NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

BSc THESIS

Event Detection in Twitter: An Experimental Comparison

Panagiotis T. Sioulas
Konstantinos G. Tsitsimpikos

Supervisor: **Dimitrios Gunopulos, Professor**

ATHENS

JUNE 2017

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανίχνευση Γεγόνοτων στο Twitter: Μια Πειραματική Σύγκριση

Παναγιώτης Θ. Σιούλας
Κωνσταντίνος Γ. Τσιτσιμπίκος

Επιβλέπων: **Δημήτριος Γουνόπουλος, Καθηγητής**

ΑΘΗΝΑ

ΙΟΥΝΙΟΣ 2017

BSc THESIS

Event detection in Twitter: An Experimental Comparison

Panagiotis T. Sioulas
S.N.: 1115201300161

Konstantinos G. Tsitsimpikos
S.N.: 1115201300187

SUPERVISOR: **Dimitrios Gunopulos, Professor**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανίχνευση Γεγονότων στο Twitter: Μια Πειραματική Σύγκριση

Παναγιώτης Θ. Σιούλας
Α.Μ.: 1115201300161

Κωνσταντίνος Γ. Τσιτσιμπίκος
Α.Μ.: 1115201300187

ΕΠΙΒΛΕΠΟΝΤΕΣ: Δημήτριος Γουνόπουλος, Καθηγητής

ABSTRACT

The recent rise to prominence for social network, especially Twitter, has resulted in the availability of data of unprecedented scale concerning the content and the interactions produced by users in the context of the network. This source of information can be utilized for a range of applications, including the detection of events which involves determining a time interval along with a set of content entries that are related to a real-world event.

In this thesis, we implement a modular event detection system that processes a collection of historical twitter data retrospectively and performs the detection procedure as a pipeline of four discrete steps, a preprocessing phase, the topic modeling, the temporal partition and a postprocessing phase. In the topic modeling step, we test two different topic model algorithms, LDA and CTM, and for the two versions of the system derived, we conduct an experimental comparison of the system's performance. In the comparison, we take both the final results of the events detection algorithm and the intermediate results of topic modeling into consideration. On the one hand, CTM demonstrates better predictive capabilities as far as topic modeling is concerned. On the other hand, regarding the events detected, LDA performs better in the detection of smaller events compared to CTM, whereas CTM achieves a higher granularity, detecting accurate subevents of the larger events. Therefore, the choice of the topic model constitutes a tradeoff between different targets of the detection.

SUBJECT AREA: Data Mining

KEYWORDS: event, detection, LDA, CTM, clustering, temporal model

ΠΕΡΙΛΗΨΗ

Η πρόσφατη ανάδειξη των κοινωνικών δικτύων και ειδικότερα του Twitter έφερε ως αποτέλεσμα δεδομένα πρωτοφανούς μεγέθους για το περιεχόμενο και τις αλληλεπιδράσεις που παράγουν οι χρήστες στο πλαίσιο αυτό. Αυτή η πληροφορία μπορεί να αξιοποιηθεί για πλήθος εφαρμογών, μια από τις οποίες είναι και η ανίχνευση γεγονότων δηλαδή ο καθορισμός ενός χρονικού διαστήματος και ενός συνόλου εγγραφών περιεχομένου που σχετίζονται με ένα γεγονός στον πραγματικό κόσμο.

Στην εργασία αυτή, υλοποιούμε ένα αρθρωτό σύστημα ανίχνευσης γεγονότων που λειτουργεί εκ των υστέρων πάνω σε μια συλλογή από ιστορικά δεδομένα του Twitter και επιτελεί την διαδικασία σε διακριτά στάδια μιας σωλήνωσης με τέσσερα βήματα, την προεπεξεργασία, την μοντελοποίηση θεμάτων, την χρονική διαμέριση και την μεταεπεξεργασία. Στο στάδιο της μοντελοποίησης θεμάτων δοκιμάζουμε δυο διαφορετικούς αλγόριθμους μοντελοποίησης, το LDA και το CTM και για τις δυο εκδοχές του συστήματος που προκύπτουν διεξάγουμε μια πειραματική σύγκριση της απόδοσης του συστήματος. Στη σύγκριση λαμβάνονται υπόψη τόσο τα τελικά αποτελέσματα του αλγορίθμου ανίχνευσης γεγονότων όσο και τα ενδιάμεσα αποτελέσματα των θεμάτων που παράγει το μοντέλο. Από τη μια μεριά, το CTM φαίνεται να έχει καλύτερες δυνατότητες πρόβλεψης ως μοντέλο σε επίπεδο θεμάτων. Από την άλλη, σε επίπεδο γεγονότων το CTM φαίνεται να υστερεί στην ανίχνευση μικρότερων γεγονότων σε σχέση με το LDA αλλά κάνει ακριβέστερη ανάλυση σε υπογεγονότα των μεγαλύτερων γεγονότων. Συνεπώς, η επιλογή του μοντέλου αποτελεί ένα αντάλλαγμα μεταξύ διαφορετικών στόχων της ανίχνευσης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Εξόρυξη Δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: γεγονότα, ανίχνευση, LDA, CTM, συσταδοποίηση, χρονικό μοντέλο

To our families, for their support, understanding and love that they provided us with throughout these years.

ACKNOWLEDGMENTS

For the present Bachelor’s thesis, we would like to thank our supervisor, professor Dimitrios Gunopulos, for the guidance and the advice that contributed towards successfully completing the work.

CONTENTS

PREFACE 13

1. INTRODUCTION 14

1.1 Background 14

1.2 Related Work 14

1.3 Our contribution 15

2. SYSTEM OVERVIEW 16

2.1 Preprocessing 17

2.2 Topic Modeling 17

2.3 Temporal Partition 18

2.4 Postprocessing 19

3. EXPERIMENTAL RESULTS 21

3.1 Dataset Description 21

3.2 Experimental Comparison 22

3.3 Topic Clustering Evaluation 22

3.4 Event Evaluation 23

4. CONCLUSIONS 25

TABLE OF TERMINOLOGY 26

ABBREVIATIONS - ACRONYMS 27

REFERENCES 28

LIST OF FIGURES

Figure 1: An example of the temporal partition of a topic 19

Figure 2: Perplexity comparison of LDA and CTM for varying number of topics.....23

LIST OF IMAGES

Image 1: Overview of the event detection pipeline 16

LIST OF TABLES

Table 1: Description of ground truth events of dataset 21

Table 2: Top-5 words for X-factor, Paul Walker and Premier League latent topics 22

Table 3: Event detection metrics for the comparison of LDA and CTM based methods 24

Table 4: Detected sub-events of Chelsea vs Southampton match 24

PREFACE

The thesis at hand was undertaken by the authors 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 October 2016 to June 2017 in Athens under the supervision of professor Dimitrios Gunopulos.

On our behalf, we would like to thank our supervising professor for the guidance he provided over the course of the project. In addition, we would like to thank the members of the KDD lab for providing us with the data that we used for our experiments.

1. INTRODUCTION

1.1 Background

Twitter has become a very successful social media platform with millions of users all over the world. Its popularity is based on the compact and concise posts - known as tweets - which are broadcast by users as a means of social interaction and are enriched by tagging mechanisms such as hashtags and mentions. Daily, Twitter users both produce and interact with content that concerns their social environment as well as a global scope. The information wealth available - combined with the social correlation between stakeholders - can contribute to detecting important social and real-life events that cause the Twitter activity by using text and twitter-specific feature mining approaches. This prospect poses the problem of event detection, that is, the identification of significant incidents (e.g. Concerts, News, TV-Shows, Landmarks, etc.) that draw the attention of the public.

One striking characteristic of the problem is the lack of consensus on a single definition of events and event detection. Some authors propose definitions that highlight the consequences of the event [1] and by extension their ability to motivate actions, such as an increased rate of producing content in the social media, while others emphasize the occurrence of posts with high topical similarity and temporal proximity that are motivated by the same event. Considering that the Twitter data available consist of tweets, it is reasonable to associate the detection of an event with the identification of a series of tweets that belong to the same context and a time window, which is closer to the second definition. The time window along with the set of tweets are used to define each of the detected events.

Several approaches have been employed towards the addressing the problem, however the solutions generally fall in two categories; the detection can either be retrospective or real-time. In the former, historical events are extracted from a pool of data that has already been collected with a batch processing technique, whereas in the latter, new tweets are processed as they arrive, one-at-a-time. The two approaches share the same goal of event detection; however, different objectives and limitations apply to each (i.e. retrospective techniques are allowed to use more expensive algorithms but is expected to be more accurate, real-time has to process data at a fast rate and yield a high throughput even if that results in accuracy trade-offs).

Event detection in Twitter comes with challenges inherent to the content of the social media platform. Each microblog entry has length constraints and contains limited information. Additionally, a significant proportion of the data is noisy in several ways, ranging from being spam content to having no importance outside a social context and containing typos. The lack of moderation of the content implies that the systems should be noise-tolerant. Another issue inherent to the content of Twitter is that unlike news articles and feeds, not all microblog entries are associated to an actual event.

1.2 Related Work

Topic discovery in social networks has attracted much more attention than in the past, due to the high volume of data provided. Event detection is promising in a wide range of applications such as identifying trends in public data [2], targeting advertisements, influential profiles and of course event detection based on tweets.

The methods used, differ according the application requirements and dataset characteristics. Relatively recent systems- based on text-mining and time intensity - focus on detecting worldwide events (e.g."Breaking News") utilizing not only Tweets from a

large number of users, but also from external resources (e.g BirdDog-API) [3,4].Other systems though follow a different approach aiming at detecting localized events in real time (e.g Jasmine System) [5]. Recent work has also utilized LDA [6] as the main approach in more sophisticated text topic models, such as author-document models, abstract-reference models, syntax-semantics models and image-caption models. The same kind of modeling tools have also been used in a variety of non-text settings such as face-recognition, image-classification. On the contrary, Correlation Topic Model (CTM), which explicitly models the correlation between the latent topics in the data collection has not been utilized widely in the field. Promising results have been extracted though on topic detection of scientific journals dataset [7]. Replacing the Dirichlet distribution of LDA with the Logistic Normal distribution - that (CTM) utilizes - claimed to result in a better fit on the dataset and enhanced the inference ability of the system. Our research aims to apply the Correlation Topic Model in a noisy dataset as is the case with Twitter in order to isolate subtopics with different contexts and achieve higher granularity.

Prior work in event detection adopts a pipelined approach. In [8] a design consisting of multiple consecutive steps is used for event detection for social streams such as blogs and emails. The proposed approach models content as nodes in a graph that is partitioned according to textual, temporal and social features.

1.3 Our contribution

Our method involves a modular pipelined system, with a clean and extensible architecture, that performs retrospective event detection over a base of tweets. The pipeline consists of a preprocessing step, a two-phase algorithm for candidate event extraction -topic modeling algorithm by temporal analysis- and a post-processing step that optimizes the results. We utilize different topic model algorithms, Latent Dirichlet Allocation [6] and Correlated Topic Model [7], and conduct a comparative experimental analysis that concerns efficiency, robustness, quality and inference ability. Both Latent Dirichlet Allocation (aka. LDA) and Correlation Topic Model (aka. CTM) are generative probabilistic models of a corpus, in which each document is represented as a mixture of latent or correlated-latent topics respectively. The main difference between these two approaches is, that even though both follow similar outline on training and inference of data, they utilize different probabilistic distributions over documents. Each document is represented by words and each word is assigned to a topic. The distribution used to represent a document as a mixture of topics differs from LDA to CTM. The former takes advantage of the Dirichlet distribution, whereas the latter claims that using the Logistic Normal instead of Dirichlet allows it to detect latent topics between documents having any type of correlation between them more efficiently.

Our intuition is that, due to the fact that events manifest through the interaction of different entities, each belonging to a topic, in the context of events and the content related with them distinct topics co-exist and are consequently correlated. Hence, we expect that correlated topic models capture the latent topic information more accurately and result in a finer-grained detection. The analysis shows that while sub-events are detected more independently and false positives are reduced, these gains are at the expense of less reported events in the dataset.

2. SYSTEM OVERVIEW

The event detection system we implemented consists of several modules that are utilized sequentially. The processing pipeline consumes a pool of tweets on the one end and produces a series of event on the other. At first, the dataset goes through a preprocessing phase that cleans the data in order to filter irrelevant information, extract useful features and improve the overall results of the subsequent steps. Next, we apply a topic model to the tweets and form clusters that correspond to the topics discovered and partition the data. After that, we retrieve a time series of the temporal intensity for each cluster and detect the intervals between the local minimums which consequently contain an intensity peak each. We assume that the peaks are caused by events and thus for each interval we aggregate tweets that have similar content and are likely to report the same information. Finally, a postprocessing phase removes the candidate clusters that are not likely to be actual novel events.

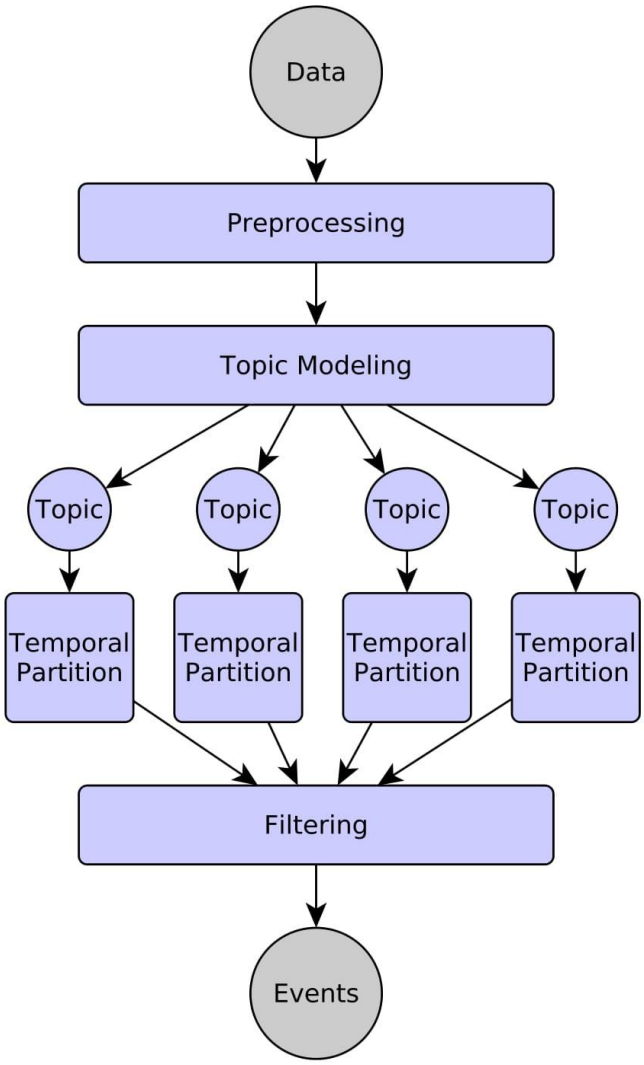

Image 1: Overview of the event detection pipeline

2.1 Preprocessing

During the data cleaning step, we process the textual elements of tweets to enable easier and higher quality processing in the subsequent steps. Initially, Twitter tags that have been added by the author such as hashtags and mentions are located. Sequences of continuous words, known as n-grams, that match with frequently used hashtags are identified and if the hashtag is not present in the original text, it is appended to it so as to enrich the information present. Index structures are maintained so that tweets that share a tag can be accessed efficiently. Next, the text is converted to a bag-of-words representation, which is convenient for different algorithms. To achieve this conversion, we remove punctuation, special characters and whitespace, thus transforming the text to an unordered set of words. Some words that are useless or even detrimental to further processing (“the”, “he”, “today”, etc.) called stopwords are removed from the bag-of-words representation of the tweet at this point. Moreover, data defining “original post-reply” relationships between tweets are exploited to construct a graph with tweets as vertices and these links as edges. Each connected component, which has a tree structure with the original post as root, constitutes a discussion and the individual texts are merged together to form an aggregated text for the discussion that is treated as an integral entity. The assumption is that tweets in a discussion are semantically related and the aggregated text is more informative than its components in total.

2.2 Topic Modeling

A topic modeling approach processes the results of the previous phase so as to estimate the extent to which each topic is related to a tweet. The relevance of a topic is expressed as a proportion of topics for each text and therefore we can assign a tweet to the topic with the highest value.

As a means to improve the results, we use a pooling technique for the training data of the models. In [9], the authors use pooling in order to improve the quality of the topics models produced by LDA. Tweets that share a feature are aggregated together to become a single document and treated as one entity by the topic modeling algorithm. Specifically, we adopt a hashtag and mention pooling scheme. In the event that a tweet has multiple hashtags or mentions, it is appended to all the relevant aggregate documents. The pooling of tweets that share the same hashtag has a global scope as hashtags are highly correlated with the concept of topics. On the contrary, due to the fact that a social entity can interact with different topics at different times, we constrain the pooling policy to only merge tweets with temporal proximity. Intensity peaks of the use of each mention are identified and individual mentions are associated to a peak. This method constrains the topic assignment with Twitter's native tags without modifying the topic model itself.

We have tried two different topic models for this step of the pipeline of the system, LDA and CTM.

LDA topic model as described in [6] is a generative probabilistic model of a corpus, in our case a text one, where the documents are represented as a mixture of latent components. This mixture is defined according the Dirichlet distribution over a specific document. For each topic, a word is generated according a multinomial distribution over the topic and becomes part of the LDA-generated document. Each real document is not necessarily generated in its exact form by LDA, due to the bag-of-words orientation of the topic model.

Even though LDA gives pretty accurate fit of topic assignments on a wide range of data, the perception that method overlooks the correlation between the subsequent assigned latent topics motivates our comparative experimental analysis on Twitter's data. The

model's inability is byproduct of using the Dirichlet distribution for randomly defining the topic proportions over the documents. For the reason mentioned above, we step into CTM topic model, which is quite similar to LDA's philosophy. CTM is a hierarchical model of document collections, which takes into account the latent components' correlation, by using the Logistic Normal Distribution instead of Dirichlet. Documents -in our case tweets- are produced from a generative process similar to that of LDA. The main difference is that in LDA topic proportions are drawn using the Dirichlet distribution in contrast with the utilization of Logistic Normal distribution by CTM.

This change trades off not only the use of the LDA's straightforward posterior inference algorithm- due to its latter's compatibility with multinomial distribution- but also sampling techniques such as Gibbs and MCMC sampling. However, we make use of the variational inference procedure that is described in [7].

2.3 Temporal Partition

Each topic cluster corresponds to a temporal model that describes the intensity I(t) of twitter activity that is related to the topic. An intensity curve that is smooth and continuous is desirable because it captures the temporal trend more accurately and small fluctuations that correspond to noisy behavior or insignificant events are ignored. This assumption stems from the fact that important events that are most worth reporting have a lasting impact and the information flow is not instant, which corroborates that they do not correspond to noisy behavior. Therefore, instead of a histogram of the tweet timestamps, we compute the intensity as the density of timestamps at a given moment as follows:

$$I(t) = \sum_{t \in T} e^{-\frac{(t-t_i)^2}{2\sigma^2}} \quad (1)$$

where T is the multiset of timestamps in the topic and σ is a parameter that determined the width in which each timestamp affects the density. The value of the parameter is usually small so that the effect is local and we only compute the contribution of the Gaussian function within 4 standard variations for efficiency. This allows for an efficient computation of the intensity because the resulting time complexity is O(|T|) rather than O(w|T|) where w is the difference between the maximum and the minimum timestamps.

The temporal model can be further processed to remove minor fluctuations. We apply a low-pass filter to the intensity function and blunt some of the brief variations. This procedure is implemented as a computation of the Fourier transform of I(t) followed by the multiplication with the filter, which removes high frequencies of the specter, and the computation of the inverse Fourier transform.

The procedure that follows is focused on the upward and downward trends of the filtered intensity rather than its actual value and for this reason we compute an approximation of the curve with a sequence of linear segments to allow more efficient computations. The linear approximation of the curve is computed with the Ramer-Douglas-Peucker algorithm [10], which refines the linear approximation recursively. Starting from a linear segment that connects the two endpoints of the curve, the point furthest away from the linear approximation is identified and the segment is replaced by a polygonal chain of two segments passing through it. The algorithm is then invoked recursively for the two new

segments. After the linearization has been computed, the minimums of the curve are located by traversing the segments and the intervals between them are are identified. Each of the intervals contains a peak of the intensity which is assumed to be caused by an event. The tweets that correspond to each interval are regarded separately. figure 1 demonstrates an example of a temporal partition for a given intensity curve.

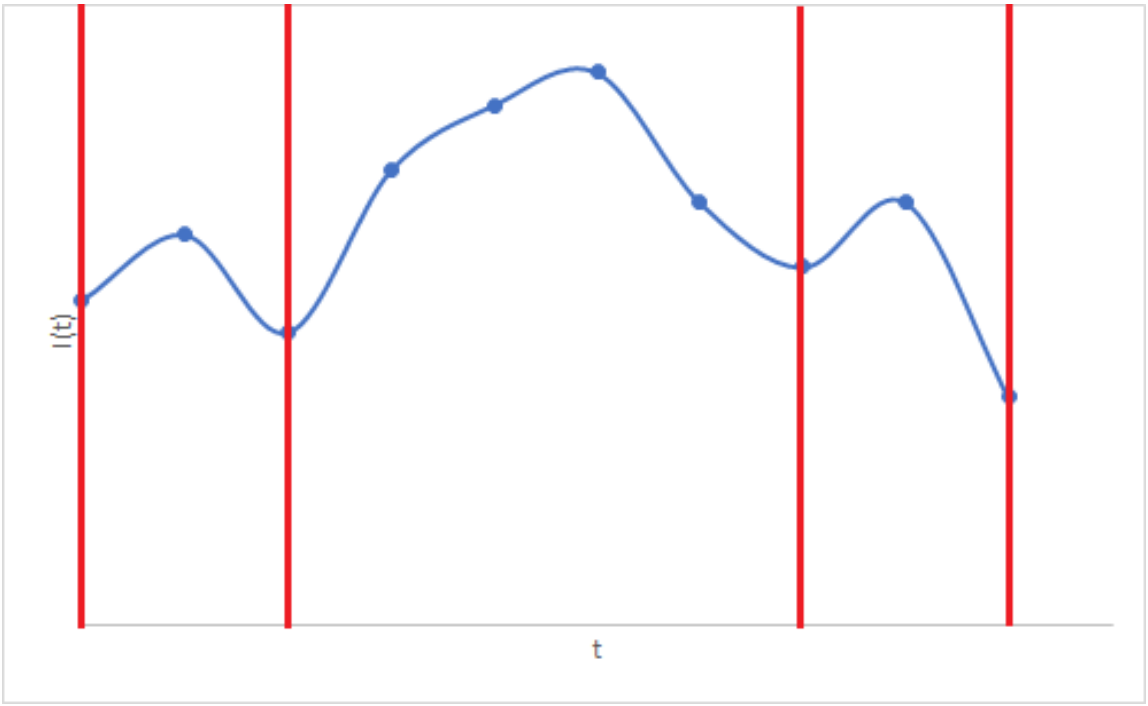

Figure 1: An example of the temporal partition of a topic

We extract candidate events from the time intervals that define the temporal partition. These events are formed by clustering the tweets based on their content. In order to account for events that have common phrases, which is a stronger indication of high similarity compared to unordered common words, the text is enriched with frequent bigrams. A count vectorizer is then used to convert the text to vectors of integers and the data is subjected to a hierarchical clustering algorithm. In each step, the clusters with the highest mean similarity merge and form a larger cluster until the similarity metric is lower than a threshold for each pair of the remaining clusters. The clusters correspond roughly to a homogeneous set of information and could represent an event.

2.4 Postprocessing

We submit the resulting clusters to a postprocessing phase, as a final step to our pipeline. We only keep clusters for which we are confident that they represent events. Clusters that consist of too few tweets or whose timestamps are too sparse are discarded. Moreover, we construct a spam filter that archives clusters for which the tweets per author count is too high or have multiple contexts as it can be inferred from the number of mentions in the tweets. We score the rest of the clusters according to their similarity with the filter's cluster pool and remove those that are closely related. The goals of the filter are to boost the precision of the detection without sacrificing recall. Ultimately, we report the events that consist of the tweets that belong to the clusters produced by the postprocessing phase.

It should be noted that in many cases merger and fragmentation of events may occur. Tweets that refer to the same event can be agglomerated to a number of different clusters that result in different detected events due to differences in vocabulary, context, time of post or constituting a notable sub-event e.g. most FIFA world cup groups are discussed separately during the draw ceremony. The event is reported through fragments that have a different context. The opposite can happen as well when there is some semantic correlation between two events happening simultaneously i.e. different football matches that overlap. This issue is related partially to granularity goals of the system, as fine-grained detection is more prone to fragmentation while coarse-grained one might suffer from mergers.

3. EXPERIMENTAL RESULTS

In this section, we elaborate on the experimental evaluation of the performance of the system described. First, we discuss the test data used for the experiments in subsection 3.1. Next, we assess both the quality of the clusters produced by the topic modeling component and the events detected at the end of the pipeline and compare the respective results for LDA and CTM-based pipelines.

3.1 Dataset Description

We test our method with a corpus of tweets collected from the geographic area around London which entails that the text is mostly in English. The dataset consists of approximately 560k tweets dating from November 29, 2012 to December 9, 2012. Each data entry consists of the unique id of the tweet, the textual content, the author's name and id, a timestamp, location information and, on the occasion that it is a reply to another tweet, the tweet id and the author's name of the original tweet. In total, the size of the dataset is 106.5 Mb. Ground truth information is available for the given time period which allows for estimating the recall performance of the event detection technique and is displayed in table 1. The ground truth events consist of a mix of breaking news and scheduled events.

Table 1: Description of ground truth events of dataset

Event type	Date(s)
Glasgow helicopter crash	29/11
Australia vs NZ rugby	30/11
Paul Walker dies	1/12
Nelson Mandela dies	5/12
FIFA world cup draw	6/12
Premier League matches	30/11,01/12,03/12,04/12,07/12,08/12
X Factor live shows	30/11,01/12,07/12,08/12
I'm a Celebrity episodes	daily

The most notable events that are present in the corpus are the deaths of Paul Walker and Nelson Mandela and the ensuing reactions. Other events concern recurrent themes such as the "Premier League", the "I'm a celebrity" reality show and the "X Factor" contest, which pose the challenge of discerning sub-events that have a similar context and vocabulary. Minor events, many of which not included in the ground truth, also occur in the dataset regularly.

3.2 Experimental Comparison

We process the dataset with the aforementioned event detection pipeline described. We examine the intermediate clustering results produced after the topic modeling component of the system as well the events detected at the end of the postprocessing phase. The two variations, using LDA and CTM respectively, are compared in terms of these results.

3.3 Topic Clustering Evaluation

We fit the topic modeling algorithms with the corpus of tweets pooled by common hashtags and mentions. We acquire models that associate each occurring word to different topics. Then, we apply a variational inference algorithm for each original individual tweet to estimate the contribution of each topic and assign each tweet to the topic that is most associated with. We use Blei's implementation of the algorithm for estimating the model.

Table 2: Top-5 words for X-factor, Paul Walker and Premier League latent topics

	Xfactor	PaulWalker	Premier
CTM	Xfactor Roughcopy Rough Copy Factor	Paul Rippaulwalker Walker Paulwalker Gutted	Arsenal Everton Spurs Comeon Moyes
LDA	Xfactor Tamera Getin Factor Luke	Mtvstars Rippaulwalker Paul Walker Paulwalker	Moyes Everton Mufc Hammersmith Manutd

In order to assess the quality of the topic clustering, we use various metrics. To compare the two distributions we use perplexity, which is a quantitative metric and measures how well each model can predict remaining words of documents, after a partition of them in observed and holdout segments. The predictive power of each model is represented in Figure 2.

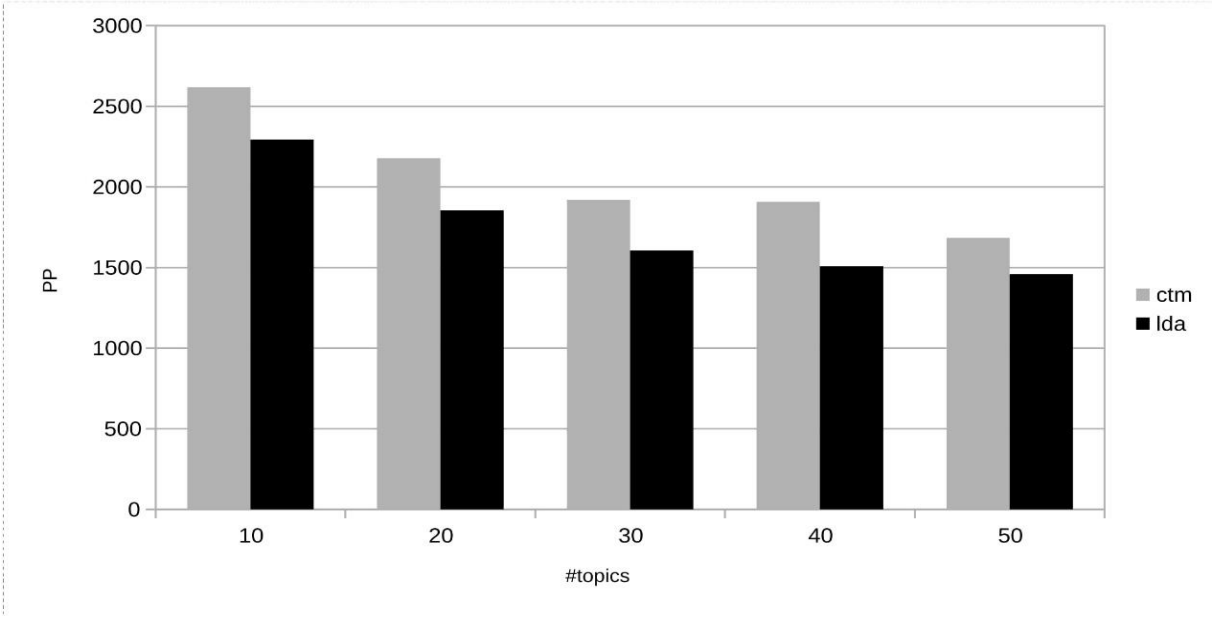

Figure 2: Perplexity comparison of LDA and CTM for varying number of topics

3.4 Event Evaluation

Detected events consist of a set of similar tweets posted during a specific time period. In order to evaluate the performance of our method, we compute the Precision, Recall and F1 score metrics. On the one hand, not all clusters at the output of the system represent actual events, therefore we measure Precision as the percentage of results that could be verified as real-world events. We approximate precision by manual inspection of a random sample of 100 result clusters. On the other hand, detected actual events are a subset of the events that occur throughout the period of data collection and thus we compute Recall by measuring the percentage of the events in the ground truth dataset that are present in the output. For event with episodic structure, each individual part is considered separately, although football matches happening at the same time are grouped together because they are often commented on as such in the text by spectators. Events with a significant duration such as an X-factor live show may be reported through a series of distinct sub-events, such as individual performances, rather than integrally. We consider that they are present in the output if at least one of their sub-events can be found in the resulting clusters. Additionally, we combine the two metrics to compute their harmonic mean called F1 score. The results of the measurements are accumulated in table 3 for pipelines that use an LDA or a CTM module in the topic modeling component of the pipeline.

Table 3: Event detection metrics for the comparison of LDA and CTM based methods

Topic Model	Precision	Recall	F1 score
LDA	0.620	0.909	0.737
CTM	0.660	0.848	0.742

We run the event detection experiments for 50 latent topics. The two variants have comparable results overall. The LDA approach exhibits a higher recall capability whereas the CTM approach has higher precision. The F1 score metric is roughly the same for both approaches. A closer examination of the results provides a better insight for understanding the metrics. Compared to the results produced with LDA topic modeling, in the case of CTM some smaller events are absent. By contrast, events that are more widely reported are broken down more clearly to their respective sub-events providing for better granularity. An example of the aforementioned observation is given in table 4. The table lists the sub-events of the football match Chelsea vs Southampton held on 1/12/2013 that are reported as independent clusters rather than parts of clusters referring to the overall game. This constitutes a trade-off as regards the use of topic models, sacrificing smaller events for more accurate sub-events.

Table 4: Detected sub-events of Chelsea vs Southampton match

Sub-event	LDA	CTM
0-1 (Rodriguez goal)	Yes	Yes
1-1 (Cahill goal)	No	Yes
2-1 (Terry goal)	Yes	Yes
3-1 (Ba goal)	No	Yes

4. CONCLUSIONS

In this project, we designed a modular system for retrospective event detection from a pool of Twitter data. The data flows through the successive components to compose events at the end of the pipeline. We study the behavior of the system for different topic modeling modules, namely one that relies on Latent Dirichlet Allocation and another based on Correlated Models. We see that the model used influences the final result. The latter yields topics that have better precision in sub-events compared to the former. This means that CTM seems to be able of recognizing more minor events, which are included in the major ones (table 4). Concerning the final event detection results, replacing LDA with CTM sacrifices recall capability for higher precision and finer granularity. Therefore, different topic models satisfy different needs in capturing events. In this direction, we should take into consideration, that both algorithms include a large number of parameters. These parameters are being highly involved in calculations and consequently the events are highly correlated to them. Thus, tweaking these parameters (ex. seed in EM, number of iterations, variances...etc.) may correspond to a major impact on the quality of results. Besides, the tweets follow a special text format, with characteristics -as described above- that either complicate the detection, such as length constraint, slang language or sometimes facilitate it, by providing useful information from the rich text format, such us hash-tags, comments and re-tweets.

Future directions include testing with a range of different components including other topic modeling techniques in the modular system described to further explore their impact in the results. Additionally, the event detection system could potentially benefit by integrating more features (i.e. location, content of links or pictures, relations between users, etc). Finally, the modular content-time partitioning approach could be modified for real-time detection taking throughput constraints into account.

TABLE OF TERMINOLOGY

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Cluster	Συστάδα
Partition	Διαμέριση
Pipeline	Σωλήνωση
Precision	Ακρίβεια
Preprocessing	Προεπεξεργασία
Postprocessing	Μεταεπεξεργασία
Real-time	Πραγματικού χρόνου
Recall	Ανάκτηση
Retrospective	Εκ των υστέρων

ABBREVIATIONS - ACRONYMS

LDA	Latent Dirichlet Allocation
CTM	Correlated Topic Model

REFERENCES

[1] J. Allan, *Introduction to Topic Detection and Tracking*, Springer US,2002, pp. 1-15.

[2] N. Panagiotou, I. Katakis, D. Gunopulos, *Detecting Events in Online Social Networks: Definitions, Trends and Challenges*, Springer International, 2016, pp. 42-84.

[3] A. Boettcher, D. Lee, "EventRadar: A Real-Time Local Event Detection Scheme Using Twitter Stream", Proc. 2012 IEEE Int'l Conf. Green Computing and Communications, 2012; doi: 10.1109/ GreenCom.2012.59.

[4] J. Sankaranarayanan, "TwitterStand: News in Tweets", Proc. 17th AC, SIGSPATIAL Int'l Conf. Advances Geographic Information Systems (GIS 09), 2009, pp. 42-51; doi: 10.1145/1653771.1653781.

[5] K. Watanabe et al., "Jasmine: A Real-time Local-event Detection System Based on Geolocation Information Propagated to Microblogs", Proc. 20th ACM Int'l Conf. Information and Knowledge Management (CIKM 11), 2011, pp. 2541-2544; doi: 10.1145/2063576.2064014.

[6] D.M. Blei, A.Y. Ng, M.I. Jordan, "Latent Dirichlet Allocation", *J. Machine Learning Research*, vol. 3, 2003, pp. 993-1022.

[7] D.M. Blei, A.Y. Ng, J.D. Lafferty, "A Correlated Topic Model of Science", *Annals of Applied Statistics*, vol. 1, no. 1, 2007, pp. 17-35.

[8] Q. Zhao, P. Mitra, B. Chen, "Temporal and Information Flow Based Event Detection from Social Text Streams", Proc. 22nd Nat'l Conf. Artificial Intelligence – Volume 2 (AAAI 07), 2007, pp. 1501-1506.

[9] R. Mehrotra et al., "Improving LDA Topic Models for Microblogs via Tweet Pooling and Automatic Labeling", Proc. 36th Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR 13), 2013, pp. 889-892; doi: 10.1145/2484028.2484166.

[10] D.H. Douglas, T.K. Peucker, *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature*, John Wiley & Sons, 2011, pp. 15-28.