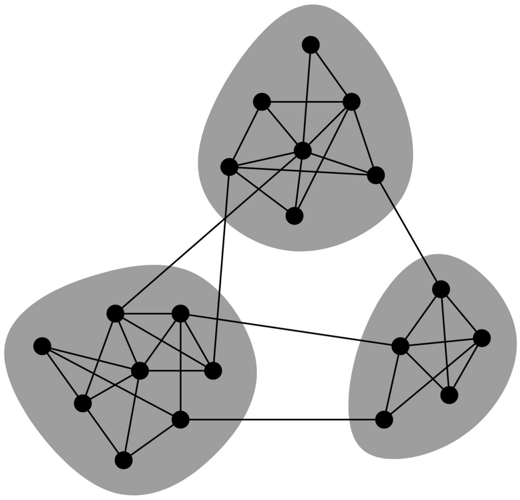


Figure 2.3: Example of a clustering structure that a modularity algorithm would generate

```
Algorithm 2: Spatial Modularity Clustering Algorithm
   Data: $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0, \mathcal{W}_0, \mathcal{S}_0)$
   Result: $\Omega = \{\mathcal{G}(V_i)\}_{(i=1)}^{K}$, where $V_i \subseteq \mathcal{V}_0$
1  Initialize $\Omega = \emptyset$;
2  for $s \in \mathcal{S}_0$ do
3      $\mathcal{V}_s = \{v \mid v \in \mathcal{V}_0, l(v) \in \mathcal{S}_s\}$;
4      Cluster $\mathcal{V}_s$ while maximizing the modularity score;
5      Add the above Clusters to $\Omega$;
6      if overlapping Clusters in $\Omega$ then
7          Keep the Clusters with the highest modularity score;
```

Figure 2.4: The Spatial Modularity Clustering algorithm in pseudocode

semantic similarities between the tweets of the graph. Semantic similar tweets are tweets that share expanded query terms F_r. The expanded query terms F_r are the whole set of features that are part of the $T_{\mathcal{Q}_p}$ tweets. A more formal explanation of the weight set w$(\mathcal{E}_0)$ is that, if A is the adjacency matrix between $T_{\mathcal{Q}_p}$ and F_r, then w$(\mathcal{E}_0) = A \cdot A'$. The $\mathcal{S}_0$ represents the GPS locations of the tweets.

The general idea behind this algorithm, is to find regions that have at the same time, proximate and semantically similar tweets. The SMC algorithm consists of a repetitive procedure (Lines 2-7), where a location s is chosen as a center (Line 2). Consequently, all the locations that are in range r from the center s are aggregated to the set $\mathcal{S}_s$. Then every tweet v that shares a location $l(v)$ with the set $\mathcal{S}_s$ is combined to a subgraph $\mathcal{V}_s \subseteq \mathcal{V}_0$ (Line 3). Immediately after that, the subgraph $\mathcal{V}_s$ gets clustered with the help of a modularity clustering algorithm [4] which takes into account the weights w$(\mathcal{E}_s)$ (Line 4). The new clusters are added to the set of clusters (Line 5). If any of the new clusters have overlapping tweets with the clusters that were done in previous iterations, then the ones that have better modularity score are kept and the other ones, are discarded (Lines 6-7). As a result we get a set of non-overlapping clusters. This means that each cluster has distinct tweets, but this doesn't imply that clusters can't overlap geographically or in other words, share GPS locations.

For example, given the graph in the figure 2.5, we pick the node 0 as its center. Every node that is in range r from the center (inside the red circle) is considered for the clustering. All the other nodes are pruned. In the next figure 2.6, the clustering takes place, which results in 3 clusters (red, blue and green) and a score representing the quality of the clustering. These 3 clusters are added to the set Ω. If there are conflicts (e.g. clusters share the same nodes), the clusters with the highest score are kept in the set Ω. This process begins again with a different center, until all the nodes are considered.

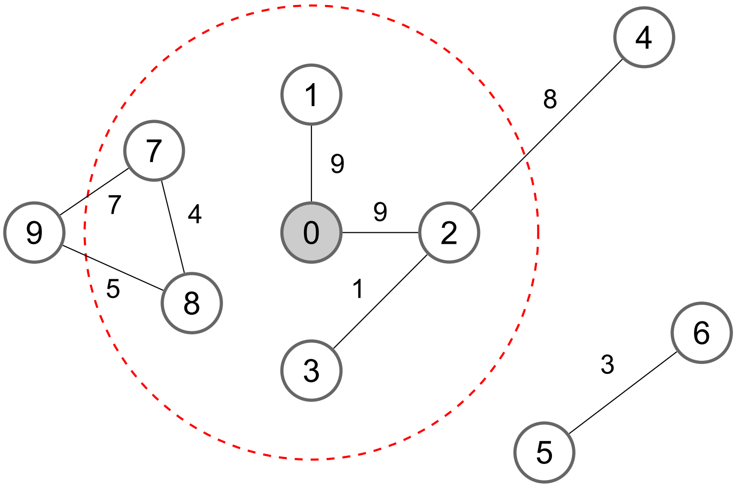


Figure 2.5: Graph representation of the tweets before the clustering

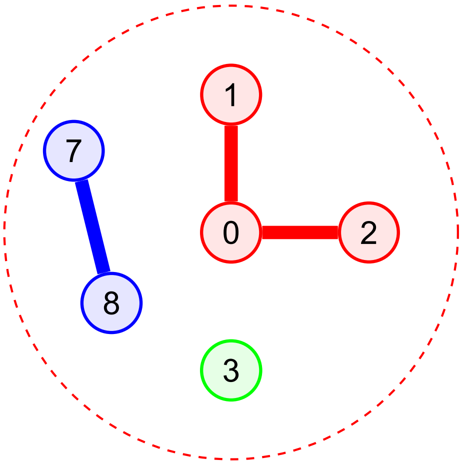


Figure 2.6: Graph representation of the tweets after the clustering

2.2.3 Identifying Affected Areas

In the original paper of Liang Zhao et al. [23], simple metrics like accuracy, recall and F-score were used to evaluate the performance of the system, but no prioritization schemes for the generated clusters were presented. In our case, since we want to compare the performance of this approach, with the work of Saravanou et al. [17], we adopt several prioritization schemes:

1. **The number of related tweets**: The areas are sorted in a descending order, by the number of related tweets they have.
2. **Modularity score**: The areas are sorted in a descending order, by the modularity score of their clustering.

In contrast to the previous system, we don’t apply the ”SNR” and ”number of tweets” prioritization schemes. That is, because both schemes, use the total number of tweets per area (related tweets + not related tweets), which apart from being inefficient to calculate in the unsupervised system, they can’t possibly take into account geographically overlapping clusters.

3. EXPERIMENTAL RESULTS

In this section we explore the dataset that is used for the experiments, and elaborate on the experimental evaluation of the performance, of the two systems that were discussed above.

3.1 Dataset Description

The dataset used for the evaluation of the two systems, consists of tweets collected in the whole area of the United Kingdom (UK), from January 13, 2014 to January 17, 2014. In this 5-day period, floods took place in the UK and more than 2.3 million geotagged tweets were collected, amounting to 469.9 MiB of data. The first and last days contain about half of the number of tweets than the rest. Each tweet of the dataset comes with several columns of data including a unique ID, a date, the text of the tweet and its GPS coordinates. Unfortunately the dataset doesn't include the replying relationship between the tweets, and for this reason we had to tune our DQE implementation in order to work without them.

Table 3.1: Dataset Statistics

Period	Total number of Tweets	Flood Related Tweets
January 13	351140	2728
January 14	577151	4973
January 15	569108	4159
January 16	578553	4994
January 17	275358	3490
Total	2351310	20344

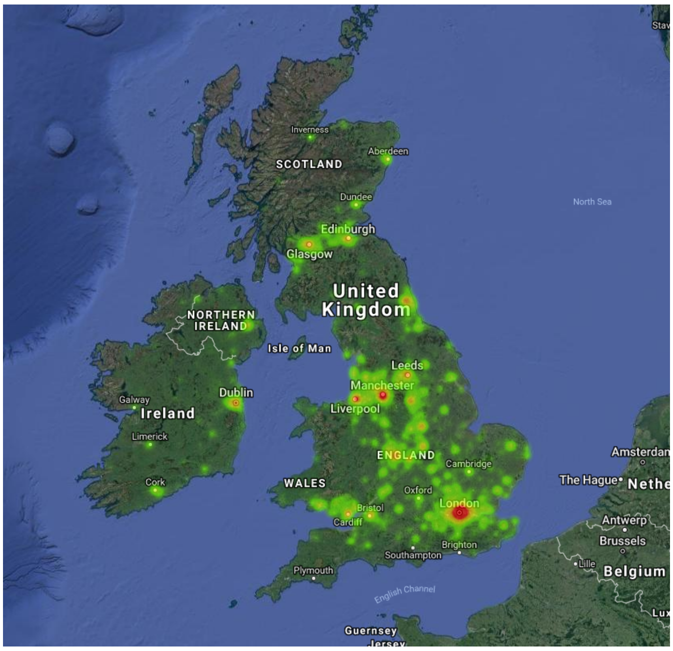


Figure 3.1: One day heatmap

In order to evaluate the results of the two systems, two independent sources are used for providing ground truth information. The first source is the ”Hydrological Summary for the United Kingdom" [3], published by Centre for Ecology & Hydrology, of the Natural Environment Research Council in the UK. The second source was published by the "Met Office" [15], UK's National Weather Service, as a response to the storms and floods that hit the UK, at the same period as our dataset.

3.2 Evaluation of the Parameters

In the following section, we will provide the results, of the evaluation, of the parameters, of the two systems that were studied in this thesis.

3.2.1 Supervised Approach

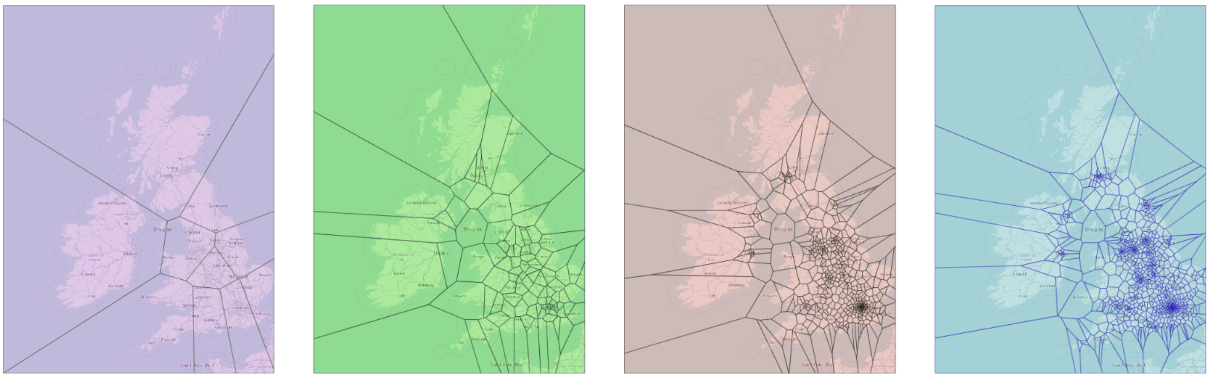
During the filtering phase of the supervised system, a small seed set of 13 related tokens, was used to extract other tokens, that contain them as substrings and are possibly related to the event we monitor (e.g. floods and storms in the UK). This approach yielded 1456 distinct keywords. After the time consuming process of labelling, the authors of the supervised approach, concluded that, from these keywords, only the 456 were related to the event. This resulting lexicon, could be possibly used for other similar events in the future, but we can't deny, that expression in tweets constantly evolves and this method may not work as well as intended, for time sensitive events that can't be easily described by general keywords, like elections, sport events etc.

Table 3.2: Lexicon Keywords

	Original Lexicon		**Flood Lexicon**	
Rank	**Keyword**	**Occurrences**	**Keyword**	**Occurrences**
1	*rain*	11235	*rain*	11235
2	*train*	6499	*weather*	3331
3	*training*	4593	*snow*	1006
4	*weather*	3331	*raining*	997
5	*brain*	1747	*rainbow*	419
6	*trains*	1251	*storm*	333
7	*snow*	1006	*showers*	273
8	*raining*	997	*rainy*	249
9	*trainers*	813	*flooding*	215
10	*drained*	435	*flooded*	214

As the next step of the system, the k-means clustering method, was used for aggregating the tweets into areas. One drawback of using the k-means, is the need to find the most appropriate number of clusters. For this reason, several numbers of k were used for experimentation (k = 10, 100, 500, 1000). Even though, the selection of k's is relatively small, other clustering approaches, would require even more experimentation due to the overwhelming number of parameters that need to be tuned (e.g. DBScan [6]).

By observing the generated clusters, it becomes clear, that the higher the k is, the more splits are generated in densely populated areas. In contrast, more rural and sparsely populated areas get less splits. This analysis could be interpreted as a kind of hierarchical clustering, by cutting the hierarchical dendrogram at different levels.



(a) 10 clusters (b) 100 clusters (c) 500 clusters (d) 1000 clusters

Figure 3.2: Spatial clusters generated by k-Means clustering

3.2.2 Unsupervised Approach

Our DQE implementation has several variables that can be fine-tuned. At first, the number of the seed query terms can vary in size. Given the throughout analysis carried out by the original authors of the paper of the DQE algorithm [23], we will use the proposed size of 5 keywords, as our seed query. As we can see from the following figure, coming from the civil unrest domain, for the most cases, the F-Measure is getting considerably higher as N increases from 1 to 3, but for values, from 5 and up, the F-Measure becomes stable.

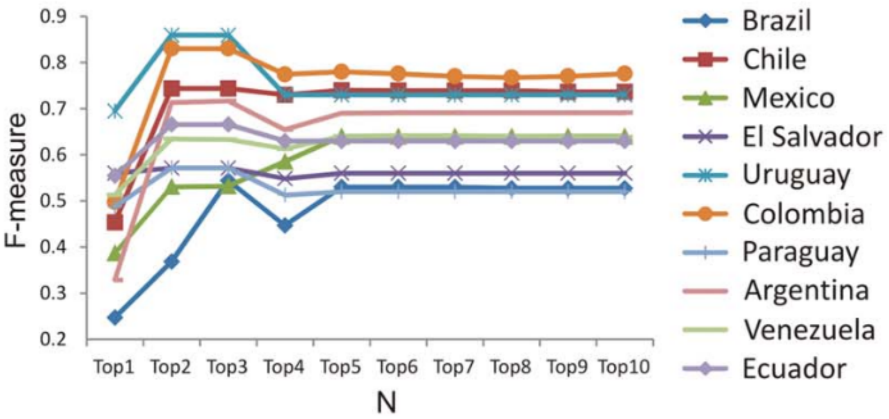


Figure 3.3: F-measure score in relation to the number of seed terms

In our case, we will use the top 5 most popular keywords (rain, weather, snow, raining, storm), that were extracted by the compiled lexicon of the supervised approach, as a best case scenario, excluding the word ”rainbow” since its not so relevant to the experiment's domain.

As we have mentioned before, the DQE algorithm, has a variable β for choosing the balance between the influences of the various features and the replying relationships of the tweets. However, due to, limitations coming from our dataset, we will have to set β to 0, which means that the replying relationships won't get considered.

The original DQE algorithm doesn't have a way to limit the number of iterations, since it relies on the property of the algorithm, which is that, it always converges after a small number of iterations. In support of this property, the authors of the DQE algorithm, observed in their experiments, that the algorithm itself, would stop after very few iterations and would provide good results. In our experiments, even though the algorithm would indeed stop at 6 to 12 iterations at most, the results of our analysis indicated that more than 1-2 iterations would result in very mediocre results. After a manual inspection of the dataset's related tweets, we noticed that, they shared very few features (e.g. hashtags and keywords), and were very diverse in their expression and content. This, in combination with the incomplete dataset (lack of replying relationships and authorship) may have contributed to the fact, that only very few iterations of the DQE are useful. This comes in contrast to the results that the original authors got, when they deployed the algorithm on a civil unrest dataset, in which the top extracted keywords contained a lot of hashtags.

Table 3.3: Results of the DQE algorithm, January 14

Iterations	**Top 5 keywords**				
1	rain	weather	snow	c	wind
2	rain	today	c	like	wind
3	like	get	day	today	good
7 (last one)	get	like	day	one	good

In addition to the previous observations, during our testing, we noticed that a lot of similar tweets were appearing in the filtering process of the system. Since the DQE algorithm was set to identify weather related tweets, the majority of these similar tweets fell under the category of automated weather reports. Since these reports don't add any value to our system, and in a sense they could be perceived as false positives, we deployed an extra preprocessing step for removing them. For this reason we used the simhash [11] algorithm, which provides an efficient way to identify similar documents. As a result, this kind of tweets were reduced by up to 86%.

Table 3.4: Example of similar tweets in the DQE algorithm

05:57 GMT: Temperature: 2.4°C, Wind: NNW, 3 mph (ave), 8 mph (gust), Humidity: 78%, Rain (hourly) 0.0 mm, Pressure: 1007 hPa, rising slowly
11:57 GMT: Temperature: 5.5°C, Wind: NNW, 2 mph (ave), 8 mph (gust), Humidity: 72%, Rain (hourly) 0.0 mm, Pressure: 1010 hPa, rising slowly
11:58 GMT: Temperature: 5.5°C, Wind: NNW, 2 mph (ave), 8 mph (gust), Humidity: 72%, Rain (hourly) 0.0 mm, Pressure: 1010 hPa, rising slowly
05:27 GMT: Temperature: 2.5°C, Wind: N, 2 mph (ave), 8 mph (gust), Humidity: 78%, Rain (hourly) 0.0 mm, Pressure: 1007 hPa, rising slowly
05:32 GMT: Temperature: 2.5°C, Wind: N, 2 mph (ave), 8 mph (gust), Humidity: 78%, Rain (hourly) 0.0 mm, Pressure: 1007 hPa, rising slowly

By testing our DQE implementation with the floods dataset, we observed that the algorithm is efficient in runtime, regardless of the dataset's input size. For example in the table 3.5, we noticed that for the 10% of the dataset (approximately 35k tweets), the runtime was below 2 seconds and for the 100% of the same datase, the runtime didn't exceed the 1 minute mark.

In the clustering phase of the unsupervised system, the expanded query, previously calculated by the DQE algorithm, is used to create a graph, which the SMC algorithm will cluster. The resulting expanded query gets cut-down in size, since a lot of the words in the expanded query are irrelevant to the domain, which is evident by the very low score

Table 3.5: DQE runtimes in seconds for various parameters, January 13

Iterations	10% of the dataset	100% of the dataset
1	0.1555	1.8984
2	0.3017	6.0543
3	0.6134	28.0743
7 (last iter. for the 10%)	1.1700	39.7210
10 (last iter. for the 100%)	-	43.1971

that most of these keywords have. For this reason, we keep only the top 10-15 keywords, for further analysis. All the tweets that contain at least one of these keywords are used for the clustering. As we can see below, in the table 3.6, by increasing the number of keywords from 10 to 15, we get an almost 10-fold increase in the number of tweets that are used for the clustering. As a result, the clustering runtime increases exponentially, partly due to the inefficient calculation of the distances between all the locations ($O(n^2)$). More efficient data structures, targeted specifically for spatial data, could help mitigate this problem, but they are not in the current scope of this thesis.

Table 3.6: SMC statistics for various parameters, January 13

	Runtime		Memory		Tweets	
Keywords	**10%**	**100%**	**10%**	**100%**	**10%**	**100%**
5	4 s	63 s	218.6 MiB	739.8 MiB	92	771
10	56 s	443 s	255.5 MiB	997.6 MiB	758	1916
15	960 s	100 h >	760.3 MiB	5.8 GiB	2387	13629

The SMC algorithm, has only one parameter which can be altered directly. This parameter is the distance r; the maximum radius that each cluster should maintain from its center. By altering this value, we can control the surface area of the clusters to our preference. In the following section, we will study how the radius, among other things, affects the quality and the quantity of the clusters.

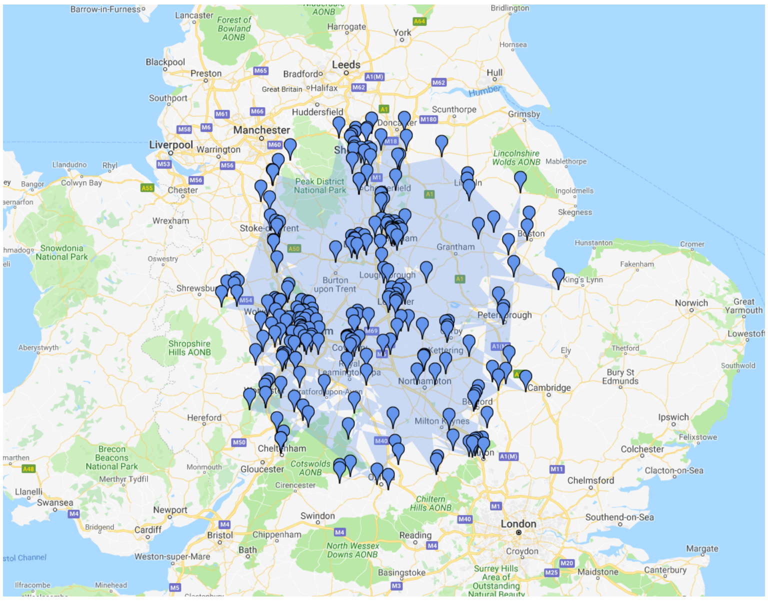


Figure 3.4: One cluster with 100 km radius

3.3 Comparing the two Systems

After processing the twitter data, both systems, return a set of clusters, containing tweets related to the domain that we monitor. In order to evaluate the clusters produced by the two systems, we follow several steps. At first, for both systems, we select the top-n clusters (n = 100 in our case), as they were ordered by the prioritization schemes of each system. Then we manually inspect all the ordered clusters (top-n), for each system, and we compare them against the ground truth information. More specifically, we score each cluster with the help of a Likert scale from 1 to 5, which represents how much these areas were affected by the floods, with "1" meaning not at all affected and "5" representing completely flooded areas. Finally, we compute the running average of the scores up to the i-th ranked cluster, with the following formula:

$$value_i = \frac{\sum_{j=1}^{i} \frac{likert\_score(j)}{5}}{i}$$

During the evaluation of the two systems, the entire collection of tweets (2.3M) is used. However, in order to keep the runtime of the unsupervised system within practical limits, we had to make an exception and use only the 10% of the whole collection as its input.

By applying the evaluation procedure as it was described above, the authors of the supervised system, compared the various prioritization schemes that they proposed.

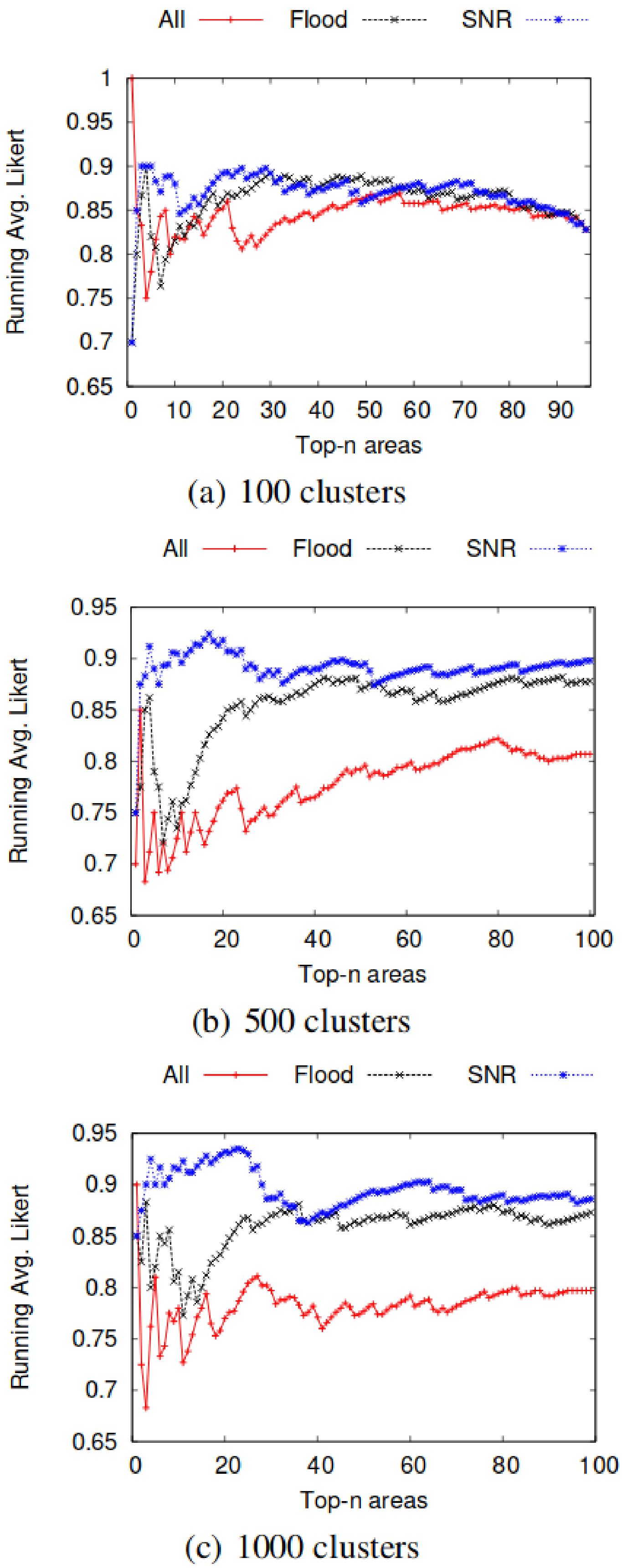
The results of this evaluation are given in the figure 3.5. The three schemes that were compared were: the number of tweets per area (All), the number of flood-related tweets (Flood) and the Signal-to-Noise Ration (SNR). We can clearly observe that for the 100 clusters generated by the k-means, the schemes All and Flood behaved in a similar way, while SNR has slightly better results than the other two schemes, at least for the first 30-40 areas. However, these differences blur after the first 50 areas. For k=500 and k=1000, the SNR clearly outperforms the two other schemes, especially on the first 30 areas. The Flood scheme catches up with SNR, after the first 30 areas, in contrast to the All scheme, which stays at considerably lower levels than the rest. Saravanou et al. [17] concluded, that the number of social media activity on an area, is not a reliable way to measure the impact of an event. The number of related events is a better indicator, but SNR performs even better, since it takes in account the number of users in that area. SNR, maintains an average of 0.9 for the top-100 areas, and never drops below 0.85.

The same evaluation was applied to the unsupervised system. In this case, two schemes are compared: the number of flood-related tweets (Flood) and the modularity score (Modularity). Contrary to the supervised system, the unsupervised approach doesn't have a direct control to the number of clusters that will be generated. The only way to influence the number of clusters, is by altering the maximum radius, that each cluster should maintain. For this reason, we experimented with three different values of radii (r= 50km, 100km, 150km) as shown in the figure 3.6.

The results of the experimentation, clearly indicate that the Flood scheme is superior to the Modularity one, across all cluster sizes. By setting the radius r to 50km, we observe a heavy difference between the two schemes for the first top 50 areas, until they blur together. The divide is even greater for r=100km, 150km, where the Modularity scheme stays consistently below the Flood one, in almost all the top-n areas. There is clear evidence, that the Modularity scheme is inconsistent and not suitable for evaluating the impact of an event. This means, that even if a clustering is well scored, the subsequent clusters, are not more relevant to the event than those that belong to a lesser scored clustering. On the other hand, the number of related tweets, provides an acceptable running average of 0.7 and never drops below 0.57. Moreover, we notice that the Flood scheme, performs similarly in the three different values of r that we tested. At first, the running average is around 0.80 and then it drops gradually to around 0.60 - 0.65. Another thing to note here, is that by increasing the maximum radius of each cluster, we get less clusters as a result. After evaluating the produced clusters, we have the insight, that at least for this dataset, the results for r=50km and r=100km are more useful than those of r=150km. For r=150km, the clusters are too wide to actually provide a meaningful representation of the affected areas.

As we expected, the supervised system performed considerably better than the unsupervised one. With the best prioritization scheme for each system, we had a running average of 0.9 for the supervised and 0.7 for the unsupervised one. This comes to no surprise, since the supervised approach, takes advantage of an extensive handpicked lexicon, deriving from the same dataset as the one that was used during the evaluation. In addition, several limitations and properties of the floods dataset, such as the lack of replying relationships, the wide variety in expression and limited use of common features (e.g. hashtags) in the the related tweets, lead to a non-ideal environment for the filtering algorithm of the unsupervised system, which has admittedly hindered its performance. Moreover, we had to improvise and implement a simpler algorithm for the clustering phase of the unsupervised system, which may have affected the system's performance too.

By analyzing the current literature, we have the intuition, that the supervised approach could be used for similar events in the future (floods), or be adopted in domains where a static historical dataset could still be utilized for classifying tweets. However, in dynamically evolving domains, the static datasets are of limited use, and this is an area where the unsupervised system could be deployed and provide useful results.



(a) 100 clusters

(b) 500 clusters

(c) 1000 clusters

Figure 3.5: Running average of the normalized Likert scores, supervised system

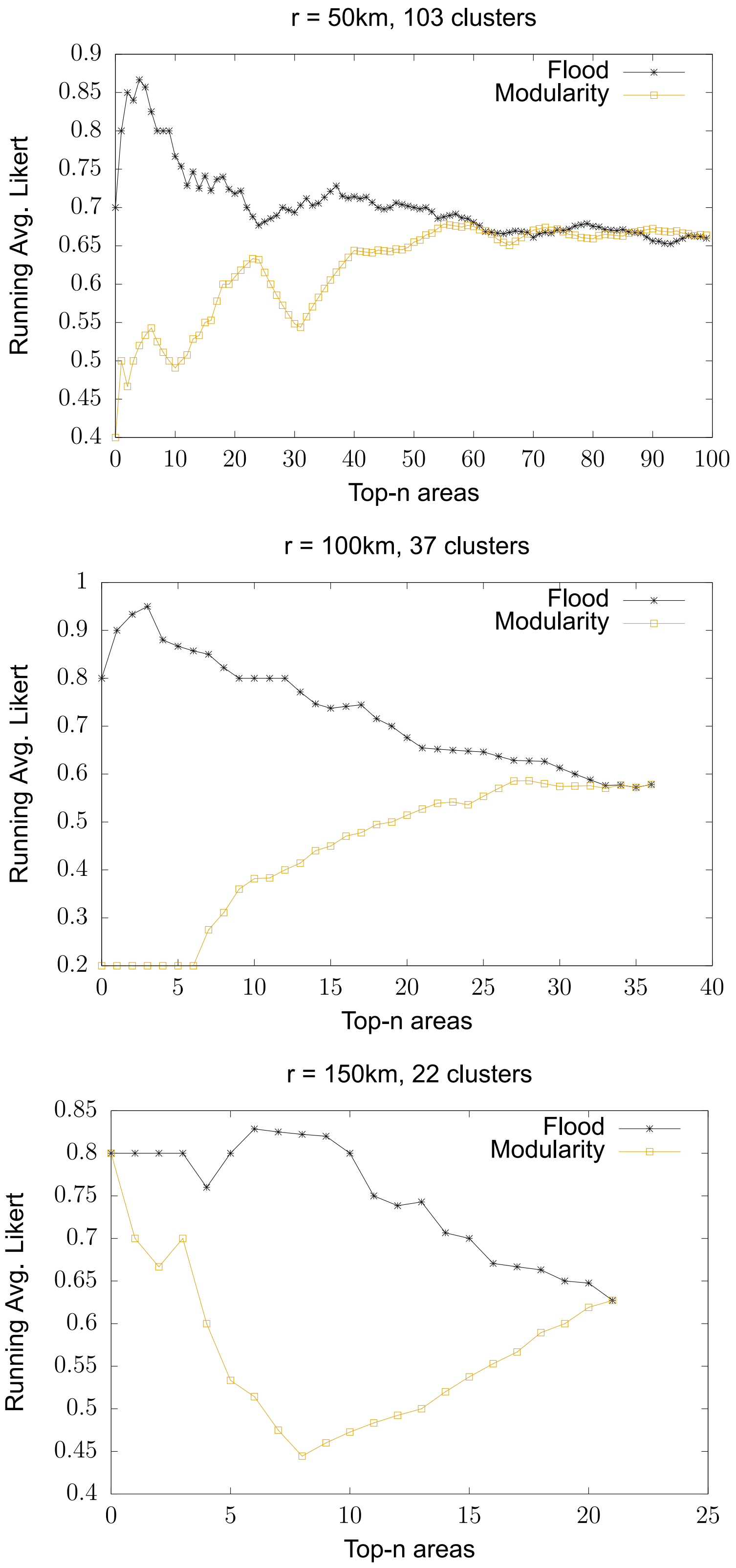


Figure 3.6: Running average of the normalized Likert scores, unsupervised system

4. CONCLUSIONS AND FUTURE WORK

In this thesis, we compared two different approaches, on spatial event detection in Twitter. More specifically, we implemented an unsupervised system which was based on the work of Zhao et al. and we compared it against a supervised system designed by Saravanou et al. We utilized a tweet dataset, extracted during a 5-day period, in which floods took place in the United Kingdom, for testing purposes. The goal of both systems was to detect areas which were hit by the floods, and at the same time prioritize the most affected ones. Both systems, apply a three stage pipeline in which, they filter the Twitter data, they cluster the filtered data and then finally rank the resulting clusters. In all of the above steps, we analyzed the performance of the two systems while taking into account the various parameters of each system. We came to the conclusion, that by altering the parameters we could greatly influence the performance of the two systems, especially the unsupervised one. Moreover, we observed that the supervised system was superior to the unsupervised one, which was to be expected, since the former was taking advantage of an extensive handpicked lexicon which was extracted by the evaluation dataset. At the same time, several limitations of our dataset and implementation, degraded the performance of the unsupervised approach, which further increased the performance gap between the two systems. Even though the performance of the unsupervised system was lesser compared to the supervised one, it employs several advantages, such as being easy to target different domains with minimal effort and being able to work efficiently on ever-changing domains. On the other hand, the supervised approach, is more suited for domains, where static datasets are successful at filtering relevant data (e.g. floods, earthquakes).

In our future work, we would like to address some of the shortcomings of our unsupervised system implementation. More specifically, we want to apply better data structures in the clustering phase of the unsupervised system, in order to speed up its computations. Furthermore, we would like to experiment with different graph clustering techniques and evaluate their performance. In addition, it would be crucial to deploy and test the system on different domains, with complete datasets and develop online or streaming capabilities. Finally, more prioritization schemes could be assessed for better ranking performance.

TABLE OF TERMINOLOGY

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Adjacency	Γειτνίαση
Automaton	Αυτόματο
Cluster	Συστάδα
Geotag	Γεωγραφική Ετικέτα
Lemmatization	Λημματοποίηση
Modeling	Μοντελοποίηση
Pipeline	Σωλήνωση
Preprocessing	Προεπεξεργασία
Real-time	Σε Πραγματικό Χρόνο
Recall	Ανάκτηση
Spatial	Χωρικός
Spatiotemporal	Χωροχρονικός
Stream	Ροή
Supervised	Με Επίβλεψη
Unsupervised	Χωρίς Επίβλεψη

ABBREVIATIONS - ACRONYMS

BBC	British Broadcasting Corporation
DQE	Dynamic Query Expansion
GPS	Global Positioning System
ID	Identification
IDF	Inverse Document Frequency
SMC	Spatial Modularity Clustering
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machines
UK	United Kingdom

BIBLIOGRAPHY

[1] Foteini Alvanaki, Sebastian Michel, Krithivasan Ramamritham, and Gerhard Weikum. See what's enblogue: Real-time emergent topic identification in social media. *ACM International Conference Proceeding Series*, 03 2012.

[2] Hila Becker, Mor Naaman, and Luis Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 291–300, New York, NY, USA, 2010. ACM.

[3] Natural Environment Research Council Centre for Ecology & Hydrology. Hydrological Summary for the United Kingdom. `https://web.archive.org/web/20140720111743/http://www.ceh.ac.uk/data/nrfa/nhmp/hs/pdf/HS_201401.pdf`. [Accessed on 2019-06-16].

[4] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004.

[5] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 198–206, Hissar, Bulgaria, September 2013. INCOMA Ltd. Shoumen, BULGARIA.

[6] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[7] Chris Grier, Kurt Thomas, Vern Paxson, and Michael Zhang. @spam: The underground on 140 characters or less. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, pages 27–37, New York, NY, USA, 2010. ACM.

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

[9] Theodoros Lappas, Marcos R. Vieira, Dimitrios Gunopulos, and Vassilis J. Tsotras. On the spatiotemporal burstiness of terms. *CoRR*, abs/1205.6695, 2012.

[10] R. Li, K. H. Lei, R. Khadiwala, and K. C. Chang. Tedas: A twitter-based event detection and analysis system. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1273–1276, April 2012.

[11] Gurmeet Singh Manku, Arvind Jain, and Anish Das Sarma. Detecting near-duplicates for web crawling. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 141–150, New York, NY, USA, 2007. ACM.

[12] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Harvesting ambient geospatial information from social media feeds*. Cambridge University Press, 2008.

[13] Michael Mathioudakis and Nick Koudas. Twittermonitor: Trend detection over the twitter stream. pages 1155–1158, 01 2010.

[14] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, Feb 2004.

[15] Met Office. The recent storms and floods in the UK. `https://web.archive.org/web/20140223174006/http://www.metoffice.gov.uk/media/pdf/n/i/Recent_Storms_Briefing_Final_07023.pdf`. [Accessed on 2019-06-16].

[16] 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*, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM.

[17] Antonia Saravanou, George Valkanas, Dimitrios Gunopulos, and Gennady Andrienko. Twitter floods when it rains: A case study of the uk floods in early 2014. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 1233–1238, New York, NY, USA, 2015. ACM.

[18] Hiroaki Shiokawa, Yasuhiro Fujiwara, and Makoto Onizuka. Fast algorithm for modularity-based graph clustering, 2013.

[19] Alessio Signorini, Alberto Maria Segre, and Philip M. Polgreen. The use of twitter to track levels of disease activity and public concern in the u.s. during the influenza a h1n1 pandemic. *PloS one*, 6(5):e19467–e19467, May 2011. 21573238[pmid].

[20] Andrew; Radzikowski Jacek Stefanidis, Anthony; Crooks. Harvesting ambient geospatial information from social media feeds. *GeoJournal*, 78:319–338, 2013.

[21] George Valkanas and Dimitrios Gunopulos. How the live web feels about events. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 639–648, New York, NY, USA, 2013. ACM.

[22] Zhijun Yin, Liangliang Cao, Jiawei Han, Chengxiang Zhai, and Thomas S. Huang. Geographical topic discovery and comparison. pages 247–256, 01 2011.

[23] Liang Zhao, Feng Chen, Jing Dai, Ting Hua, Chang-Tien Lu, and Naren Ramakrishnan. Unsupervised spatial event detection in targeted domains with applications to civil unrest modeling. *PloS one*, 9:e110206, 10 2014.