
# National and Kapodistrian University of Athens

## School of Sciences

## Department of Informatics and Telecommunications


**PhD Thesis**

# Temporal Search in Document Streams


**Dimitrios Kotsakos**


**Athens**

**October 2015**


European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS, CULTURE & SPORTS
MANAGING AUTHORITY

NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND

Co-financed by Greece and the European Union


# Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

## Σχολή Θετικών Επιστημών

## Τμήμα Πληροφορικής και Τηλεπικοινωνιών


**Διδακτορική Διατριβή**

# Τεχνικές Εκμετάλλευσης Χρονικής Πληροφορίας από Δεδομένα Κειμένου


**Δημήτριος Κωτσάκος**



**Αθήνα**

**Οκτώβριος 2015**




# PhD Thesis

## Temporal Search in Document Streams

**Dimitrios Kotsakos**

## Advisor

**Dimitrios Gunopulos**, Professor, NKUA

## Main Advisory Committee

**Dimitrios Gunopulos**, Professor, NKUA
**Manolis Koumparakis**, Professor, NKUA
**Michael Hatzopoulos**, Professor, NKUA

## Examination Committee

**Dimitrios Gunopulos**
Professor
NKUA

**Manolis Koumparakis**
Professor
NKUA

**Michael Hatzopoulos**
Professor
NKUA

**Alexis Delis**
Καθηγητής
ΕΚΠΑ

**Ioannis Emiris**
Professor
NKUA

**Vasiliki Kalogeraki**
Assistant Professor
Athens University of Economics and Business

**Kostas Tsioutsiouliklis**
M5 Director, Research, Principal Research Scientist
Yahoo! Labs, Sunnyvale, CA, USA

**Examination Date:** 29/10/2015

# Διδακτορική Διατριβή

## Τεχνικές Εκμετάλλευσης Χρονικής Πληροφορίας από Δεδομένα Κειμένου

**Δημήτριος Κωτσάκος**

**Επιβλέπων**

**Δημήτριος Γουνόπουλος**, Καθηγητής, ΕΚΠΑ

**Τριμελής Επιτροπή Παρακολούθησης**

**Δημήτριος Γουνόπουλος**, Καθηγητής, ΕΚΠΑ
**Μανόλης Κουμπαράκης**, Καθηγητής, ΕΚΠΑ
**Μιχαήλ Χατζόπουλος**, Καθηγητής, ΕΚΠΑ

**Επταμελής Εξεταστική Επιτροπή**

**Δημήτριος Γουνόπουλος**
Καθηγητής
ΕΚΠΑ

**Μανόλης Κουμπαράκης**
Καθηγητής
ΕΚΠΑ

**Μιχαήλ Χατζόπουλος**
Καθηγητής
ΕΚΠΑ

**Αλέξης Δελής**
Καθηγητής
ΕΚΠΑ

**Ιωάννης Εμίρης**
Καθηγητής
ΕΚΠΑ

**Βασιλική Καλογεράκη**
Επίκουρη Καθηγήτρια
Οικονομικό Πανεπιστήμιο Αθηνών

**Κώστας Τσιουτσιουλικλής**
Διευθυντής Έρευνας
Ερευνητικό Ίδρυμα Yahoo! Labs, Καλιφόρνια, ΗΠΑ

**Examination Date:** 29/10/2015

## Abstract

In this thesis, we address major challenges in searching temporal document collections. In such collections, documents are created and/or edited over time. Examples of temporal document collections are web archives, news archives, blogs, personal emails and enterprise documents. Unfortunately, traditional IR approaches based on term-matching only can give unsatisfactory results when searching temporal document collections. The reason for this is twofold: the contents of documents are strongly time-dependent, i.e., documents are about events happened at particular time periods, and a query representing an information need can be time-dependent as well, i.e., a temporal query. On the other hand, time-only-based methods fall short when it comes to reasoning about events in social media. During the last few years users create chronologically ordered documents about topics that draw their attention in an ever increasing pace. However, with the vast adoption of social media, new types of marketing campaigns have been developed in order to promote content, i.e. brands, products, celebrities, etc.

The contributions in this thesis focus on a main IR topic: content analysis. In particular, we aim at improving the retrieval effectiveness by analyzing the contents of temporal document collections and disambiguating between different types of popular content in social media.

In this thesis, we analyze the contents of documents in order to determine the time of non-timestamped documents combining burstiness information with textual similarity. In contrast to the previous approaches which can only report time based on a pre-defined granularity (that reflects the segmentation of the reference corpus), this approach proposed in this dissertation can also report non-fixed intervals of application-defined length $l$. The approach is based on the intuition that similar documents are more likely to discuss similar events and hence being created closer in time, and that the burst intervals of significant terms (for example selected based on *tf-idf*) in those documents having high degree of overlap. The document dating process is performed by first finding the documents most similar to the docu-

ment to be dated. Second, a weight is assigned to each of the related documents based on the overlap of burst intervals of common terms between the relevant document and the document to be dated. Finally, each publication date along the timeline is assigned the sum of the weights of documents published at that time, and the result interval is chosen as the time interval of length $l$ having maximum sum of weights. Based on the experimental evaluation, this is the current state-of-the art approach to learning-based document dating.

Memes

Through extensive evaluation, we show that our proposed time-aware approaches outperform traditional retrieval methods and improve the retrieval effectiveness in searching temporal document collections.

Subject area: Web Mining, Text Mining

Keywords: time series, classification, time evolution, social networks, trends, web mining

## Περίληψη

Καθώς ο αριθμός και το μέγεθος των μεγάλων χρονοθετημένων συλλογών εγγράφων (π.χ. ακολουθίες ψηφιοποιημένων εφημερίδων, περιοδικά, blogs) αυξάνεται, το πρόβλημα της αποτελεσματικής και αναζήτησης αυτών των δεδομένων γίνεται πιο σημαντικό. Στην παρούσα διδακτορική διατριβή μελετάται το πρόβλημα της εκμετάλλευσης της εκρηκτικότητας λεκτικών όρων (burstiness) με σκοπό την αποτελεσματικότερη και πιο εύχρηστη αναζήτηση σε μεγάλα σύνολα δεδομένων κειμένου. Η εκρηκτικότητα όρων έχει ερευνηθεί εκτενώς στη βιβλιογραφία με την έννοια ενός μηχανισμού για την ανίχνευση γεγονότων που απασχόλησαν τις συλλογές αυτές την περίοδο συγγραφής τους. Περιγράφεται η σχετική με το πρόβλημα της εκρηκτικότητας όρων βιβλιογραφία, παρουσιάζεται αναλυτικά μια συγκεκριμένη προσέγγιση για τη μοντελοποίηση της εκρηκτικότητας ενός όρου χρησιμοποιώντας τη θεωρία διαφορών (discrepancy theory). Η μέθοδος ανακάλυψης εκρηκτικότητας που παρουσιάζεται επιτρέπει να οικοδομηθεί μια ελεύθερη παραμέτρων, γραμμικού χρόνου προσέγγιση για τον προσδιορισμό χρονικών διαστημάτων της μέγιστης εκρηκτικότητας για ένα συγκεκριμένο όρο.

Μελετήθηκε το πρόβλημα της χρονοσήμανσης εγγράφων άγνωστης χρονικής στιγμής δημιουργίας δεδομένου ενός συνόλου αναφοράς αποτελούμενου από χρονοσημασμένα έγγραφα. Χρησιμοποιήθηκε ο αλγόριθμος της εργασίας [3] έτσι ώστε να υπολογίζονται τα χρονικά διαστήματα εκρηκτικής συμπεριφοράς των όρων ενός συνόλου αρχειακών δεδομένων και συγκρίθηκε με τον αλγόριθμο που προτείνεται στην εργασία [6] και επιλύει παρόμοιο πρόβλημα. Αποδείχτηκε πως η τεχνική ανεύρεσης των μεγίστων κλικών σε ένα γράφο, έτσι ώστε να υπολογίζονται τα διαστήματα που οι περισσότεροι σημαντικοί όροι ενός εγγράφου εμφανίζουν εκρηκτική συμπεριφορά, αποδίδει καλύτερα αποτελέσματα από την πρόσφατη βιβλιογραφία και βελτιώνει τις τιμές precision και recall των αποτελεσμάτων εκτίμησης της χρονικής

στιγμής ενός εγγράφου. Έγινε ενδελεχής πειραματική μελέτη του προτεινόμενου αλγορίθμου εκτίμησης της χρονικής στιγμής της δημιουργίας ενός εγγράφου, δεδομένων μόνο των περιεχομένων του. Πιο συγκεκριμένα, ο αλγόριθμος βασίστηκε σε τεχνικές εξαγωγής πληροφορίας, όπως οι τεχνικές επιλογής όρων tf*idf, temporal entropy και topic modeling.

Η χρήση ετικετών για την κατηγοριοποίηση περιεχομένου στον Παγκόσμιο Ιστό (tagging) παρατηρείται ιδιαίτερα σε πλατφόρμες με δημοσιευμένο από τους χρήστες περιεχόμενο. Η κύρια εκμετάλλευσή τους από τα συστήματα σχετίζεται με υπηρεσίες αναζήτησης και εξαγωγής πληροφορίας. Στις περισσότερες πλατφόρμες κοινωνικής δικτύωσης οι χρήστες υποσημειώνουν τις αναρτήσεις τους χρησιμοποιώντας όρους και το σύμβολο της δίεσης ('#') ονομάζοντας τους συγκεκριμένους όρους hashtags. Εκτός των περιπτώσεων που τα hashtags σχετίζονται με πραγματικά γεγονότα (Events), παρατηρείται το φαινόμενο μεγάλες ομάδες χρηστών να χρησιμοποιούν τα hashtags για να προωθήσουν συζητήσεις, προϊόντα και ιδέες ή θέματα γνωστά ως Memes. Στην εργασία αυτή ορίζεται η διαφορά μεταξύ των Events και των Memes. Ένα κοινό χαρακτηριστικό και των δύο εννοιών είναι ότι ωθούν τους χρήστες κοινωνικών δικτύων και πλατφορμών δημοσίευσης περιεχομένου (είτε μικρής είτε μεγάλης έκτασης) - κείμενο, εικόνες, βίντεο κτλ - να δημιουργούν και να δημοσιεύουν περιεχόμενο σχετικό με συγκεκριμένα γεγονότα, πρόσωπα, συμβάντα, σημαντικά ή μη. Παρέχεται ένας τυπικός ορισμός του τι είναι ένα *Meme* και τι είναι ένα *Event* στα κοινωνικά δίκτυα και προτείνεται και αξιολογείται ένα σύνολο χαρακτηριστικών μη σχετικών με τη γλώσσα συγγραφής του περιοχομένου για την κατηγοριοποίηση των hashtags σε *Events* ή *Memes*. Αξιολογείται η προτεινόμενη προσέγγιση όσον αφορά την ακρίβεια της κατηγοριοποίησης χρησιμοποιώντας δύο μεγάλα πραγματικά σύνολα δεδομένων από την κοινωνική πλατφόρμα Twitter με μηνύματα γραμμένα τόσο στην αγγλική και όσο και στη γερμανική γλώσσα. Τέλος παρουσιάζεται η χρησιμότητα και η αναγκαιότητα του διαχωρισμού των *Memes* και των *Events* για την ανίχνευση

γεγονότων, εφαρμόζοντας τη μέθοδο αναζήτησης εκρηκτικών όρων που παρουσιάζεται στο πρώτο κεφάλαιο της εργασίας.

Θεματική Περιοχή: Εξόρυξη Δεδομένων

Λέξεις-Κλειδιά: χρονοσειρές, κατηγοριοποίηση, χρονική εξέλιξη, κοινωνικά δίκτυα, τάσεις

## Συνοπτική Παρουσίαση της Διδακτορικής Διατριβής

Καθώς ο αριθμός και το μέγεθος των μεγάλων χρονοθετημένων συλλογών εγγράφων (π.χ. ακολουθίες ψηφιοποιημένων εφημερίδων, περιοδικά, blogs) αυξάνεται, το πρόβλημα της αποτελεσματικής και αναζήτησης αυτών των δεδομένων γίνεται πιο σημαντικό. Στην παρούσα διδακτορική διατριβή μελετάται το πρόβλημα της εκμετάλλευσης της εκρηκτικότητας λεκτικών όρων (burstiness) με σκοπό την αποτελεσματικότερη και πιο εύχρηστη αναζήτηση σε μεγάλα σύνολα δεδομένων κειμένου. Η εκρηκτικότητα όρων έχει ερευνηθεί εκτενώς στη βιβλιογραφία με την έννοια ενός μηχανισμού για την ανίχνευση γεγονότων που απασχόλησαν τις συλλογές αυτές την περίοδο συγγραφής τους. Περιγράφεται η σχετική με το πρόβλημα της εκρηκτικότητας όρων βιβλιογραφία, παρουσιάζεται αναλυτικά μια συγκεκριμένη προσέγγιση για τη μοντελοποίηση της εκρηκτικότητας ενός όρου χρησιμοποιώντας τη θεωρία διαφορών (discrepancy theory). Η μέθοδος ανακάλυψης εκρηκτικότητας που παρουσιάζεται επιτρέπει να οικοδομηθεί μια ελεύθερη παραμέτρων, γραμμικού χρόνου προσέγγιση για τον προσδιορισμό χρονικών διαστημάτων της μέγιστης εκρηκτικότητας για ένα συγκεκριμένο όρο.

Η έννοια της εκρηκτικότητας έχει μελετηθεί σε αρκετές και διαφορετικές περιοχές της εξόρυξης πληροφορίας. Αρκετές εργασίες βασίζονται στη γνωστή πρωτότυπη εργασία του J. Kleinberg "On the bursty and hierarchical structure of streams", στην οποία προτείνεται ένας αλγόριθμος ανεύρεσης χρονικών διαστημάτων κατά τα οποία ένας όρος εμφανίζει εκρηκτική συμπεριφορά. Πιο συγκεκριμένα, ο J. Kleinberg εξάγει σημασιολογική δομή από ροές εγγράφων, αναδεικνύοντας τη σημασία της διάστασης του χρόνου μέσω της αξιοποίησης της πληροφορίας που παρέχει ο χρόνος δημιουργίας κάθε εγγράφου. Έτσι, μοντελοποιεί μια ροή κειμένου χρησιμοποιώντας ένα αυτόματο απείρων καταστάσεων το οποίο με τη σειρά του βασίζεται στη Θεωρία των Κρυφών

Μαρκοβιανών Μοντέλων (Hidden Markov Models - HMMs). Τα bursts - οι εκρήξεις, σε μια ατυχή μάλλον προσπάθεια μετάφρασης του όρου - σηματοδοτούνται ως μεταβάσεις μεταξύ καταστάσεων στο αυτόματο αυτό. Ο αλγόριθμος δε χρησιμοποιεί τις απλές συχνότητες εμφάνισης των λέξεων αλλά ένα πιθανοτικό αυτόματο του οποίου οι καταστάσεις αντιστοιχούν στις συχνότητες εμφάνισης ενός όρου. Πιο συγκεκριμένα, οι μεταβάσεις καταστάσεων αντιστοιχούν σε σημεία του χρόνου κατά τα οποία η συχνότητα εμφάνισης μιας λέξης αλλάζει σημαντικά. Ο συγγραφέας εξετάζει τον αλγόριθμο στο αρχείο των προσωπικών ηλεκτρονικών μηνυμάτων του. Μια άλλη εργασία ανεύρεσης bursts παρουσιάζεται από τους Fung et al. [3] Στην εργασία αυτή, οι εκρηκτικοί όροι συσταδοποιούνται και αναπαριστούν γεγονότα που απασχολούν τα έγγραφα προς ανάλυση. Οι συγγραφείς της εργασίας [4] κατηγοριοποιούν τους όρους σε τέσσερις κατηγορίες εκρηκτικότητας ανάλογα με την πορεία εκρηκτικότητάς τους. Στην εργασία [13] οι συγγραφείς χρησιμοποιούν μια δομή βασισμένη σε wavelets και τη χρησιμοποιούν στην παρακολούθηση ροών δεδομένων. Η εκρηκτικότητα των όρων χρησιμοποιείται και στο πλαίσιο άλλων εφαρμογών εκτός από την ανεύρεση σημαντικών γεγονότων, όπως η συσταδοποίηση ροών δεδομένων [5] ή στη μελέτη γράφων [8]. Οι He et al. [6] εφαρμόζουν το μοντέλο του Kleinberg για να συσταδοποιήσουν θέματα που απασχολούν τη συλλογή εγγράφων προς μελέτη. Οι Bansal και Koudas [1][2] παρουσίασαν το Blogscope, ένα σύστημα για την ανάλυση μεγάλου όγκου χρονικά ταξινομημένων καταχωρήσεων κειμένου και το εφαρμόζουν σε ένα μεγάλο σύνολο καταχωρήσεων ιστολογίων (blogposts). Προσπαθούν να εκμεταλλευτούν τρία ειδικά χαρακτηριστικά των ιστολογίων:

1. Η πληροφορία που περιέχεται στα ιστολόγια συνδέεται με μια το χρόνο δημιουργίας μια καταχώρησης.
2. Οι καταχωρήσεις στα ιστολόγια μπορούν εύκολα να αντιστοιχηθούν στη γεωγραφική τοποθεσία στην οποία βρίσκεται ο συγγραφέας.

3. Τέλος, κάποιες αναρτήσεις σε ιστολόγια ενδέχεται να προκαλέσουν νέες σχετικές αναρτήσεις από τον ίδιο ή άλλους συγγραφείς που με τη σειρά τους θα οδηγήσουν στην έναρξη μιας συζήτησης.

Οι ερευνητές τονίζουν πως, παρόλο που η εργασία τους αφορά τα ιστολόγια, το σύστημα μπορεί πολύ εύκολα να τροποποιηθεί για να χειριστεί κάθε είδους, ταξινομημένες στον άξονα του χρόνου, ροές κειμένου όπως ειδησεογραφικές ανακοινώσεις, λίστες ηλεκτρονικού ταχυδρομείου, διαδικτυακά forums και άλλα μέσα κοινωνικής δικτύωσης. Παρόλο που στις σχετικές εργασίες για το Blogscope δεν δίνονται αρκετές λεπτομέριες σχετικά με τις υϊοθετημένες μεθόδους, η συνολική τους προσέγγιση σχετίζεται με την παρούσα εργασία, υπό την έννοια ότι αντιστοιχούν εκρηκτικούς όρους σε συγκεκριμένες καταχωρήσεις ιστολογίων. Η εργασία [9], που παρουσιάζεται στην παρούσα διδακτορική διατριβή, είναι η πρώτη που περιλαμβάνει την πληροφορία της εκρηκτικότητας στην ευρετηριοποίηση και κατάταξη εγγράφων με άμεσο τρόπο, δημιουργώντας έτσι μια πλήρη πλατφόρμα αναζήτησης εγγράφων βασισμένη στην εκρηκτικότητα των όρων. Τα βασικά πλεονεκτήματα της μεθόδου ανεύρεσης εκρηκτικών όρων και των αντίστοιχων χρονικών περιόδων που παρουσιάζεται στην εργασία [9] και μελετάται στην παρούσα διατριβή είναι ότι εκτελείται σε γραμμικό χρόνο και είναι ελεύθερη παραμέτρων. Τα δύο αυτά στοιχεία την καθιστούν ιδανική για πολύ μεγάλες ακολουθίες εγγράφων, οι οποίες μπορούν να καλύπτουν μεγάλες χρονικές περιόδους. Η πλατφόρμα αναζήτησης που περιγράφεται στην εργασία μπορεί να λειτουργήσει με κάθε μέθοδο ανεύρεσης εκρηκτικών όρων, με την προϋπόθεση ότι η μέθοδος παράγει μη-επικαλυπτόμενα χρονικά διαστήματα εκρηκτικότητας και τα αντίστοιχα βάρη για κάθε όρο.

Στα πλαίσια της παρούσας εργασίας γίνεται προσπάθεια εκμετάλλευσης των εκρηκτικών όρων σε δεδομένα που προέρχονται από ιστολόγια, δηλαδή λέξεις οι οποίες παρουσιάζουν ραγδαία αύξηση στον αριθμό εμφανίσεών τους σε

μικρές περιόδους του χρόνου συγκριτικά με το συνολικό χρόνο παρατήρησης. Η ακρίβεια των όρων, που εμφανίζονται ως εκρηκτικοί, αξιολογείται με την αντιστοίχησή τους σε γεγονότα της πραγματικής ζωής που ενδεχομένως έλαβαν χώρα κατά την περίοδο του burst. Μια συγκεκριμένη ιστορία σχηματίζεται από μια ομάδα πολλών συσχετιζόμενων όρων. Καθώς η δημοτικότητα ενός συγκεκριμένου θέματος συρρικνώνεται, η ομάδα αυτή παύει να υφίσταται. Η εύρεση συσχετίσεων μεταξύ των όρων, για την αυτόματη ανίχνευση τέτοιων ομάδων είναι ο έτερος στόχος της εργασίας. Η προσέγγιση που ακολουθείται βασίζεται στην υπόθεση ότι συσχετιζόμενοι όροι εμφανίζουν παρόμοια δραστηριότητα όσον αφορά την εκρηκτικότητα και επομένως συγκρίνονται οι καμπύλες εκρηκτικότητας στην περίοδο μελέτης. Η αξιολόγηση συνίσταται και πάλι στην προσπάθεια εξόρυξης ενός θέματος από την πραγματική ζωή που θα μπορούσε να είχε σχηματίσει μια τέτοια ομάδα συσχετιζόμενων όρων.

Μελετήθηκε το πρόβλημα της χρονοσήμανσης εγγράφων άγνωστης χρονικής στιγμής δημιουργίας δεδομένου ενός συνόλου αναφοράς αποτελούμενου από χρονοσημασμένα έγγραφα. Χρησιμοποιήθηκε ο αλγόριθμος της εργασίας [3] έτσι ώστε να υπολογίζονται τα χρονικά διαστήματα εκρηκτικής συμπεριφοράς των όρων ενός συνόλου αρχειακών δεδομένων και συγκρίθηκε με τον αλγόριθμο που προτείνεται στην εργασία [6] και επιλύει παρόμοιο πρόβλημα. Αποδείχτηκε πως η τεχνική ανεύρεσης των μεγίστων κλικών σε ένα γράφο, έτσι ώστε να υπολογίζονται τα διαστήματα που οι περισσότεροι σημαντικοί όροι ενός εγγράφου εμφανίζουν εκρηκτική συμπεριφορά, αποδίδει καλύτερα αποτελέσματα από την πρόσφατη βιβλιογραφία και βελτιώνει τις τιμές precision και recall των αποτελεσμάτων εκτίμησης της χρονικής στιγμής ενός εγγράφου. Έγινε ενδελεχής πειραματική μελέτη του προτεινόμενου αλγορίθμου εκτίμησης της χρονικής στιγμής της δημιουργίας ενός εγγράφου, δεδομένων μόνο των περιεχομένων του. Πιο συγκεκριμένα, ο αλγόριθμος βασίστηκε σε τεχνικές εξαγωγής πληροφορίας, όπως οι τεχνικές επιλογής

όρων tf*idf, temporal entropy και topic modeling. Εξετάστηκε η ιδέα συνδυασμού των παραπάνω μεθόδων έτσι ώστε να προσεγγιστεί ο διπλασιασμός των εγγράφων που χρονοσημαίνονται έγκυρα, αφού οι δύο μέθοδοι επιτυγχάνουν να χρονοσημάνουν έγκυρα διαφορετικά έγγραφα του συνόλου εισόδου. Όλοι οι αλγόριθμοι εκτελέστηκαν πάνω στα δεδομένα που συλλέχθηκαν από το λογισμικό που αναπτύχθηκε το εξάμηνο 1/2012-6/2012 καθώς και σε δεδομένα των ψηφιοποιημένων εφημερίδων New York Times και San Francisco Call. Το λογισμικό που αναπτύχθηκε για τη συλλογή των δεδομένων που ανήκουν στον Παγκόσμιο Ιστό γράφτηκε στις γλώσσες Python, Java και C++. Τα δεδομένα ευρετηριοποιήθηκαν και αποθηκεύτηκαν με τη χρήση της βιβλιοθήκης Java Lucene και χρησιμοποιήθηκε η βάση δεδομένων MySQL. Η προεπεξεργασία όλων των δεδομένων έγινε σε λογισμικό που αναπτύχθηκε στις γλώσσες Python και Java. Τα δεδομένα αφορούν σε:

- Δεδομένα Εφημερίδων. Για την πειραματική μας αξιολόγηση χρησιμοποιούνται περισσότερα από 390000 άρθρα από την εφημερίδα San Francisco Call, μία ημερήσια εφημερίδα του Σαν Φρανσίσκο, με ημερομηνίες δημοσίευσης ανάλσα στα έτη 1900 και 1909.
- Δεδομένα Ιστολογίων. Συλλέχθηκαν δεδομένα από τα δημοφιλή ιστολόγια engadget.com, slashdot.org, Allthingsd.com, GigaOM.com, Mashable.com, Mashable.com/socialmedia, New York Times, Pcmag.com, ReadWriteWeb.com, TechCrunch.com και Fastcompany.com.
- Δεδομένα Ειδήσεων. Το συγκεκριμένο σύνολο δεδομένων προέρχεται από τη δημοφιλή ιστοσελίδα ειδήσεων topix.com. Περιλαμβάνει 65540 άρθρα για διάστημα 333 ημερών, από τον Σεπτέμβριο του 2008 μέχρι τον Ιούλιο του 2009.

Για την πειραματική αξιολόγηση και τον ορισμό των τιμών των παραμέτρων για τους αλγορίθμους ελήφθησαν υπόψιν οι κατανομές και η συμπεριφορά των δεδομένων στην πάροδο

του χρόνου. Οι κατανομές των δεδομένων αποτελούν σημαντική πληροφορία για την ανάλυση των αποτελεσμάτων, καθώς κάποια σύνολα δεδομένων παρουσιάζουν εκρήξεις στο επίπεδο των άρθρων που δημοσιεύονται κάποιες μέρες ενώ άλλα παρουσιάζουν ομοιογενή κατανομή. Αυτή η διαφορά είναι πολύ σημαντική, καθώς καθιστά επιτακτική την ανάγκη κανονικοποίησης των ακολουθιών συχνοτήτων εμφανίσεων των όρων, πριν την τροφοδότηση των αλγορίθμων ανακάλυψης εκρήξεων

Η χρήση ετικετών για την κατηγοριοποίηση περιεχομένου στον Παγκόσμιο Ιστό (tagging) παρατηρείται ιδιαίτερα σε πλατφόρμες με δημοσιευμένο από τους χρήστες περιεχόμενο. Η κύρια εκμετάλλευσή τους από τα συστήματα σχετίζεται με υπηρεσίες αναζήτησης και εξαγωγής πληροφορίας. Στην κοινωνική πλατφόρμα Twitter οι χρήστες υποσημειώνουν τις αναρτήσεις τους χρησιμοποιώντας όρους και το σύμβολο της δίεσης (`'#'`). Οι ετικέτες αυτές ονομάζονται hashtags. Τα hashtags στο Twitter εμπλουτίζουν το περιορισμένης έκτασης κείμενο με χρήσιμη μετα-πληροφορία, δεδομένου ότι οι χρήστες συμφωνούν άτυπα να χρησιμοποιούν συγκεκριμένα hashtags για συγκεκριμένα γεγονότα (events), π.χ. `#worldcup2014`. Εκτός των περιπτώσεων που τα hashtags σχετίζονται με πραγματικά γεγονότα, παρατηρείται το φαινόμενο μεγάλες ομάδες χρηστών να χρησιμοποιούν τα hashtags για να προωθήσουν συζητήσεις, προϊόντα και ιδέες ή θέματα γνωστά ως memes.

Με βάση τις παραπάνω στοιχειώδεις έννοιες, στην εργασία αυτή ορίζουμε τη διαφορά μεταξύ των Events και των Memes. Ένα κοινό χαρακτηριστικό και των δύο εννοιών είναι ότι ωθούν τους χρήστες κοινωνικών δικτύων και πλατφορμών δημοσίευσης περιεχομένου (είτε μικρής είτε μεγάλης έκτασης) - κείμενο, εικόνες, βίντεο κτλ - να δημιουργουν και να δημοσιεύουν περιεχόμενο σχετικό με συγκεκριμένα γεγονότα, πρόσωπα, συμβάντα, σημαντικά ή μη. Και οι δύο εκδηλώσεις και μιμίδια σε ένα κοινωνικό δίκτυο κατευθύνει τους χρήστες να δημιουργούν και να δημοσιεύουν περιεχόμενο στο κοινωνικό

ρεύμα. Ως εκ τούτου, περιεχόμενο σχετικό τόσο με Events όσο και με Memes μπορεί να παρατηρηθεί σε μια ροή εγγράφων $s$ ενός κοινωνικού δικτύου, εμφανίζοντας απρόσμενα υψηλές συχνότητες γύρω από συγκεκριμένες χρονικές στιγμές ή περιόδους. Η διαφορά μεταξύ ενός πραγματικού και σημαντικού συμβάντος (Event) και ενός Meme - δυστυχώς η ελληνική μετάφραση *μιμίδιο* χαρακτηρίζεται ως ατυχής, ως εκ τούτου στο εξής θα χρησιμοποιηθούν οι αγγλικοί όροι Meme και Event - είναι ότι ένα Event μπορεί να επισημανθεί και σε μια ροή ειδήσεων $n$ της ίδιας χρονικής περιόδου με τη ροή εγγράφων $s$, ενώ αντικείμενα (ή χρήστες) σχετικά/σχετικοί με ένα Meme παρατηρούνται μόνο εντός της ροής $s$ του κοινωνικού δικτύου.

Πιο συγκεκριμένα, ένα Event θα μπορούσε να προσδιοριστεί παρατηρώντας τα μηνύματα και τις συζητήσεις σε μια πλατφόρμα κοινωνικής δικτύωσηςσχετικά με τις βουλευτικές εκλογές στη Γερμανία, έναν αγώνα ποδοσφαίρου μεταξύ των ομάδων της Βαρκελώνης και της Μάντσεστερ Γιουνάιτεντ, ένα σεισμό, ή σχετικά με την τελετή των Όσκαρ. Από την άλλη πλευρά σχετικά με Memes θα μπορούσαν να είναι τα μηνύματα που σχετίζονται με μια συγκεκριμένη ομάδα οπαδών μιας διασημότητας/ειδώλου που μέσω μιας καμπάνιας στα κοινωνικά δίκτυα ζητούν από το είδωλό τους να δώσει μια συναυλία στην πόλη/χώρα τους, μια συζήτηση σχετικά με το γιατί οι άνθρωποι δεν μπορούν να κοιμηθούν εκείνη τη βραδιά, κτλ. Σε αμφότερες τις περιπτώσεις, τόσο τα Memes όσο και τα Events, όπως και άλλα έγγραφα στα κοινωνικά δίκτυα, πολύ συχνά υποσημειώνονται με hashtags. Για παράδειγμα, τα αντίστοιχα hashtags για τα Events που περιγράφονται πιο πάνω θα μπορούσε να είναι: `#GermanyElections`, `#BarcaVsMancester`, `#earthquake`, `#Oscars2014`, ενώ για τα Memes που αναφέρονται στην προηγούμενη παράγραφο, τα αντίστοιχα hashtags θα μπορούσαν να είναι τα εξής: `#WeWantJustinInIreland`, `#20ReasonsIAmCute»`, `#loveit`, `#insomnia`.

**Αναζήτηση γεγονότων (Events):** Οι πλατφόρμες κοινωνικής δικτύωσης μπορούν να επωφεληθούν από τη διάκριση μεταξύ

των διαφόρων τύπων τάσεων ή των δημοφιλών θεμάτων. Με ένα τέτοιο εργαλείο, οι πλατφόρμες θα μπορούν να καταλάβουν καλύτερα *γιατί* κάποιο περιοχόμενο είναι ή γίνεται δημοφιλές μια συγκεκριμένη χρονική στιγμή, με αποτέλεσμα να μπορούν να εκμεταλλευτούν ή να προβλέψουν την απότομη αύξηση της δημοτικότητας συγκεκριμένων θεμάτων, με σκοπό να είναι σε θέση να προσφέρουν καλύτερες υπηρεσίες στους χρήστες τους. Για παράδειγμα μπορούν να προσφέρουν διαφορετικού τύπου σελίδες αποτελεσμάτων στα εργαλεία αναζήτησης που παρέχουν ανάλογα με το αν οι όροι αναζήτησης σχετίζονται με Memes ή Events ή διαφορετικές επιλογές διαφήμισης - με το αντίστοιχο κοστολόγιο - στους συνεργάτες τους. Επιπλέον, οι περισσότερες μέθοδοι ανακάλυψης γεγονότων σε ροές κοινωνικών δικτύων βασίζονται σε μεθόδους ανακάλυψης και εκμετάλλευσης της έννοιας της εκρηκτικότητας (burstiness) που αναλύεται στην παρούσα εργασία και παρουσιάστηκε συνοπτικά παραπάνω. Η υπόθεση που ακολουθούν οι περισσότερες μέθοδοι ανακάλυψης γεγονότων σε ροές κοινωνικών δικτύων είναι ότι εκρηκτική συμπεριφορά ενός όρου ή ενός διγράμματος (bigram) υποδεικνύει πώς κάτι σημαντικό συνέβη σχετικά με το συγκεκριμένο όρο, και ως εκ τούτου οι χρήστες της πλατφόρμας επηρεάστηκαν και έγγραψαν για το συγκεκριμένο γεγονός στο διαδίκτυο, προκαλώντας έτσι συζητήσεις και ανταλλαγή απόψεων σχετικά με αυτό. Παρόλαυτα, η συγκεκριμένη υπόθεση δεν απεικονίζει ολόκληρη την πραγματικότητα και δεν ανταποκρίνεται πάντα στην αλήθεια, μιας και οι δυναμικές των κοινωνικών δικτύων, συχνά οδηγούν στη δημιουργία θεμάτων ενδιαφέροντος που σχετίζονται μόνο με χρήστες και θέματα εντός του δικτύου και όχι πέρα από αυτό. Παραδείγματα τέτοιων περιπτώσεων παρουσιάστηκαν παραπάνω. Ως εκ τούτου, οι μέθοδοι που ανακαλύπτουν γεγονότα και βασίζονται στην υπόθεση αυτή δεν λαμβάνουν υπόψιν τους ότι διαφορετικοί τύποι δημοφιλούς περιεχομένου ακολουθούν και διαφορετικά πρότυπα συμπεριφοράς, παρόλο που έχουν κάποιες ομοιότητες σχετικά με την απότομη αύξηση της δημοτικότητάς τους. Εκτός λοιπόν από τη χρονική συμπεριφορά, ένα δημοφιλές θέμα (που αναπαρίσταται ως ένας ή περισσότεροι όροι) μπορεί

να χαρακτηρίζεται από μια πλειάδα παραγόντων, όπως την κοινότητα που ενδιαφέρεται για το θέμα αυτό ή τον τύπο των μηνυμάτων που είναι σχετικές με το θέμα και τα χαρακτηριστικά τους, π.χ. ο αριθμός των συνδέσεων σε εξωτερικούς ιστοτόπους (urls), ο αριθμός των συνημμένων φωτογραφιών, η παρουσία των hashtags, κλπ. Στην παρούσα εργασία, διαχωρίζουμε μεταξύ των διαφόρων ειδών των τάσεων και των δημοφιλών θεμάτων και σχολιάζουμε τι χαρακτηρίζει τους διαφορετικούς αυτούς τύπους, εστιάζοντας τη μελέτη μας στα Memes και τα Events.

Πιο συγκεκριμένα, η συνεισφορά της παρούσας εργασίας στο ερευνητικό πρόβλημα που παρουσιάστηκε παραπάνω συνοψίζεται ως εξής:

- Παρέχεται ένας τυπικός ορισμός του τι είναι ένα *Meme* και τι είναι ένα *Event* στα κοινωνικά δίκτυα, αναγνωρίζοντας το γεγονός ότι δεν συμπεριφέρονται όλα τα δημοφιλή θέματα με τον ίδιο τρόπο.
- Προτείνεται και αξιολογείται ένα σύνολο χαρακτηριστικών μη σχετικών με τη γλώσσα συγγραφής του περιοχομένου για την κατηγοριοποίηση των hashtags σε *Events* ή *Memes*.
- Αξιολογείται η προτεινόμενη προσέγγιση όσον αφορά την ακρίβεια της κατηγοριοποίησης χρησιμοποιώντας δύο μεγάλα πραγματικά σύνολα δεδομένων από την κοινωνική πλατφόρμα Twitter μηνύματα γραμμένα τόσο στην αγγλική και όσο και στη γερμανική γλώσσα.
- Παρουσιάζεται η χρησιμότητα του διαχωρισμού *Memes* και *Events* για την ανίχνευση γεγονότων, εφαρμόζοντας τη μέθοδο αναζήτησης εκρηκτικών όρων που παρουσιάζεται στο πρώτο κεφάλαιο της εργασίας.
- Παρέχεται μια εκτενής μελέτη της συμπεριφοράς που χαρακτηρίζει *Memes* και *Events* και παρουσιάζεται μια ταξινόμηση των προτεινόμενων χαρακτηριστικών με βάση το Gain-Ratio για τους χρησιμοποιούμενους κατηγοριοποιητές στο περιβάλλον μελέτης.

# Contents

# List of Figures

# List of Tables

# Ευχαριστίες

# Chapter 1

# Introduction

This PhD thesis addresses different challenges in searching temporal document sequences, where documents are created and/or edited over time, and the contents of documents are strongly time-dependent. Examples of temporal document collections are web archives, news archives, blogs, social networking platforms, and personal emails. The main focus of this dissertation is how to exploit temporal information provided in documents and combine it with textual information with the goal of improving the effectiveness of searching temporal document collections.

This chapter describes the motivation and research questions addressed in the thesis. In addition, we explain our research context and methods. Our contributions to this thesis are composed of different approaches to solving the addressed research questions. In the end of this chapter, the organization of the rest of the thesis is presented.

## 1.1 Motivation

The ease of publishing content on social media sites brings to the Web an ever increasing amount of content captured during various types of events and/or before/after these events take place. Event content shared on social media sites such as blogs, Twitter, Facebook, YouTube, and others varies widely, ranging from planned, known occurrences such as a concert or a parade, to unplanned incidents

such as an earthquake, floods or death of a celebrity. By exploring and proposing techniques to automatically identify and characterize these events and the relevant user-contributed social media documents (e.g., blog posts, photographs, videos, messages, status updates), we can enable rich search and presentation of all event content. In this dissertation we present approaches for leveraging the wealth of social media documents available on the Web for search purposes and content filtering and characterization.

In this work, we address major challenges in searching temporal document collections. In such collections, documents are created and/or edited over time. Examples of temporal document collections are web archives, news archives, blogs, personal emails and enterprise documents. Unfortunately, traditional IR approaches based on term-matching only can give unsatisfactory results when searching temporal document collections. The reason for this is twofold: the contents of documents and queries are strongly time-dependent, i.e., documents discuss events that took place at particular time periods, and a query representing an information need can be time-dependent as well, i.e., a temporal query.

One problem faced when searching temporal document collections is the large number of documents possibly accumulated over time, which could result in the large number of irrelevant documents in a set of retrieved documents. Therefore, a user might have to spend more time in exploring retrieved documents in order to find documents satisfying his/her information need. A possible solution for this problem is to take into account the time dimension, i.e. extending keyword search with the creation or published date of documents.

### 1.1.1 Document Dating

During the recent years, the amount of user-contributed and digitized content on the Internet has dramatically increased, and makes web search even more challenging. Perhaps, the most useful tool the Web has to offer in order to make use of the vast amount of information is web search. Thus, the precision of search results is a very important factor, directly affecting the user satisfaction, engagement

and productivity. Although well-known search engines (e.g. Google, Bing, etc) deliver very good results for pure keyword searches, they still do not take full advantage of the temporal dimension that characterizes most document collections. A motivating example would be to extend keyword search with the creation or update time of the web pages/documents. In this way, the search engine would retrieve documents according to both text and temporal criteria, i.e., temporal text-containment search [14]. In addition to searching the current web, searching in old versions of web pages is sometimes useful. This can be of interest in large-scale archives like the Internet Archive.

However, in order for temporal text-containment search to give good enough and actually useful results, it is obvious that the timestamps of crawled, stored and indexed documents have to be as accurate as possible. In the case of local document archives, trustworthy metadata that includes time of creation and last update is available. However, in the case of web search and web warehousing, having an accurate and trustworthy timestamp is a serious challenge. One way to solve the problem, is to use the time of discovery as timestamp (i.e., the time a document/web page is first found by the web crawler). This will give an accurate timestamp if the creation time of a document and the time when it is retrieved by the crawler coincide in time. Unfortunately there is no guarantee that this is always the case. Another motivational example for research in the area of estimating a document's focus or creation time is that of old digitized documents or of partially failing optical character recognition applications (OCR). Moreover, a web page/document can be relocated and discovery time in this case will be very inaccurate. In some cases metadata about documents on the web can be retrieved but they can also in general not be trusted and often are simply just wrong.

As can be seen, in the case of web search and web warehousing it will in general be impossible to get trustworthy timestamps based on information acquired during crawling time. Thus, our research challenge is: for a given document with uncertain timestamp, can the contents of the document itself be used to determine the timestamp with a sufficient high confidence? To our knowledge, the only previous work on this topic is the work by de Jong, Rode, and Hiemstra [3], which is

based on a statistic language model. In this paper, we present approaches that extend the work by de Jong et al. and increases the accuracy of determined timestamps. Our main contributions in this paper are 1) a semantic-based preprocessing approach that improves the quality of timestamping, 2) extensions of the language model and incorporating more internal and external knowledge, and 3) an experimental evaluation of our proposed techniques illustrating the improved quality of our extensions.

Several related research efforts have focused on estimating a document's focus or creation time, mostly by the information retrieval community. Purely statistical methods have been proposed [9]. Other approaches have tried to deal with the problem by utilizing information from linguistic constructs with clear references to time periods or moments, by mentioning, for example, a specific date or year. Another line of work cosiders the entire vocabulary used used in a document in order to reason about when it was created [11]. Kanhabua and Nørvåg in [26, 27] propose a document-dating method that extends the one proposed by De Jong et al. Specifically, the authors propose the application of semantic-based preprocessing of the reference collection, and apply a term-weighting scheme based on their previous work on temporal entropy [26]. The authors further enhance their approach by considering search statistics from Google Zeitgeist.

A serious drawback and disadvantage of most proposed methods, that is being addressed in this dissertation, is that most methods initially pre-segment the timeline of study into intervals of the same fixed length (e.g. a week) and afterwards choose the interval that is most likely to be the temporal origin of the query document, by comparing its vocabulary with the model built for each of the candidate intervals. The drawback of this approach is obvious: it limits the choices of possible time intervals.

As the amount of social media content grows, research will have to identify robust ways to organize and filter that content. In this dissertation we aim to provide time-aware text processing techniques for organizing social media documents associated with events and popular social media content. With event identification, characterization,

and content selection and filtering techniques, we provide new opportunities for exploring and interacting with social media event data.

### 1.1.2 Memes and Events

As a motivational example, consider the part of the homepage of many social media sites that is devoted to *Trending Topics*. Most modern platforms like Twitter, Yahoo!, Facebook, etc. offer such a functionality, where different types of algorithms are used to identify the most popular topics in the platform during a current time window. *Trending Topics* lists may include popular items people search for in an e-commerce site like Amazon.com, trending queries in a web search engine like Google.com, popular topics of interest people write about in a micro-blogging platform like Twitter.com or trending tags people use to annotate their blog posts in a blogging site like Wordpress.com. Most of the items that appear in this lists have been caused by real-life events that triggered the interest of the users and they wrote or posted about them in the social media. They main functionality of the *Trending Topics* lists is that of facilitating search and discovery of new content. However, not all trending items are related to real life events. A significant percentage of popular content has become strategically popular, especially within microblogging environments like Twitter and Tumblr, where fan- and sports-related communities thrive and dominate the usage of the media. Thus, social media platforms and search engines would benefit from better understanding *why* a specific item became popular, and offer a variety of landing pages for different types of content. Specifically, Figure 1.1 illustrates the trending topics on December 28, 2014 in Twitter and Yahoo! respectively. A user clicking on a news-related trending topic would expect to land on a page with a series of news articles, maybe chronologically ordered, describing the timeline and the current state of that specific topic. On the other hand, a user clicking *'iPhone 6 Plus'*, which is a consumer product would expect to find offers, technical specs or reviews of the item. Last, users interested in a celebirty named *Nash*, would expect to find fanpages, photos and videos of the celebrity when clicking on `#HappyBirthdayNash` trending topic.

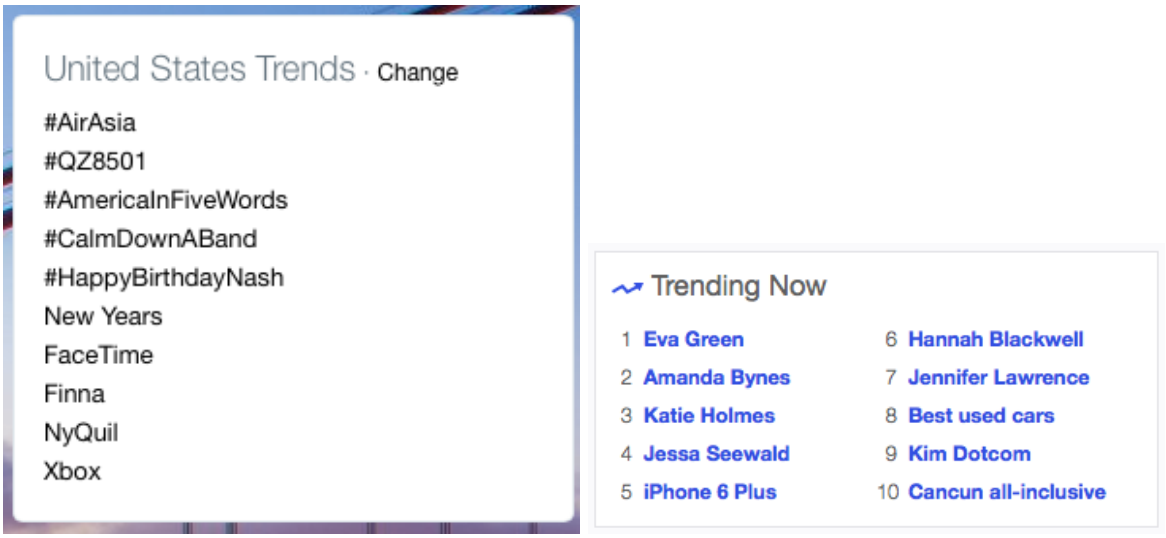


Figure 1.1: Trending Topics in Twitter.com and Yahoo.com on *December 28, 2014*

For our analysis of trending topics, we specifically focus on one social media site, namely, Twitter, due to its transient, large-scale publicly available content. In particular, we collected a vast amount of Twitter messages from various locations posted during various periods of time and focus on the analysis of the most popular hashtags. We show that event-detection methods can not only be based on the detection of terms and phrases that exhibit trending behavior on Twitter, characterized by an unusual increase in message frequency during a particular time period in a Twitter message stream. While some of these trends might refer to actual real-world events, others might include non-event information, triggered by strategically planned advertising campaigns or by user communities trying to promote themselves. To organize and understand this content, we define a taxonomy of popular content types, which includes trending events and trending memes. Unlike related efforts in this area, which focused on characterizing or analyzing content from individual events on Twitter, or characterizing aggregate trend characteristics for manually identified terms, the features we use in this study in order to discriminate between the various classes of content can be used for more than the two classes that are defined in this dissertation.

Overall, we show that social media sites contain substantial, useful information about different types of popular content that can be ex-

ploited and utilized in order to provide more effective and useful services to users of social media sites. With the features we propose in this dissertation, we can effectively identify different types of popular content and their associated social media documents across various social media sites. Regardless of the classifier we use, the type of event/meme, or the social media site, any single popular topic might have hundreds or thousands of associated social media documents. While some of these associated documents might contain interesting and useful information (e.g., event time and location in case of events, participants and opinions in case of memes), others might provide little value (e.g., using heavy slang, incomprehensible language without accompanying media) to people interested in learning about an event or meme. Techniques for effective selection of quality event content may then help improve applications such as event browsing and search. Therefore, we propose a noise-filtering mechanism for selecting a subset of the social media documents associated with the significant real-world events.

## 1.2 Contributions and roadmap of this thesis

In summary, the contributions of this dissertation are as follows:

- An event detection method that is based on the notion of term burstiness in document streams.
- An effective and efficient document timestamping algorithm that makes use of the burstiness detection framework in order to estimate a document's focus time based only on the textual content of the document.
- An extensive study of memes and events in social media, yielding in a complementary quantitative study, examining the differences between the two different types of popular content along various descriptive characteristics.

The remainder of this dissertation is organized as follows. In Chapter 2 we review the literature in event discovery in document streams, present a term burstiness modeling method and describe

a burstiness-based event detection framework which we applied in the context of blog posts. In Chapter 3 we deal with the document dating problem and present a state of the art method as of the time of the writing of this dissertation. Chapter 4 presents our approach on meme-filtering in document streams and evaluates the proposed method in the context of burstiness-based event detection. Chapter 5 presents a distributed stream monitoring framework and Chapter 6 concludes this dissertation.

# Chapter 2

# Events in Document Streams

## 2.1 Introduction

Everybody reads blogs. Almost everybody maintains one. Wikipedia defines a blog as a website, usually maintained by an individual, with regular entries of commentary, descriptions of events, or other material such as graphics or video. Over the last few years, blogs (web logs) have gained massive popularity and have become one of the most influential web social media in our times. Anyone with an internet connection can create his own blog for free, using web platforms developed for this specific reason (e.g. blogger.com, wordpress.com etc.). According to blog search engine Technorati.com there are over 175,000 new blogs every day, 1.6 million new posts per day and over 113 million blogs (not including millions of non-English blogs) exist today. The huge growth of blogging provides a wealth of information waiting to be extracted. Blog analysis and searching in blogs introduces new challenges for research in information retrieval because blogs' contents have a very specific characteristic not present in traditional web content: a timestamp exists in every blog post. Every blog post in the Blogosphere has a well defined value in the temporal axis. Traditional blogs' search engines don't take into account the temporal dimension and treat the blogs as plain web content; or just pay attention to the category tags that usually accompany a post. By taking into consideration the timestamp of each blog post we can try to detect the period in which the popularity of a specific keyword increases or decreases. Such functionality is important because it allows us to gauge the users' interests related to a specific topic over

time. Our contribution: In this paper we develop a technique to address the problem of identifying events in the Blogosphere. In our technique we apply Kleinberg's automaton ([2]) on extracted titles of blog posts to discover bursty terms, we introduce a novel representation of a term's burstiness evolution called State Series and we apply a Euclidean-based distance in order to discover potential correlations between terms without taking into account their context. Related work: As the number and size of large time-stamped collections increases this problem becomes more and more important [1], resulting in an evolution clearly presented in [3].

The main benefits of our method are that it runs in linear-time and is also completely parameter-free. This makes it ideal for very large sequences of documents, spanning significant periods of time. That being said, our search framework is compatible with any burst detection method that can report non-overlapping bursty intervals and their respective scores, for any given term.

## 2.2 Our Approach

We search for events in the Blogosphere. We define an event in the Blogosphere as a small subset of keywords able to describe one or more real life events that occurred during the period of study. To discover them we try to identify correlated bursty terms, meaning bursty terms whose burstiness exhibits a similar behavior in the temporal axis. A burst is marked whenever the popularity of a specific keyword dramatically and unexpectedly increases. Doing so, we omit taking into account a keyword's possible co-existence with another keyword in the same title. Ignoring all those keyword pairs enables us to gain significant computational time and to search for conceptually correlated keywords although they may not appear in the same document (e.g. separately used synonyms).

In order to identify bursty terms, meaning specific words whose appearances increase radically in short periods of time in comparison to the long period we study, we use the technique proposed by Kleinberg [2] as decribed in [4]. Afterwards we evaluate the accuracy of

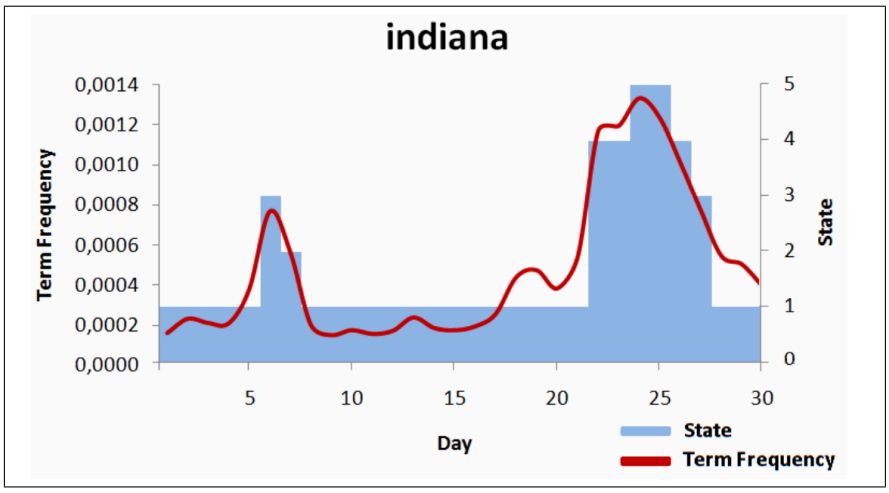


Figure 2.1: Frequency curve and SS of the keyword "indiana"

the bursty terms by trying to match them with real life events that took place in the bursty period of time. A certain event is formed by a group of correlated terms. As the popularity of a specific topic diminishes, this group ceases to exist. We try to obtain keywords' correlations, in order to automatically identify such groups. We address this problem, assuming that related keywords produce similar activity as far as burstiness is concerned. A mechanism for burstiness representation of a term $t$ called State Series is introduced and is defined as follows:

$$SS_t = (s_{t1}, s_{t2}, ..., s_{tn})$$

where $s_{ti}$ represents the burstiness state of term $t$ at timestamp $i$, produced by the automaton. Figure 2.1 compares the frequency curve of the term *'indiana'* as it appeared in our dataset to the corresponding $SS$, proving that the latter is a satisfactory representation. Furthermore we employ a Euclidean-based distance metric to calculate the dissimilarity between the $SS$s of two different terms. Finally, we obtain events by accumulating the $5$ Nearest Neighbors for each keyword, assuming that $5$ terms can adequately describe an event. Last but not least we evaluate these events by trying to pair this topic with a real life event that took place in the period of study.

## 2.3 Related Work

### 2.3.1 Event Detection

Event detection in social media and document streams has attracted the interest of the relevant research communities over the last few years, with the widespread expansion of user-generated content plaforms. Specifically, event detection is the task of identifying event related messages and documents from a document or article stream. Users of social networking and microblogging platforms tend to provide plenty of information about what is happening in the world, ranging from very personal messages like what happened in their work some morning or a minor car accident to messages describing globally important topics like the U.S. elections or an earthquake. Thus, exploiting the vast amount of available information in order to reason about what is happening in the world seems as a natural consequence for the information retrieval community. The main idea is that messages that are about real-world events have different structure and content in general in comparison to the rest of the messages that are about users' personal lives. The identified events could reflect a natural disaster such as a flood or an earthquake or a show of global significance like the Oscar ceremony. As a result, a definition of what an event really is can be very vague and the proposed approaches for different types of event detection may pose significant differences.

Most of the proposed methods model event detection as a clustering problem. The clustering may be performed on the documents' textual features (an approach denoted as topic clustering) or on their spatio-temporal aspects (an approach denoted as spatio-temporal clustering). Both approaches group messages into clusters. Some of these clusters correspond to real events while others just contain similar messages. The identification of event clusters is often performed with the aid of scoring functions or supervised classifiers. Other approaches deal with the problem using novelty tests such as [Petrovic et al. 2010] and others focus on sentiment peaks [Valkanas and Gunopulos 2013] or on keyword bursts[Abdelhaq et al. 2013]. The common part of almost all approaches is that a ”change detection” module is necessary in order to detect an event. This *change* could be in terms

of term frequency or of network structure such as an increasing number of new connections in the social graph, which indicates a burst as well.

### 2.3.2 Term Burstiness

A considerable amount of work has been devoted to developing efficient burst-detection methods [RRR10, 11, 18, 19]. The concept of burstiness has been studied in several domains. A significant portion of this work has been inspired by Kleinberg's seminal paper on the bursty and hierarchical structure of streams [RRR13].

Kleiberg's algorithm is based on a Hidden Markov Model, with states that correspond to frequency levels for individual terms. State transitions (bursts) correspond to points in time, around which the frequency of a term changes significantly and unexpectedly. Given the frequency sequence $Y_t$ of a term $t$, dynamic programming is used to fit the most possible state sequence that is likely to have generated the sequence $Y_t$. The state assigned to each interval will serve as its burstiness score. For the rest of this chapter, we refer to this algorithm as KLEIN. Another burst-detection method is presented by Fung et al.[RRR10]. In this work, bursty terms are clustered to represent events discussed in the data. In [RRR11], the authors classify terms in four burstiness categories, based on their frequency trajectory. Their use of spectral analysis is similar to the one used by Vlachos et al. in [RRR18], where the authors focus on periodic and bursty artifacts in query logs. In [RRR19], the authors use a wavelet-based structure for aggregate monitoring of data streams.

Pioneered by the Kleinberg's automaton model described above, many techniques have been proposed for burst detection such as the $\chi^2$-test based method proposed by Swan and Allan [**?**] and the moving average method proposed by Vlachos et al. [47].Burstiness has also been evaluated in the context of other applications, such as stream clustering [RRR12], and even in the context of graphs [RRR14]. Further, He et al. [RRR16] apply Kleinberg's model to topic clustering. Bansal and Koudas [RRR2, RRR3] have presented a system for the

analysis of streaming blogs. Even though no details on the employed methods are given, their work is relevant to ours, in that they ultimately map bursty terms to specific blogposts. Yin et al. developed a burst-detection module that continuously monitors a Twitter feed to identify unexpected incidents. The proposed method raises an alert for immediate attention when it detects an unexpected incident. To achieve real-time efficiency, they adopt a parameter-free algorithm to identify bursty words from Twitter text streams in their system. The basic idea is to determine whether a word is bursty on the basis of its probability distribution in a time window [**?**]. Zhao et al. propose to identify event-related bursts via social media activity data. They study how to correlate multiple types of activities to derive a global bursty pattern. To model smoothness of one state sequence, they propose a novel function which can capture the state context [**?**].

In [VLDB2012] Lappas et al. studied spatiotemporal term burstiness and how it relates to real-life events. More specifically, thousands of documents published daily in online news sites, blogs and microblogs record real-life events. By collecting these documents the authors created a spatiotemporal collection that consists of document streams from different locations, e.g. countries, cities, etc. As mentioned above, during an event's time, characteristic terms that relate to the corresponding event exhibit atypically high frequencies in the document collection. Moreover, these terms are unexpectedly popular in the affected locations too. Spatiotemporal burstiness can be utilized in a variety of settings, like document search, document selection or trends and events identification. More specifically, given a query $q = t_1, t_2, \ldots, t_n$ a search engine can retrieve documents related to events with strong spatiotemporal footprint on the document collection, namely events that affected a lot of users in a variety of locations for extended and bounded time periods. The identification of spatiotemporal patterns can assist algorithms regarding trend detection in document streams, in that using bursty terms we can reason about when and where items related to the bursty terms were popular. More specifically, given a term $t$ and a document collection from different locations, Lappas et al. formalized spatiotemporal burstiness patterns and presented efficient algorithmic methods in order to identify and evaluate them. In particular, they studied combinatorial patterns and

regional patterns. Combinatorial burstiness patterns encode that unusually high frequencies were simultaneously observed for term $t$ in all streams in the document collection, during the same temporal interval $I$. Regional burstiness patterns consider the geographical proximity among document streams and encode that unusually high frequencies were observed for term $t$ in some specific geographical region $R$ during a temporal interval $I$.
In this dissertation we present an algorithm to independently extract the sets of bursty time intervals for each independent document stream and use it to search for events in the blogosphere.

## 2.4 Finding the bursty intervals

In [L17] the authors present a linear-time algorithm for solving the All Maximal Segments Problem. The algorithm accepts as input a sequence of real numbers and reports the set of all maximal segments. For the rest of this dissertation, we refer to this algorithm as `GetMax`. The details and pseudocode of the algorithm can be found in [L17] and in short description below. `GetMax` filters out maximal segments with a negative score. This is ideal for the purposes of burstiness evaluation, since negative-scoring intervals represent regions where the observed frequency of a term was less than the expected. Finally, in addition to being linear, the approach is completely parameter-free. Next, we present an extension of MAX-1 and discuss its advantages.

In [L13], Kleinberg discusses anisochronies, the non-uniform relationships between the time spanned by a story's events and the amount of time devoted to these events in the actual telling of the story. Considering the coverage of events in news streams (e.g. newspapers, blogs), we identify two primary levels of bursty behavior for the terms describing an event: the first level represents the extended time period when the event was generally discussed in the news. Depending on the nature and significance of the event, this period can be extended to include weeks or even months. The second burstiness level pertains to smaller intervals within this extended period, when the event was particularly popular and extensively covered in the news. In the

context of a newspaper, such intervals may represent the first time an event made the headlines, or a new development in an older event that brings it back to the front page.

Conceptually, the intervals reported by `GetMax` capture the first level of burstiness activity for a given term. By reapplying the algorithm on each of the reported maximal intervals independently, we can easily identify the second- level burstiness intervals. Multiple iterations of `GetMax` could be used to obtain a hierarchical structure of the bursty intervals. For the rest of this dissertation, all experimental results refer to a single iteration, since we found that it is enough to capture the burstiness patterns of events.

The `GetMax` algorithm was introduced in [31]. Given a discrete time series of frequency measurement for a given term, `GetMax` returns the set of non-overlapping bursty intervals. A brief description of the algorithm is given below.
The `GetMax` algorithm computes a set of bursty intervals, after reading the time series consisting of the frequency values for a term from left to right. A *burst* on the timeline is marked whenever the popularity of a specific term dramatically and unexpectedly increases. To identify the bursty intervals we use the getmax algorithm which is briefly described below. Segments that are candidates for maximality, and thus candidate bursty intervals, are kept in a list $L$. For each candidate $l_j \in L$, we record the sum $l_j.L$ of all scores up to the leftmost score of $l_j$ (exclusive) and the sum $l_j.R$ up to the rightmost score of $l_j$ (inclusive). Non-positive scores require no special handling. If a positive score is read, a new sequence $l_k$ containing only this score is created and processed as follows:

1. Search the list $L$, from right to left, for the maximum value of $l_j$ satisfying $l_j.L < l_k.L$.
2. If there is no such $l_j$, then append $l_k$ to the list $L$.
3. If there is such a $l_j$, and $l_j.R \geq l_k.R$, then append $l_k$ to the list $L$.
4. Otherwise (i.e., there is such a $l_j$, but $l_j.R < l_k.R$), extend $l_k$ up to the leftmost score in $l_j$ (inclusive). Remove candidates

$l_j, l_{j+1}, ..., l_{k-1}$ from $L$ (none of them is maximal) and reconsider the newly extended segment $l_k$ (now numbered $l_j$) from step 1.

After the entire input has been processed, the candidates left in the list $L$ are the maximal segments representing the bursty intervals on the timeline [41][31].

## 2.5 Experimental Evaluation

Our experimental evaluation shows that blog posts' titles prove sufficient to mine the underlying bursts. Not using the whole body of each blog post reduces the total computational time required. Therefore, we extend this approach to search for events through the burstiness pattern of keywords appearing in blog posts' titles. Data description: We experimented on posts from millions of blogs around the web's free blog hosts (e.g. blogger.com, wordpress.com, livejournal.com etc.) After some pre-processing of our initial dataset we ended up with $11,198,076$ titles containing $38,814$ different keywords with various appearances during the period May 1 – May 30, $2008$.

We used an $n$-state automaton, incrementing $n$ and monitoring the percentage of the terms with altered 5-NNs in comparison to the results of the n-1-state automaton. As shown in Figure2.2, the greater state value that could be reached was 13. The ratio of the exponential rate of the automaton's each subsequent state to the rate of the previous state was picked to be $1.3$, after several experimental trials. This value provides us with increased diversity in the state series. The automaton identified $21.53$% of the terms as bursty.

Semantic evaluation: A visualized example of identifying bursty keyword correlations using the SS similiarity is depicted in Figure 2.3, where the SSs for the 3-NNs of the term Indiana are shown. The terms *indiana*, *jones*, *crystal* and *skull* appear in the results as bursty ones. While trying to evaluate the accuracy of this result, we found out that on May 22nd 2008 the movie *"Indiana Jones and the Kingdom of the Crystal Skull"* was released. Additional results shown in Table 2.1

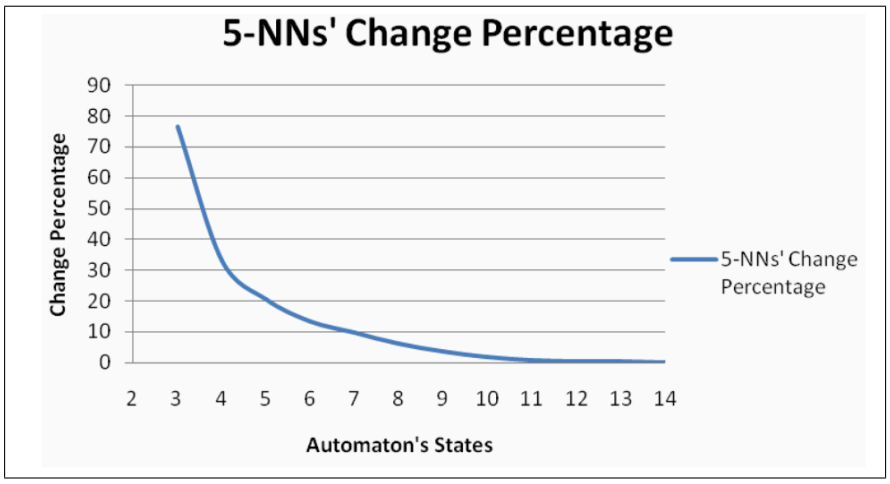


Figure 2.2: Changing the number of states

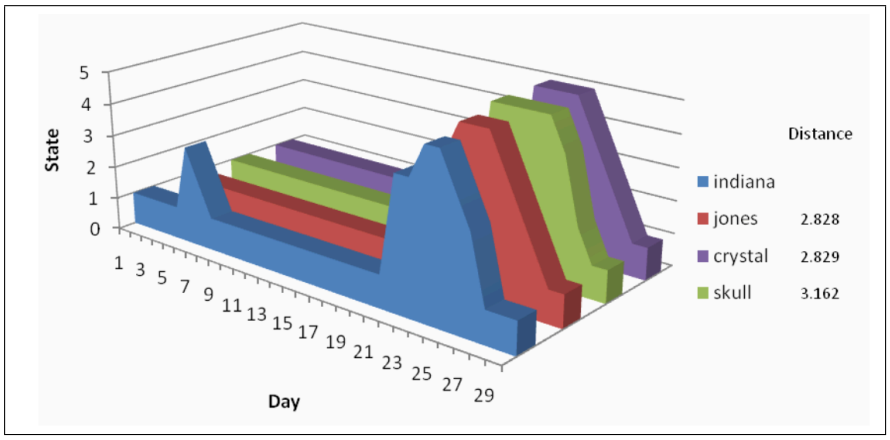


Figure 2.3: A comparison of the SS of related keywords

Table 2.1: Semantic evaluation results

| **Term** | 5**-NNs** | **Bursty Intervals** |
|---|---|---|
| `pharaoh` | {*physique,feminine,akhenaten,liver,transplant*} | Mar 2 - Mar 3 |
| `liver` | {*transplant,marijuana,wig,feminine,physique*} | Mar 2 - Mar 3 |
| `myanmar` | {*burma,burmese,appreciation,chait,brutality*} | Mar 5 - Mar 14 |
| `cialis` | {*tadalafil,trent,prescription,pharmacy,impotence*} | Mar 5 - Mar 6, Mar 19 - Mar 20 |
| `indiana` | {*jones,crystal,kingdom,skull,islander*} | Mar 6 - Mar 7 |

add to the safe assumption that events can be mined through the extraction of state series.

As seen in Figure 2.3, in this case the proposed method came out to be resistant to effects of other bursts of a term, that seem to be irrelevant to the event being described by the 5-NNs; indiana exhibits two bursts during May, one lasting from 6th to 7th day and one from 22nd to 27th day, but the former one does not affect the high similarity between indiana and the other three terms.

# Chapter 3

# A Burstiness-aware Approach for Document Dating

## 3.1 Introduction

Temporal text mining is at the core of a large number of mainstream applications. The input to such applications consists of a collection of documents, with each document being associated with the timestamp of its creation. This temporal dimension can then be used for, among others, event detection [3], document search [31], rule mining [36], topic and trend tracking [35], classification [42], clustering [4] and text summarization [48].

The assumption made by all such applications is that the timestamp of each document is both available and accurate. In practice, however, this assumption can be false. A characteristic instance emerges in the context of large repositories of old digitized documents. Such repositories are becoming increasingly large and abundant, due to initiatives such as The National Digital Newspaper Program [1] by the Library of Congres, and other similar ventures for the digitization of periodicals by large corporations such as Microsoft and Google. In these cases, the timestamp may be corrupted during the digitization process, or may simply be unavailable due to the decay of the original.

Another example of timestamp ambiguity comes with the domain of online articles. Even if an article is discovered immediately after it has been uploaded, there may be an arbitrarily large discrepancy between

the date it was uploaded and the date it was originally written. This is a typical phenomenon that occurs when older digitized documents are made available online.

When searching temporal document collections, it is difficult to achieve high effectiveness using only a keyword query because the contents of both documents and queries are strongly time-dependent. Possible solutions to increase the retrieval effectiveness are, for instance, extending keyword search with the publication time of documents, or automatically re-ranking retrieved documents using time. Incorporating the time dimension into search will increase the retrieval effectiveness if a document is assigned to its time of creation or publication date. However, for many documents, like the ones mentioned in the previous paragraph, it is difficult to find an accurate and trustworthy timestamp. In a web warehouse or a web archive, there is no guarantee that the creation time and the time of retrieval by a web crawler are related. The purpose of determining time for non-timestamped documents is to estimate the time of publication of a document or the time of the topic the document discusses. The process of determining the time of documents is called *document dating* or *document timestamping*.

There has been significant work addressing the problem of estimating the timestamp of a document given a collection of timestamped documents. Document Dating through content has been addressed significantly however it is known to be a difficult problem. Unless a text has a specific mention of a date in its contents, it is almost impossible to identify the creation time of documents that do not discuss specific events. This could be accomplished for very large granularity, where the writing style and word frequencies change, but in useful time intervals that is not the case. Even for documents that discuss timely events, the reported interval can be quite large. For instance if there is a text that talks about the Obama presidency, without any other temporal information, it can be dated to a granularity of the eight years of his terms, however reporting a more limited timeframe is very difficult

In this chapter we describe a content-based, purely statistical method

for approximating the true timestamp of a given document. Our approach reports timeframes of arbitrary length for a query document. We address the problem by considering two main factors, (i) the lexical similarity between the query document and the documents in the reference corpus $\mathcal{D}$, and (ii) the burstiness of the terms in the query document.

The first factor captures the intuition that lexically similar documents are more likely to discuss the same topics and events, and are thus more likely to be associated with adjacent timestamps. The second factor builds upon work on term burstiness [31] presented in Chapter 2, in which we described an algorithm for identifying the timeframes of bursty activity for a given term in the context of a sequence of documents. At a high level, a timeframe is considered bursty if the term exhibits atypically high frequencies for its duration. Bursts in terms frequency capture in essence the trends in vocabulary usage during each corresponding timeframe and can thus prove useful in document dating.

The second factor aims to take advantage of the fact that when an event takes place in real life (e.g. a major earthquake, sports finals), the event's characteristic terms (e.g. ``earthquake'', ``shooting'', ``overtime'') appear more frequently in the media. In the context of document dating, our intuition is that a timeframe that is bursty for many of the terms in the query document is more likely to overlap with the document's true (but unknown) timestamp. Looking again at Fig. 3.1, it is clear that the query document should be placed in June 2013, since the terms it contains would exhibit a bursty behavior during that period.

The proposed algorithm is more flexible and more effective than previous approaches. Our method is the first one to utilize temporal information through a burstiness-aware approach, without depending on specific language rules, datasets, or meta-information.

**Our contributions:** The contributions of this chapter can be summarized as follows:

- We propose a novel, purely statistical algorithm for estimating the

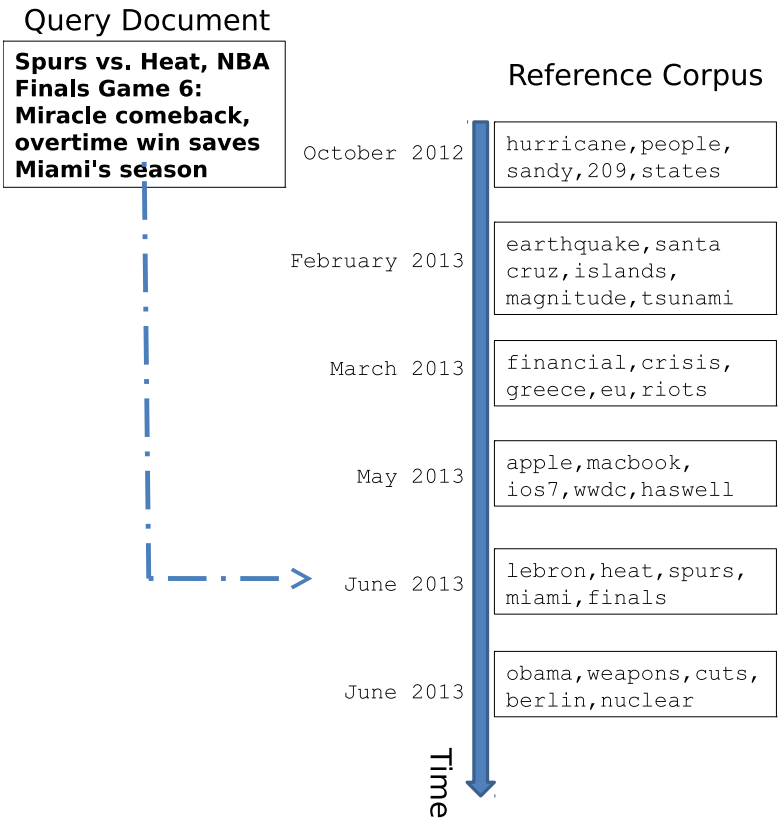


Figure 3.1: An example for a document dating application. In this case, a document dating algorithm would assign to the query document a timestamp sometime around June 2013.

timestamp of a document based on its content and burstiness.

- Our approach reports non-fixed periods of time, in contrast to previous approaches, that report one timeframe among the pre-segmented timespan of the reference corpus.
- We provide an extensive experimental evaluation, by using three different datasets spanning different time periods.

The organization of the rest of this chapter is as follows. In Section 3.2, we give an overview of the related work on the document dating problem. In Section 3.3, we outline preliminaries that will be used as the basis of our approach and formally define the problem. Section 3.4 presents our approach in detail. In Sections 3.5 to 3.7, we describe our experiments, evaluate the proposed technique and compare with the state of the art. Finally, in Section 3.8, we give conclusions.

## 3.2 Related Work

The problem of document dating has proven to be a really tough one for the information retrieval community, as even the most recent methods and best published results do not achieve more than $50\%$ precision for estimating 1-year long intervals in corpora that span 10 years, using purely statistical methods [9]. The underlying reason for this, is that not all documents contain temporal information, which makes a significant percentage of the corpus useless for testing and training purposes.

Previous literature on determining the time of a document can be categorized into 2 clusters: learning-based and non-learning-based methods. The difference between the two clusters is that the former determines the time of a document by learning from a set of training documents, while the latter does not require a corpus collection.

Non-learning methods are presented in [N77, 81, 93]. They require an explicit time-tagged document and try to address the problem of document dating by identifying linguistic constructs with a clear temporal interpretation (e.g. the mention of the date or time). In addition

to being sparse, such tokens can also be ambiguous, referring to irrelevant timeframes. In order to determine the time of a document, each time-tagged word is resolved into a concrete date and a relevancy of the date is computed using the frequency of which the date appears in the document. The most relevant date is used as a reference date for the document, however, if all dates are similar relevant, the publication date will be used instead. In the end, the event-time period of the document is generated by assembling all nearly dates to the reference date where their relevancy must be greater than a threshold. Nunes et al. [N93] propose an alternative approach to dating a non-timestamped document using its neighbors, such as 1) documents containing links to the non-timestamped document (incoming links), 2) documents pointed to the non-timestamped document (outgoing links) and 3) the media assets (e.g., images) asso- ciated with the non-timestamped document. They compute the average of last-modified dates extracted from neighbor documents and use it as the time for the non-timestamped document.

Learning-based methods are presented in [N29, 116, 115]. In [N116, 115], they use a statistical method called hypothesis testing on a group of terms having an overlapped time period in order to determine if they are statistically related. If the computed values from testing are above a threshold, those features are coalesced into a single topic, and the time of the topic is es- timated from a common time period associated to each term. Another method presented by de Jong et al. in [11] is based on a temporal language model where the time of the document is assigned with a certain probability and characterizes a specific line of works. These works consider the entire vocabulary of a document in order to identify its timestamp. While this is a clear improvement over approaches that rely only on linguistic constructs, these methods are limited by their static consideration of the candidate timeframes.

Initially, these methods pre-segment the timeline into intervals of the same fixed length (e.g. a week). A language model is then used to select the interval that is most likely to be the temporal origin of the query document, by comparing its vocabulary with the model built for each of the candidate intervals.

Kanhabua and Nørvåg in [26, 27] propose a document-dating method that extends the one proposed by De Jong et al. Specifically, the authors propose the application of semantic-based preprocessing of the reference collection, and apply a term-weighting scheme based on their previous work on temporal entropy [26]. The authors further enhance their approach by considering search statistics from Google Zeitgeist.

Chambers [9] proposed a discriminative model, using a Maximum Entropy classifier, as well as defining rules for processing temporal linguistic features, as year mentions in documents. While this model outperformed the methods proposed by De Jong and Kanhabua, it has the limitation that it only works well for *year* predictions, because temporal linguistic features that refer to months or days are ambiguous. Moreover, in this study there were no running time experiments, an issue that we address in the current paper.

All of the above approaches are limited by the fact that they require a pre-segmentation of the timeline into fixed intervals. Our approach has not such requirements and can handle intervals of arbitrary length. In addition it can report results various time-intervals without retraining.

In [15] Garcia-Fernandez describes an approach for determine time with particular emphasis on older document. The methods proposed in the paper are based on determining time of named entities in the text based on Wikipedia (for example birth date of people) and knowledge of neologisms or archaisms. These features are integrated in a classification approach presented in the paper. While these techniques might be interesting for larger granularity detection liked decades and centuries, they are not useful for smaller granularities, and also they are language-dependent.

Comparing the non-learning to learning-based methods, both of them return two different aspects of time. The first line of work focuses on the time of events that appear in the document content, while the latter one focuses on the most likely document creation time interval. In this chapter, we focus only on purely statistical and content-based meth-

ods because information about links is not available in all domains, and content-based analysis seems to be more practical for a general search application.

The concept of burstiness is a central component of our approach. In this chapter, we use the method that we introduced in Chapter 2. for term burstiness and document search. Our choice is motivated by the parameter-free nature, as well as its linear-time complexity. Nonetheless, our document-dating framework is compatible with any method that can identify bursty intervals given a sequence of frequency measurements.

## 3.3 Problem Definition

### 3.3.1 Preliminaries

The problem we address in this paper is defined in the context of a collection of documents $\mathcal{D}$, spanning a timeline of $Y = t_1, t_2, ..., t_n$ of $n$ distinct timestamps. We define a function $t(d)$ to return the timestamp of a given document $d \in \mathcal{D}$. Given a query document $q \notin \mathcal{D}$, for which the timestamp $t(q)$ is unknown, our goal is to find the best possible interval of size $\ell$, $I = t_i, ..., t_{\ell+i}, 1 \leq i, j \leq n$ within $T$, so that $t(q)$ most likley falls within $I$. Throughout the paper, we refer to $\mathcal{D}$ as the *reference corpus*

Among other things, our approach considers the *burstiness* of the terms in the query document $q$. Given a term $x \in q$, we use $\mathcal{B}(x, \mathcal{D})$ to represent the set of non-overlapping bursty intervals for $x$, as computed over the given corpus $\mathcal{D}$. Each bursty interval is defined within the timeline $T$ spanned by $\mathcal{D}$. In addition, we define $s(b)$ to return the burstiness score of a given bursty interval $b \in \mathcal{B}(x, \mathcal{D})$. Since we are only considering a single corpus, we henceforth refer to $\mathcal{B}(x, \mathcal{D})$ simply as $\mathcal{B}(x)$.

In order to evaluate the effectiveness of our method, we compare its precision against state of the art methods. Given the variety of our datasets as well as their uniform distribution of the time periods exam-

ined, we believe that the ability to place a document within a desired timeframe, in other words the precision we achieve, to be the most accurate valuation.

### 3.3.2 Problem Definition

The examples described in section 3.1 motivate the need for an algorithm that is able to estimate the timestamp of a given document based on its content. At a high-level, the problem can be defined as follows:

**Problem 1 [Document Dating]:** *Let $\mathcal{D}$ be a collection of documents spanning a timeline of $T = t_1, t_2, ..., t_n$ of $n$ discrete timestamps (e.g. days). Each document $d \in \mathcal{D}$ is associated with exactly one timestamp from $T$. Let $q \notin \mathcal{D}$ be a query document for which the timestamp is unknown. Then, we want to find we want to find the smallest possible timeframe within $T$ during which the document was written.*

The definition of the problem assumes that the query document was written within the timeline spanned by the corpus $\mathcal{D}$. No constraints are placed on the size or nature of $\mathcal{D}$. We observe that the presence of such a reference-corpus is necessary, otherwise it would be impossible to arbitrarily assign a timestamp to $q$.

## 3.4 Our approach

In this section, we introduce our algorithm for the Document Dating problem. As mentioned in the introduction of this chapter, our approach considers (i) the lexical similarity of the query document with the documents in the reference collection $\mathcal{D}$ (ii) the burstiness of the significant terms of the query document $q$, e.g., top-$k$ terms ranked by *tf-idf*.

The use of lexical similarity captures the intuition that similar documents are more likely to discuss similar topics and events, and are

thus more likely to originate in the same timeframe. In practice, however, similar documents may appear on different timestamps across the timeline. We address this, by introducing term burstiness. When an event or topic is recorded in a textual corpus, its characteristic terms exhibit atypically high frequencies. We refer to these timeframes as *bursty intervals*. Our algorithm is orthogonal to the actual mechanism used for computing non-overlapping bursty intervals. By identifying the bursty intervals of different terms, we can identify the timeframe of relevant events, as well as relevant documents that discuss them.

A conceptual view of our approach is given by the example in Figure 3.2. Figure 3.3 illustrates the architecture of our approach. In this section we describe our steps in detail.

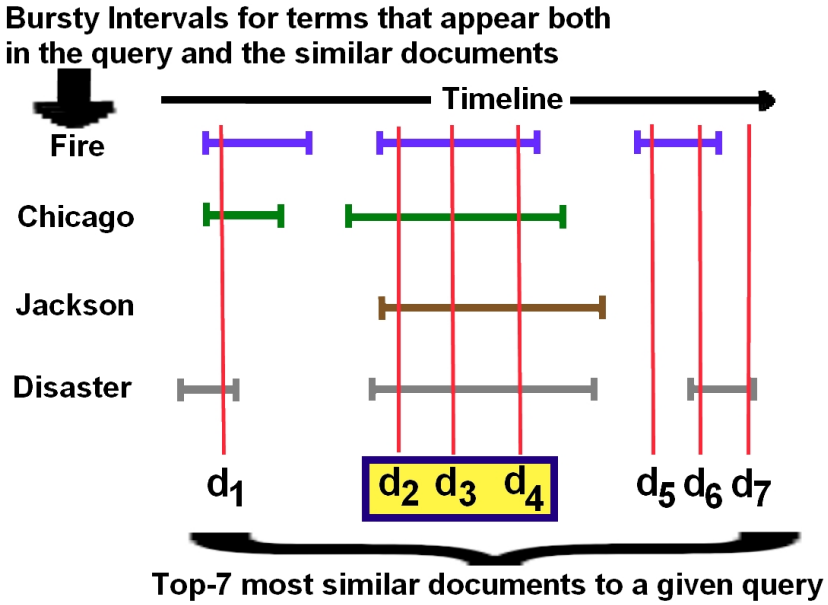


Figure 3.2: An example of how `BurstySimDater` identifies the appropriate timestamp for a given query document. In this case, the three documents $d_2, d_3, d_4$ will be selected by our algorithm, since they are both close to each other and overlap with multiple bursty intervals of the considered terms.

We are given a query document $q$, discussing a disastrous fire in the Jackson theater in Chicago. The figure shows the $7$ most lexically similar documents to $q$: $d_1, d_2, d_3, d_4, d_5, d_6$ and $d_7$ . Each of these docu-

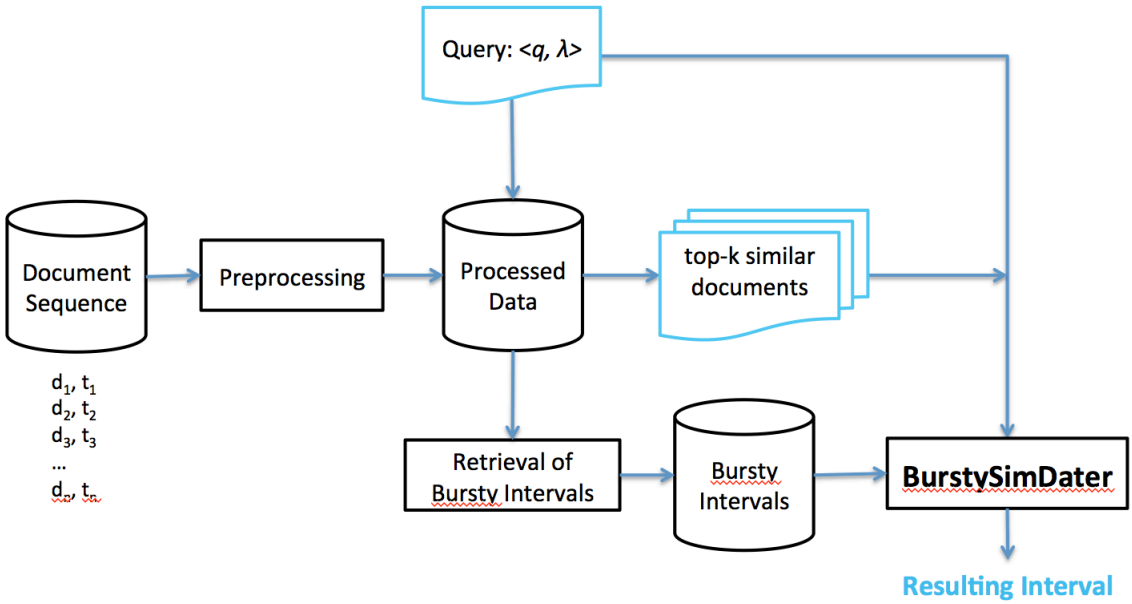


Figure 3.3: The architecture of our approach

ments has a subset of the following four terms in common with $q$: *Fire, Chicago, Jackson, Disaster*. The figure shows the bursty intervals for each of these terms. In this example, there are three visible sets of neighboring documents: $\{d_1\}$, $\{d_3, d_4, d_5\}$ and $\{d_5, d_6, d_7\}$. The documents in the second set overlap with multiple bursty intervals from the four characteristic terms, and are thus more likely to discuss the the actual event. Therefore, our approach will report the interval that starts with the first document on the ($d_2$) and ends with the last one ($d_4$) as the most likely timeframe for the query document.

We refer to our algorithm as `BurstySimDater`. The pseudocode is given in Algorithm 3.1.

The input to the algorithm consists of the query document $q$, the reference corpus $\mathcal{D}$, the set of precomputed bursty intervals $\mathcal{B}$ and the upper bound on the reported timeframe $\ell$. The output is an interval of length at most $\ell$, within the timeline $T$ spanned by $\mathcal{D}$.

First, the algorithm retrieves the top-$k$ most similar documents to $q$ from $\mathcal{D}$. In our own evaluation, we experimented, among others, with

**Algorithm 3.1** `BurstySimDater`

**Input:** reference corpus $\mathcal{D}$, bursty intervals $\mathcal{B}$, query document $q$, max timeframe length $\ell$
**Output:** timeframe of $q$

1: $\mathcal{S} \leftarrow$ top-k most similar documents to $q$ from $\mathcal{D}$
2: $W_{\mathcal{S}} \leftarrow \emptyset$
3: **for** $d \in \mathcal{S}$ **do**
4: $\quad w_d \leftarrow 0$
5: $\quad \mathcal{Y} \leftarrow d \cap q$
6: $\quad$ **for** $x \in \mathcal{Y}$ **do**
7: $\quad\quad w_d \leftarrow w_d + |\{I \in \mathcal{B}(x) : t(d) \in I\}|$
8: $\quad w_d \leftarrow w_d / |\mathcal{Y}|$
9: $\quad W_{\mathcal{S}} \leftarrow W_{\mathcal{S}} \cup \{w_d\}$
10: $A_{\mathcal{S}} \leftarrow (d \in \mathcal{S}, W_{\mathcal{S}})$
11: $\mathcal{I} \leftarrow \texttt{GetMax}(A_S, \ell)$
12: **Return** $\mathcal{I}$

the *tf-idf* measure and the Jaccard similarity. We use the latter in the experimental section of this paper, since it led to the best results. We refer to the retrieved set of the $k$ most similar documents as $\mathcal{S}$.

In steps 2-9, we assign a weight $w_d$ to each document in $d \in \mathcal{S}$, based on its overlap with the burstiness patterns of its terms. Initially, $w_d$ is set to zero. Let $\mathcal{Y}$ be the overlap of $d$'s vocabulary with the vocabulary of the query document $q$. For each term $x \in \mathcal{Y}$, let $\mathcal{B}(x)$ be the pre-computed set of bursty intervals for $x$. We then increment $w_d$ by the number of the intervals from $\mathcal{B}(x)$ that actually contain $t(d)$. After the iteration over all terms in $\mathcal{Y}$ is complete, we normalize $w_d$ by dividing it by $|\mathcal{Y}|$. Conceptually, the weight $w_d$ of a document $d$ is the average number of bursty intervals that it overlaps with, computed over all the terms that it has in common with the query $q$. The computed weights are kept in the set $\mathcal{W}_{\mathcal{S}}$.

We want to identify the interval when the most terms from the top-$k$ similar documents are *simultaneously* bursty. This period is the intersection of intervals with the maximum sum of weights. To do this, in steps 10-11, we create an array $A_{\mathcal{S}}$ of size $T$, where cell $i$ equals to the sum of weights $w_d$ for all documents $d \in \mathcal{S}$ that were written at $t_i$. Next, we find the interval $\mathcal{I}$ of length $\ell$ with the maximum sum. By

tuning $\ell$, we tune the level of desired accuracy. In order to compute the sets of bursty intervals we use the `GetMax` algorithm [32]. Given a discrete time series of frequency measurements, `GetMax` returns a set of non-overlapping bursty intervals with respect to the frequencies.

A *burst* on the timeline is marked whenever the popularity of a specific term dramatically and unexpectedly increases. In order to compute the sets of bursty intervals we use the `GetMax` algorithm, introduced in [31]. Given a discrete time series of frequency measurements for a given term, `GetMax` returns a set of non-overlapping bursty intervals with respect to the number of frequency measurements.

### 3.4.1 Complexity of the Algorithm

The running time of our algorithm depends critically on the time required to compute the top-$k$ most similar documents to the query (line 1 in the pseudo-code), and the time required to find the maximum sum intervals (line 11). The bursty intervals of a word can be computed in linear time ([31],[41]), hence the complexity of this process is $O(tw)$, where $t$ is the number of days in the timeline and $w$ is the number of words, in order to find all the bursty intervals of each word. This step however is only performed once as we train our model. For each query document, we have to find the top-$k$ most similar documents. In order to do that, we need to iterate over all documents in the corpus $\mathcal{D}$ and compute a similarity function. However in our case, the number of words of a document has an upper bound and thus this can be considered a constant factor $c$ in the overall complexity of our algorithm. Therefore, this step requires $c * |D|$ steps, but remains $O(|D|)$. For each of the similar documents, we have to assign a score based on how many of its words were bursty during the time of writing, which takes $O(1)$ if we have stored a binary array of size $t$ for each term in the corpus. The final step is to find the interval of size $\ell$ with the maximum sum of scores. Since we know the size, this can be done in linear time.

Given that the number of distinct terms in each document has an upper bound, which can be considered constant, the complexity of our

algorithm for each query is $O(|D|)$. In order to find the the intersection of intervals with the maximum sum of weights, we can create an interval graph $G_I$ and compute the maximum weight clique (MWC). Our algorithm identifies for each day in the timespan the top-$k$ similar documents to our query document, computes their weights and adds them. (Fig. 3.4(a)) For each day, we create an interval with length $\ell/2$ before and $\ell/2$ after the timestamp of that day. (Fig. 3.4(b)) We then create an interval graph from these intervals, where each node is weighted by the sum of the scores of that day. An edge is added between two nodes, if there is an intersection between the corresponding intervals. If we were to compute the maximum weight clique (MWC) of graph $G_I$, it would produce a subset of similar documents that are all close to each other and also maximize the sum of their respective weights. The reported interval would be the one that extends from the smallest (earliest) to the largest (latest) timestamp within the clique. MWC in this graph can be found in linear time, since the interval nodes are sorted [18]. The intersection of intervals with the maximum sum of weights $\mathcal{I}$ is the same with the corresponding interval of the MWC in $G_I$.

This can be proved if we create an equivalent unit interval graph $\mathcal{G}_S$ from $\mathcal{G}_\mathcal{I}$. A node is created for each day and has the same weight as before, however we add an edge between two nodes if (the timestamps of) their corresponding documents have a distance $dist \leq \ell$ on the timeline (Fig. 3.4(c)). The two graphs $G_I$, $\mathcal{G}_S$ are equivalent and have the same MWC because in the former, two nodes (days) would be connected of they were at most $\ell/2 + \ell/2 = \ell$ apart, which is the case in $\mathcal{G}_S$ as well.

Therefore, both graphs can be seen as an ordered set of nodes $n_i, i \in [1, T]$, each of which has at most 2*$\ell$ edges, since each node can be connected with nodes of maximum distance $\ell$ before and after it (Fig. 3.4(c)).

In order to identify the maximum weight clique we can follow a simple reasoning: If MWC contains the first node $n_1$, then the clique consists of nodes $n_1, n_2, n_3..., n_\ell$. There is no point in having less than $\ell$ nodes in the clique since all scores are positive or zero, and we cannot have

more nodes, since node $n_1$ is not connected with node $n_{\ell+1}$. If the first node is not in the MWC, then in order for the second node to be in, the MWC would be composed by nodes $n_2, n_3..., n_{\ell+1}$. Similarly, if nodes $n_1, n_2, n_3..., n_{i-1}$ are not in MWC, then node $n_i$ could only be included if the MWC consisted of nodes $n_i$ to $n_{\ell+i}$. In other words, we only have to check for $T - \ell + 1$ possible maximum cliques. This is obviously equivalent with having an array with scores and finding the maximum interval of length $\ell$ (Fig. 3.4(d)).

## 3.5 Experimental Setup

For our experimental evaluation we used three real-world news datasets, each of them being a chronologically ordered sequence of documents. Table 3.1 describes each dataset in detail. Datasets $1, 2$ are parts of the New York Times[1] dataset, datasets $3, 4$ are articles from *The San Francisco Call* newspaper and datasets $5, 6, 7$ are articles from the website `Topix.com`, which host news articles from $181$ countries. After POS tagging and Word Filtering for all competing methods, we kept only nouns, verbs and adjectives. Specifically:

- **The *New York Times* dataset:** This dataset contains 1.8 million timestamped articles spanning a timeline of 20 years (7300 days) between Jan 1, 1987 and Jun 19, 2007, written and published by the New York Times. Multiple pieces of information are available, however we only used the timestamp and the content of each article. More specifically we used:

  1. **NYT10**: A sequence of 665,741 timestamped news documents spanning from Jan 01, 1987 to Dec 31, 1996, containing 1,036,204 distinct terms. Following [10] and to compare directly, we report results using this 10-year period.
  2. **NYT1987**: A sequence of 73,279 timestamped news documents spanning from Jan 01, 1987 to Dec 31, 1987, containing 277,000 distinct terms. We chose to present the results of this year in random because it was the first one in the dataset.

[1] `http://catalog.ldc.upenn.edu/LDC2008T19`

The results from different years were very similar and are not presented due to lack of space.

- **The *San Francisco Call* dataset:** This dataset consists of 297,701 chronologically ordered articles from *The San Francisco Call*, a daily newspaper with publication dates between 1903-1909. Several attributes for each article are available, however we only use the timestamp and the content fields. The dataset consists of two separate segments:

  1. **SF-Call1**: A sequence of 144,289 timestamped news documents spanning from Jan 01, 1903 to Dec 31, 1904, containing 115,000 distinct terms
  2. **SF-Call2**: A sequence of 153,412 timestamped news documents spanning from Jan 01, 1908 to Dec 31, 1909, containing 102,000 distinct terms.

- **The *Topix* dataset:** This dataset consists of 65,540 timestamped articles spanning a timeline of 365 days. The articles were colected from the website `Topix.com`, which host news articles from 181 countries around the world. Multiple pieces of information are available for each article, including the timestamp, the title, the content, and also the country of origin. This dataset spans a timeline from Jan 1, 2008 to Dec 31, 2008. More specifically we used:

  1. **TopixAll**: A sequence of 65,540 timestamped news documents spanning from Jan 1, 2008 to Dec 31, 2008, with 527,000 distinct terms.
  2. **TopixCanada** and **TopixSAfrica**: These two datasets consist of the subset of articles from the Topix dataset that originated from Canada and South Africa respectively. We chose Canada and South Africa since they are associated with more articles ($3,326$ and $2,389$) than any other country in the dataset. They contain 67,616 and 63,254 distinct terms respectively.

  Our motivation for focusing specific countries is to investigate the benefit of including spatial information in the document dating process.

After carefully examining the relevant literature on the document dating problem, we chose the algorithm proposed by Chambers [10] and a modification of the highly cited approach by Kanhabua and Nørvåg [28] as the competing methods for our experiments.

- `MaxEnt`: The algorithm proposed by by Chambers in [10] trains a discriminative version of a Maximum Entropy classifier. We used the `MaxEnt` classifiers from the freely available Stanford toolkit, leaving all settings to their default values (quadratic prior), as was done in the original paper.
- `NLLR`: The algorithm proposed by de Jong et al. [12] and extended by Kanhabua and Nørvåg [28] initially splits the timeline to segments of fixed (and equal) length (e.g. weeks). It then uses temporal language modeling to compare the vocabulary between each query document and the available segments, in order to choose the segment that is most likely to include the query document's true timestamp. Kanhabua and Nørvåg proposed some semantic-based preprocessing steps, among of which only **Part-Of-Speech** (POS) tagging and **Word Filtering** proved to be meaningful in our datasets. However the proposed use of external statistics, like Google Zeitgeist, is not feasible for old datasets. The precision values reported here are lower than in [28], due to the justified elimination of certain preprocessing steps as well as the usage of a different and larger document collection.

Following the notational convention in [10], by `NLLR` we refer to the De Jong et al. model [12]. The `NVA` abbreviation refers to the semantic-based preprocessing enhancement proposed by Kanhabua and Nørvåg, that includes POS tagging and Word Filtering. After the POS tagging, we kept only certain classes of words, namely Nouns, Verbs and Adjectives. This preprocessing process can be combined with all three methods, namely `BurstySimDater`, `NLLR` and `MaxEnt`. In the following, all precision results include the `NVA` preprocessing, unless stated otherwise.

In `BurstySimDater` experiments we used $k = 10$ most similar documents. Changing this parameter did not result in a big difference in

Table 3.1: Description of the datasets

| # | Dataset | Start Date | End Date | # docs |
|---|---|---|---|---|
| 1 | **NYT10** | 01/01/1987 | 12/31/1996 | 665,741 |
| 2 | **NYT1987** | 01/01/1987 | 12/31/1987 | 73,279 |
| 3 | **SF-Call1** | 01/01/1903 | 12/31/1904 | 144,289 |
| 4 | **SF-Call2** | 01/01/1908 | 12/31/1909 | 153,412 |
| 5 | **TopixAll** | 01/01/2008 | 12/31/2008 | 65,540 |
| 6 | **TopixCanada** | 01/01/2008 | 12/31/2008 | 3,326 |
| 7 | **TopixSAfrica** | 01/01/2008 | 12/31/2008 | 2,389 |

precision. In `MaxEnt` and `NLLR` we used all unigrams features. As is proposed in the respective papers [10, 28] and was validated in our experiments, performance of `MaxEnt` and `NLLR` is best when all features are used. All results in this section were computed with these parameters.

We evaluated all approaches on each of the available datasets via a 10-fold cross validation, omitting the known timestamp of a query document. In each of the 10 folds, $10$% of the dataset are used as queries, while the remaining $90$% serves as the reference corpus.

Precision is computed as the percentage of the query documents for which the actual timestamp was included in the timeframe reported by each of the algorithms. The reported results in the various comparison experiments are generated using the exact same training and testing sets for all approaches.

## 3.6 Scalability Experiments

In this experiment we applied the three methods on various random samples of increasing size from **NYT10**, which is the largest document collection. We experimented with various timeframe lengths $\ell = 1, 6, 12$ months. Figures **??**, **??** and **??** depict the total running time for each method as a function of the sample size. X-axis illustrates the total size of the dataset, $90$% of which serves as training- and $10$% as testing-set.

The total running time for NLLR includes partitioning of the dataset, indexing of the documents and building the language models for each partition. The total running time for MaxEnt includes the training of the Maximum Entropy Classifier. The total running time for BurstySimDater includes indexing of the documents and computation of the bursty intervals for all terms in the corpus. Moreover, all running time values include the computation of the reported intervals for all testing documents.

As depicted in Figures **??**, **??** and **??** BurstySimDater achieves the best precision for all dataset sample sizes and all timeframe lengths $\ell$. More importantly, in terms of total running time BurstySimDater scales much better than MaxEnt and is directly comparable to NLLR. Due to the computational complexity of MaxEnt some of the experiments did not terminate in a reasonable amount of time.

Figure 3.8 illustrates the difference in total running times of the three methods for a multiple query experiment. More specifically, for each document three intervals of respective lengths $\ell = 1, 6, 12$ months were desired. The reason for the depicted running time difference is that BurstySimDater algorithm does not need to repeat the indexing of documents and the computation of the bursty intervals in order to evaluate the three different queries, whereas NLLR and MaxEnt need to re-partition the timeline into segments of length $\ell = 1, 6$ and $12$ months. This is another benefit for not pre-partitioning the timeline into fixed-length segments.

## 3.7 Precision Experiments

In this experiment we evaluate and compare the precision values achieved by BurstySimDater, MaxEnt and NLLR in all datasets and settings. In order to measure the precision values as a function of (the length of the target timeframe) $\ell$, we experimented on **NYT10**, which is our largest dataset. Both MaxEnt and NLLR algorithms require a pre-segmented timeline in intervals of length $\ell$. Our BurstySimDater algorithm has no such requirement. Instead, $\ell$ is provided as an *upper bound* of the reported timeframe. We tune $\ell$ so that the results of

Table 3.2: Precision (%) for **NYT10** dataset

| **Timeframe Length** | NLLR | MaxEnt | BurstySimDater |
|---|---|---|---|
| **1 month** | 18 | - | **23.4** |
| **3 months** | 24 | - | **32** |
| **6 months** | 25 | 36 | **40** |
| **1 year** | 38.4 | 48.6 | **49.8** |

Table 3.3: Precision for **1 month** in **1 or 2 year(s)**

| **Dataset** | NLLR | MaxEnt | BurstySimDater |
|---|---|---|---|
| **NYT1987** | 29 | 24 | **32** |
| **SF-Call1** | 35 | **38.5** | **38.6** |
| **SF-Call2** | 29 | **36** | 34 |
| **TopixCanada** | 44.5 | 61.8 | **63** |
| **TopixSAfrica** | 75 | 81 | **84** |

the competing approaches are directly comparable. We evaluate the approaches for $\ell \in [4, 12, 24, 48]$ weeks.

Table 3.2 contains all achieved precision values for all methods. As described above, the experiments for MaxEnt algorithm did not terminate in reasonable time for target timeframes of length $\ell = 4$ and $12$ weeks. BurstySimDater not only outperforms the state of the art methods in all timeframe lengths, but also this difference in precision increases as the number of candidate time intervals becomes larger (Figures **??**, **??** and **??** for target timeframe lengths $\ell = 1, 6$ and $12$ month(s) respectively).

Table 3.3 depicts all precision values for $\ell = 1$ month for all $1$ or $2$ year datasets. This experiment also demonstrates the challenging nature of the problem: while some of the documents discuss specific events, others simply discuss topics that are not relevant to current events and can thus be associated with any timestamp.

BurstySimDater algorithm outperforms NLLR in all datasets for all values of $\ell$, while achieving similar values to MaxEnt, which in turn has the scalability problems analyzed in Section 3.6. Another interesting observation comes from the results on the **TopixCanada** and **TopixSAfrica** datasets. For this corpora, the achieved precision values

were significantly higher for all methods, reaching up to $63$% and $84$% respectively. This verifies our intuition that spatial information can be utilized to improve the results of our document-dating algorithm.

## 3.8 Conclusion

This chapter reviews the literature on the document timestamping problem and proposes a new approach for document dating that overcomes the drawbacks of previous methods: it doesn't depend on temporal linguistic constructs and it can report timeframes of arbitrary length. The proposed method outperforms the previous state-of-the-art in precision and computational efficiency in most of the cases, while being the most versatile of all, since it performs well for many reporting intervals and throughout a variety of datasets. This is achieved by taking into consideration the burstiness of the terms and lexical similarity of testing documents with the timestamped training corpus. An extensive experimental evaluation on real datasets demonstrated the efficacy of the algorithm and its advantage over the state of the art.

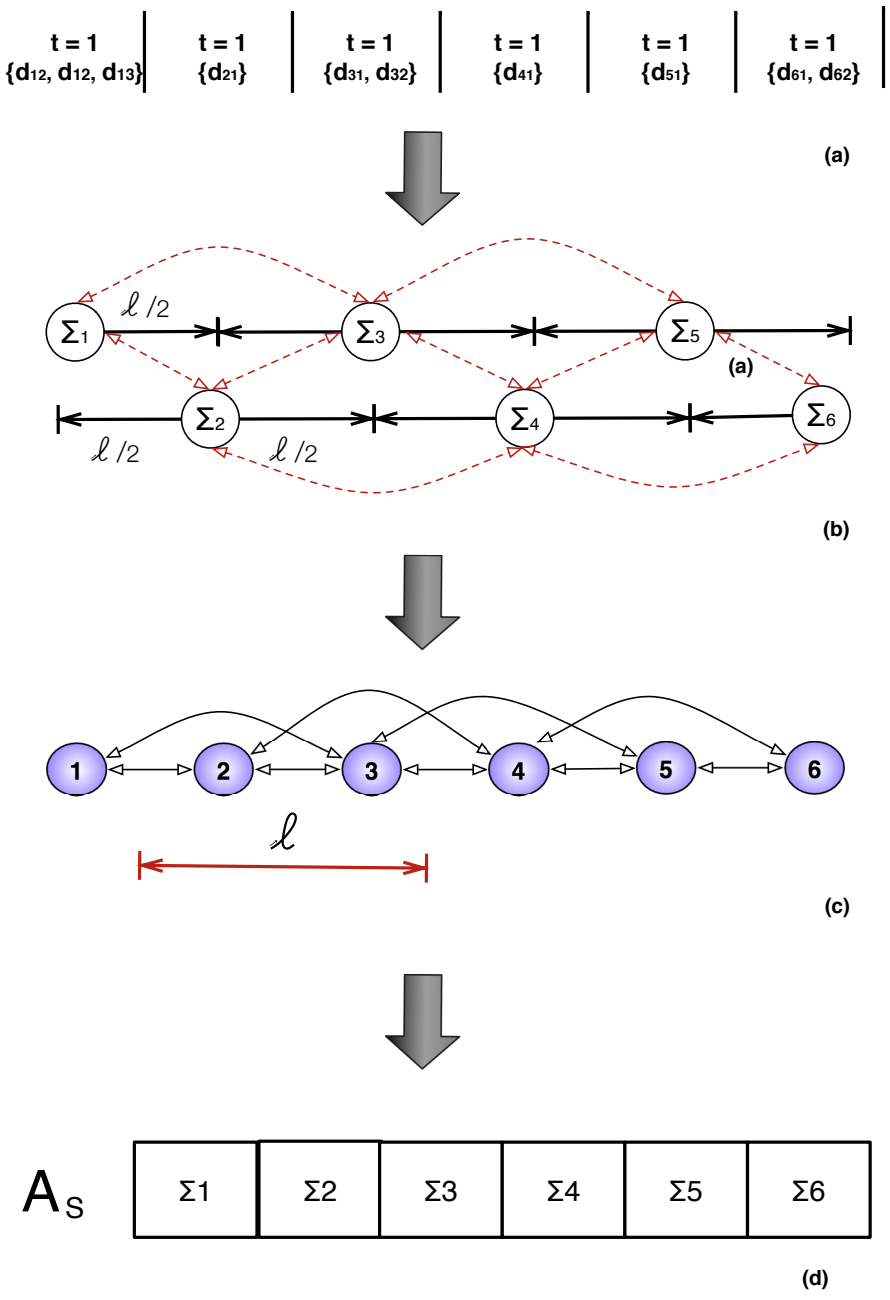


Figure 3.4: $\mathcal{G}_S$ can be reduced to $\mathcal{G}\prime_S$ which in turn can be reduced to array $A_S$ and the MWC is equivalent to a maximum sum window of length $\ell$

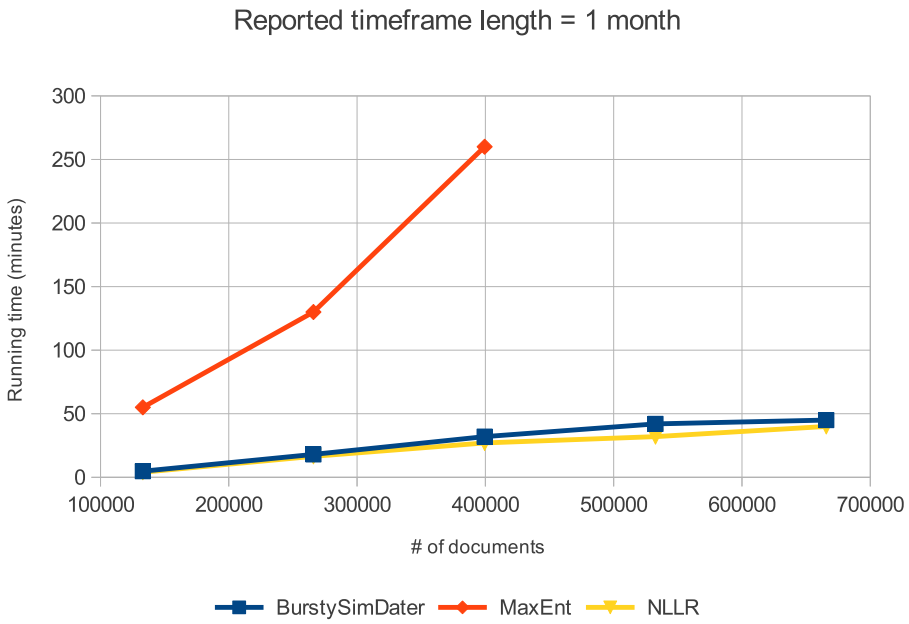


Figure 3.5: Comparison of total running time for the three methods vs. sample size for the **NYT10** dataset for target timeframe length = *1-month*.

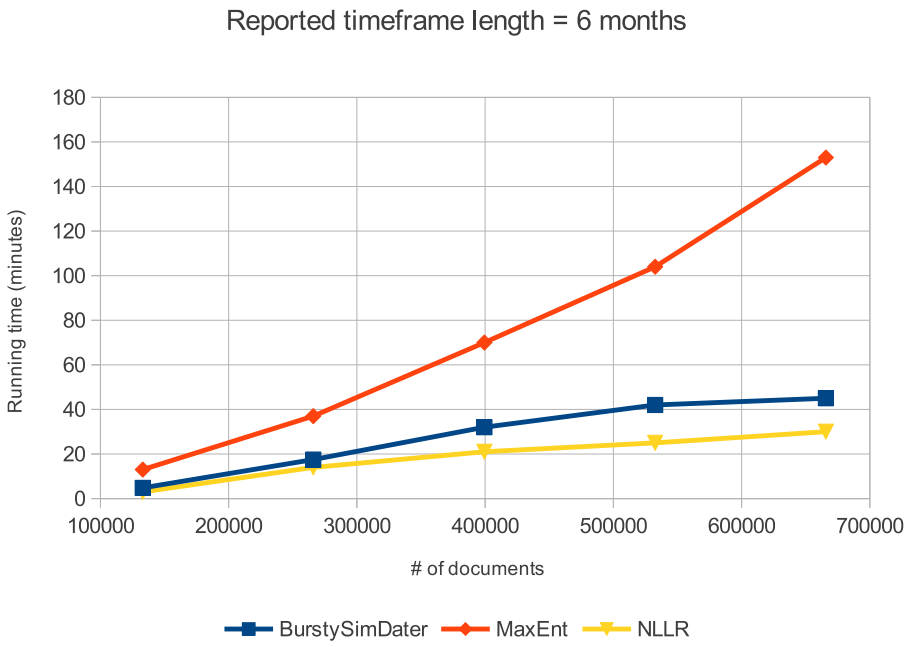


Figure 3.6: Comparison of total running time for the three methods vs. sample size for the **NYT10** dataset for target timeframe length = *6-months*.

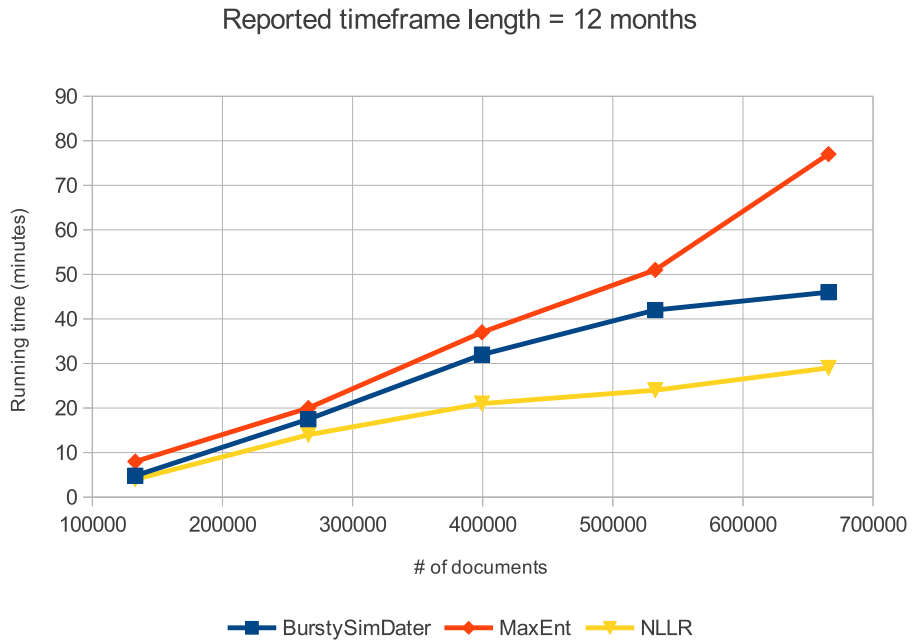


Figure 3.7: Comparison of total running time for the three methods vs. sample size for the **NYT10** dataset target timeframe length = *12-months*.

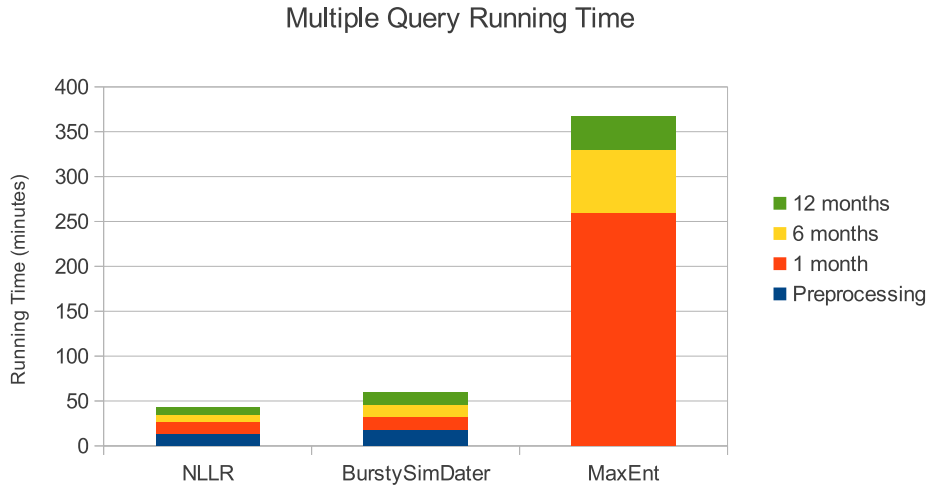


Figure 3.8: A multiple query experiment on a $60\%$ sample of the **NYT10** dataset yields the depicted total running time for the three approaches.

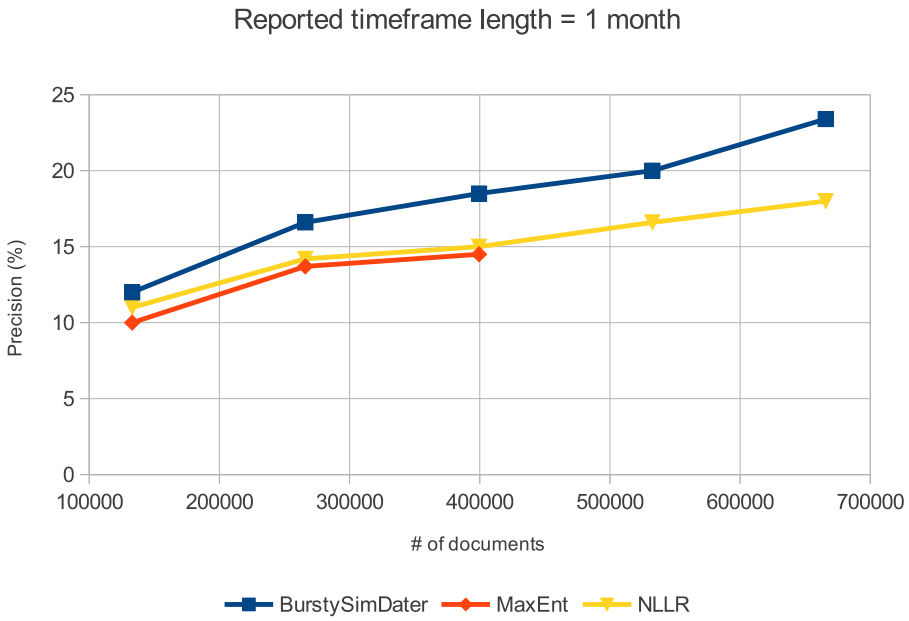


Figure 3.9: Comparison of precision values for the three methods vs. sample size for the **NYT10** dataset for target timeframe length = *1-month*.

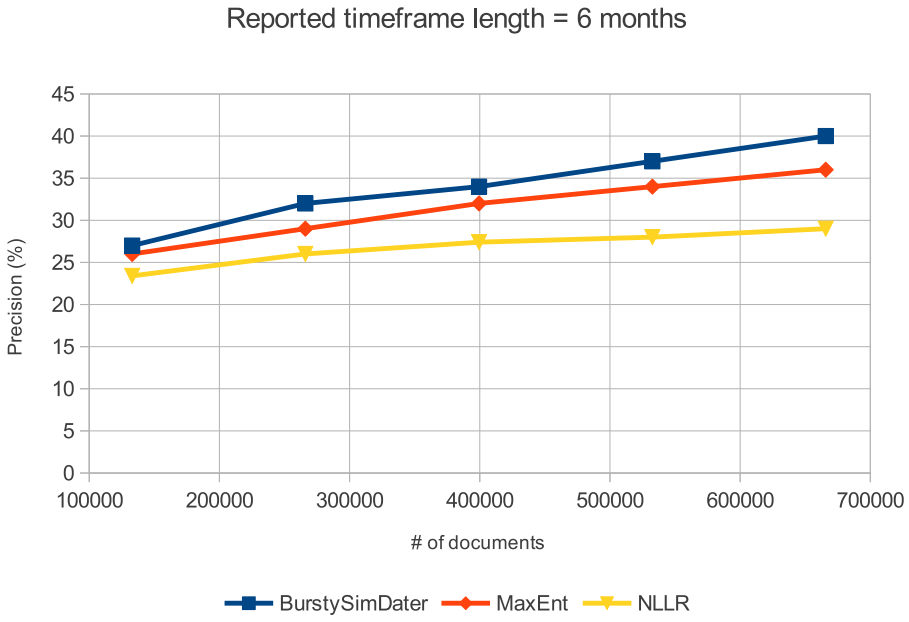


Figure 3.10: Comparison of precision values for the three methods vs. sample size for the **NYT10** dataset for target timeframe length = *6-months*.

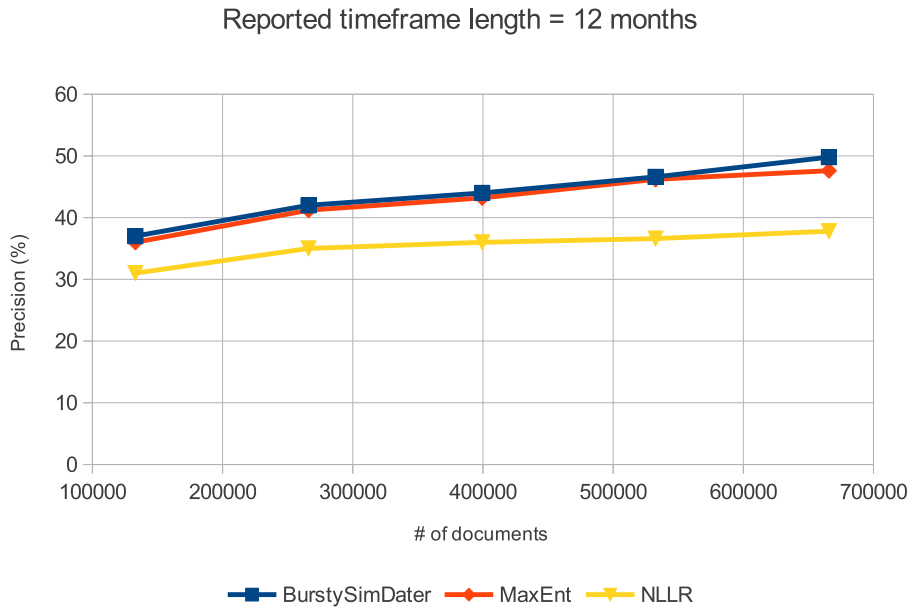


Figure 3.11: Comparison of precision values for the three methods vs. sample size for the **NYT10** dataset for target timeframe length = *12-months*.

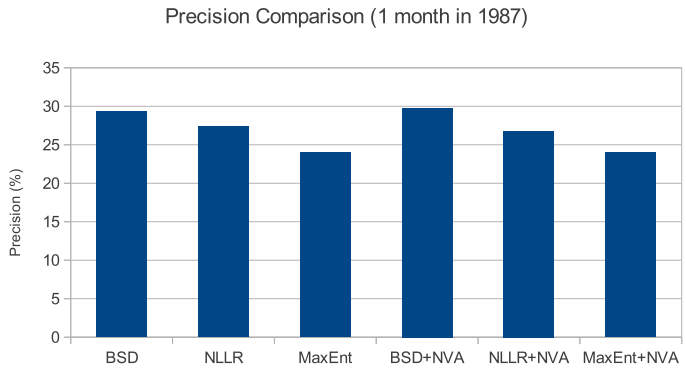


Figure 3.12: Comparison of the precision values between keeping all classes of words and keeping only Nouns, Verbs and Adjectives (NVA). Target timeframe length = 1 month, Year: 1987

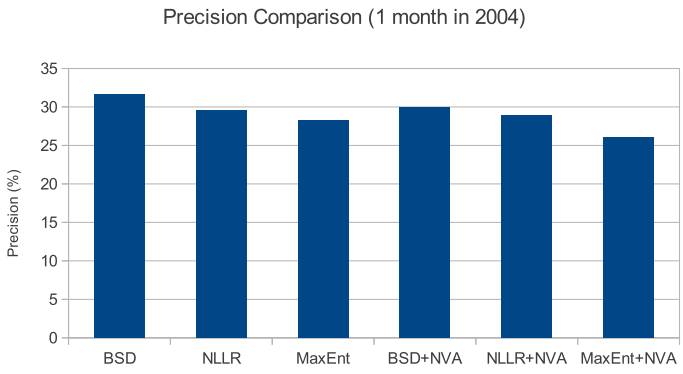


Figure 3.13: Comparison of the precision values between keeping all classes of words and keeping only Nouns, Verbs and Adjectives (NVA). Target timeframe length = 1 month, Year: 2004

# Chapter 4

# Language Agnostic Meme-Filtering in Document Streams

## 4.1 The Use of Hashtags in Social Networks

**hashtag:** a keyword that is marked with the hash # character.

Initially, hashtags were used only within Internet chat rooms. However, there seems to be a consensus on the origin of hashtags in social networks and most people attribute the proposal to use hashtags in order to annotate (mostly user-generated) content to Chris Messina, through a tweet dating back to *August 23, 2007* (Figure 4.1). Since then, users of Twitter.com social network (described in more detail in 4.2) have been using hashtags extensively as a way to categorize and annotate tweets and describe in a compact manner what the tweet (or the status update) is about, and thus, facilitate search and dissemination purposes. Twitter users have been using hashtags in an ever increasing frequency during many important events.

However, the first time a hashtag was extensively used and adopted by the public was during a fire in the city of San Diego on *October 23, 2007*, when a Twitter user named Nate Ritter used the social networking platform to report on the fire and included the hashtag #sandiegofire (Figure 4.2) [13].

Over the years, hashtags have been adopted by users of other social networking platforms as well (e.g. Facebook, Instagram, Pinter-


Figure 4.1: Chris Messina tweet on *August 23, 2007*, first hashtag ever: #barcamp


Figure 4.2: Nate Ritter tweet on *October 23, 2007*, first widely adopted hashtag: #sandiegofire

est, Flickr, Google+, etc. [49]) and have evolved to something more than just a way to annotate posts or add a narrative to status updates. Social media marketing companies and individuals have been using hashtags to create campaigns by finding new, innovative ways to use them in order to drive conversation, harness the public support, and garner attention to their brands.

## 4.2 Twitter streaming data

Twitter.com, being the platform where most events have a real-time representation of their evolution, offers a streaming Application Program Interface (API) and users are able to subscribe to the service in order to receive data. The default Twitter sampling service offers the subscribers the ability to download a random $1$% sample of all public tweets. Valkanas et al. compared the default $1$% Twitter sample to the Gardenhose sample, which returns $10$% of all public data, and evaluated their performance in a variety of applications [**?**].

In order to perform the memes and events analysis we developed crawlers for Twitter data, which can be configured to collect tweets published within a specific location (defined as a geographical bounding box), written by specific users or containing the desired keywords. Our crawlers interact with the Twitter API and store all available information for each downloaded piece of data, e.g. GPS location, number of followers/following users, contained urls, hashtags etc. As of the time of the writing of this dissertation, Twitter has $271$ million monthly active users (MAUs) and more than $500$ million tweets are posted per day. Even the $1$% sample of all public tweets provides a vast amount of data to analyze and extract knowledge from.

## 4.3 Memes and Events

On-line social networks analysis recently attracted attention from various scientific fields like Social Psychology [40], Political Science [16], Media and Communication [6], Marketing [8], Health Care [20], and, naturally, Computer Science [30]. In many cases, research on social data is interdisciplinary. This constantly raising interest is certainly expected, since social network data are easy to access and reflect multiple aspects of human behaviour and community dynamics. Probably the most well studied social network, is the Twitter micro-blogging platform.

From a data-science perspective, new mining tasks have recently appeared in micro-blog environments presenting interesting research

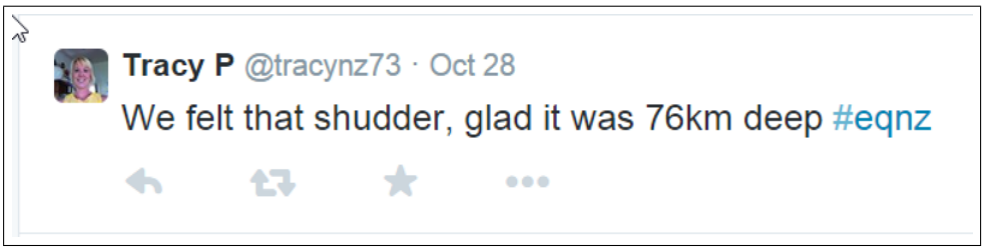


Figure 4.3: A Tweet that uses an Event hashtag to annotate content

challenges as well as commercial value. Sentiment Analysis [30], Event Recognition [46], Trend Identification [37], Community Recognition [39], Influence Propagation [17] are just a few characteristic examples.

*Tagging* thrives in Internet platforms with user-submitted content where tags are voluntarily assigned for information retrieval purposes: Users can do tag-based searches or browse objects of a particular tag. Tags are currently utilized in many different types of content such as, images (Flickr), videos (YouTube) and music (Last.fm). Twitter is a tag-rich service. Users annotate their posts by inserting keywords marked with the hash (#) character. These keywords are known as *hashtags* (see Figure 4.4).

Hashtags in Twitter are considered very important keywords since they add valuable meta-knowledge to a particular piece of text that is by nature limited to 140 characters. In order to track certain events and to annotate them properly users indirectly agree to hashtag Tweets with a predefined keyword (e.g. `#eqnz` - the hashtag citizens of New Zealand used to annotate content related to earthquakes - see Figure 4.3). Many micro-blog analysis tasks, like the ones mentioned in the previous paragraph, are exploiting tagging behaviour in multiple ways. Hence, hashtag quality plays an important role not only to information organization within Twitter but also to the efficiency of the state-of-the art tools for social network analysis.

Unfortunately in social media, users use hashtags not only to annotate specific events and topics but also to promote certain ideas or discussions known as *internet memes*. Many times Memes arise when a group of Celebrity fans try to promote a discussion topic related to their

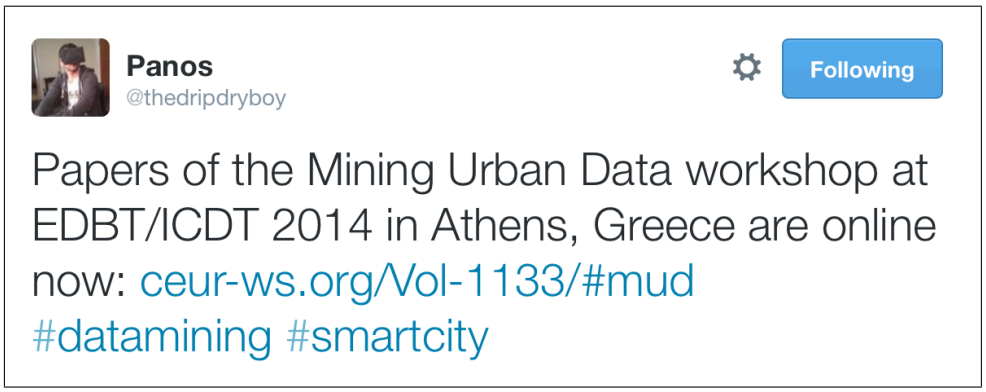


Figure 4.4: A Tweet that utilizes hashtags to annotate content


Figure 4.5: Hashtags used to promote celebrities.

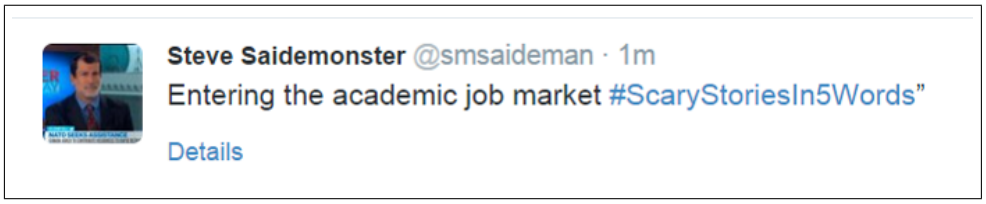


Figure 4.6: A hashtag used to promote a discussion.


Figure 4.7: A Tweet promoting a Meme

pop idol (Figure 4.7). Other types of Memes include internet hoaxes or marketing material. Memes are not inherently detrimental. However, since their data volume is many times significant, they can obstruct other tasks like trend or event detection. In these cases Memes are considered noise.

Social networking platforms can benefit from discriminating between different types of trending topics. For example by providing different landing pages or different advertising options for memes and events. Moreover, most event discovery approaches for social media are based on burst-detection mechanisms, assuming that a bursty behavior of a term or an $n$-gram may indicate that something important or extra-ordinary is happening in the world, and thus it is triggering popular discussions in social media. However, this is not always the case, as social media dynamics often lead to the creation of topics-of-interest that are internal to the network. As a result, most methods that attempt to discover events in social networking platforms do not take into account the fact that different types of trending topics - that is, topics whose popularity dramatically and unexpectedly increases - have different patterns of behavior in the network. The behavior of a topic can be characterized by a variety of factors, such as the community that is interested in the topic or the type of messages that are relevant to the topic and their characteristics, e.g. the number of hyperlinks to external sites, the number of attached pictures, the presence of hashtags, etc. In this chapter, we attempt to disambiguate between the different kinds of trends and reason about what characterizes them. Figure 4.8 illustrates an example of how both a meme and an event can reach the top *Trending Topics* list of platforms like Twitter.

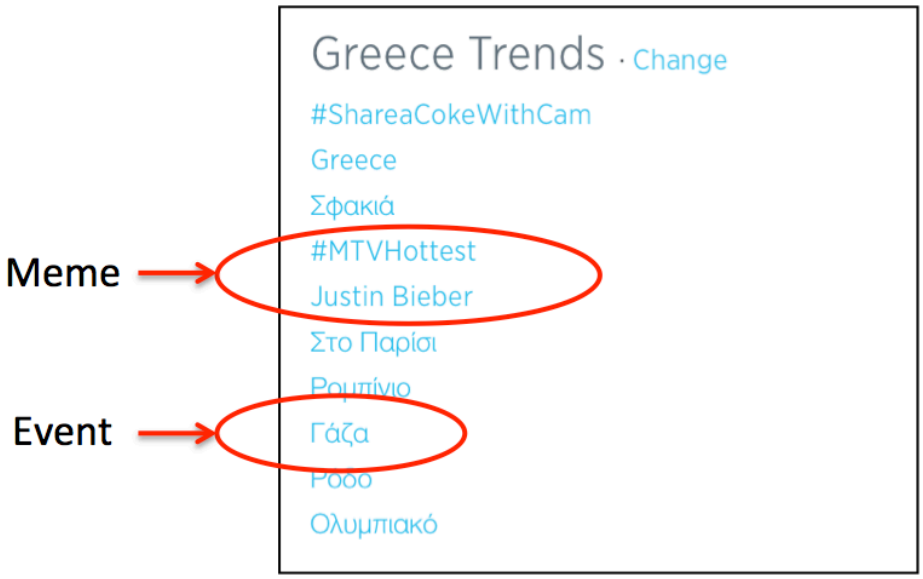


Figure 4.8: A real example of a meme and an event that appeared in the trending topics list for Greece on Twitter

Our contributions can be summarized in the following points:

- We provide a definition of *meme* and *event* and discriminate between them in social networks, recognizing the fact that not all trends behave in the same way.
- We propose a set of language-agnostic features to aid the classification of hashtags into *Event* or *Meme*. A variety of attributes is proposed and evaluated.
- We provide an extensive study of the behavior that characterizes *memes* and *events* and present an Gain-Ratio-based ranking of the proposed features in our setting.

The rest of this chapter, is structured as follows: In Section 4.4 we review the related work, in Sections 4.5 and 4.6 we define the problem and provide definitions of *Memes* and *Events* in the context of this chapter and in Section 4.7 we describe our approach. Section 4.8 contains our experimental evaluation, Section 4.9 describes a proof-of-concept meme-filtering method in terms of event detection and Section 4.10 concludes the chapter.

## 4.4 Related Work

In this section, we present recent and representative work that is related to the research challenges dealt in this chapter. More specifically, we discuss a) research efforts that study Meme phenomena, b) papers tackling the problem of trend and event detection, and c) studies on hashtag analysis.

### 4.4.1 Memes

Bauckhage [5] defines internet Memes as evolving content that rapidly gains popularity or notoriety on the Internet. Moreover, the author states that Memes are spread voluntarily rather than in a compulsory manner, which fact, although true, does not describe the full picture. Very often, Memes are produced by advertising or community campaigns, so they are expected to have different behavior to organically and not strategically created memes. For example, fans of groups or celebrities organize petitions in order to ask their idol to visit their country or say something about them. In this cases, the goal is to make a Meme so popular that it appears in the *Trending Topics* list of the platform, affecting the true meaning of the list. The related bibliography lacks methods of recognizing these campaigns. In this sense, we offer an initial approach towards this direction. Leskovec et al. define memes as ``short, distinctive phrases that travel relatively intact through on-line text'' [33]. They prove that information propagates from news sites to blogs. In their experiments there is an average lag of $2.5$ hours between peaks of attention in news sites and blogs. However, with the spread of social networks and microblogging platforms, like Twitter and Tumblr, this claim has to be re-examined. Kamath et al. study the spatio-temporal properties of online memes, by specifically limiting their research to the propagation of hashtags across Twitter, arguing that hashtags may associate statuses with particular events or with memes and conversations [25].

### 4.4.2 Trends and Events

In [50] the authors employ time series clustering in order to uncover temporal patterns in the popularity of content in social media and focus on the propagation of hashtags on Twitter. The authors (as in [33]) claim that mainstream media accounts (CNN, BBC, etc) produce content and push it to the other contributors, including twitter "first consume-then produce" accounts and professional bloggers. However, in [38] Petrovic et al. study the time aspect in Newswire and Twitter data and argue that Twitter covers most events that are mentioned by major news providers like CNN, BBC etc. Moreover it covers even smaller events that are not mentioned elsewhere. In their study they show that Twitter reports first sports events and unpredictable disaster-related events. In this sense, in the real time world, the highly credited news accounts are not always the ones that produce the *important content* first and the definition of what *important content* is still open. Regarding this problem, Petrovic et al. [38] use classes of content, i.e. *sports, politics, business, tv, etc.*. In [50] the authors use a time series shape similarity approach to find common temporal patterns and form clusters. They show that media agency news show a very rapid rise followed by a relatively slow decay. In [45] the authors try to predict the popularity of a hashtag in a given time frame using linear regression.

### 4.4.3 Hashtag Analysis

Many approaches in social network mining research have been devoted to the analysis of the role of tags and hashtags in social networking platforms. The hash symbol (`#') has been used to indicate the special meaning of a word or the union of multiple words and tag user-generated content in social networks like *Twitter, Instagram* or *Facebook*. Apart from tagging, social network users use hashtags for various other reasons, including search, annotations or starting viral conversations, often called *memes*. As opposed to traditional web search, queries in Twitter search that contain the hash symbol a significant portion of the total queries issued to the system [44]. Moreover, many Twitter queries reference words used in hashtags, but without the preceding `#' in the query. Since the amount of possible hashtags

a user can use to either tag content or search for results is essentially unlimited, both these tasks would benefit if users were aware of tags used by other users for the same or similar purposes [43]. Kamath et al. study the spatio-temporal dynamics of hashtags, proving that the spatial distance among locations affects the propagation of hashtags, although the latter are a global phenomenon [24]. Interestingly however, related work has not focused on the problem of distinguishing between *memes* and *events*.

## 4.5 Preliminaries

A proper formulation of the problem under study requires the definition of some basic concepts that in this setting can be rather abstract and ubiquitous. In the following, we define some basic concepts that we utilize in order to define the problem and propose a solution.

- **On-line Social Network (OSN)**: A web application in which *users* can: i) post content (text, video, images, etc), ii) connect and iii) interact with each other (follow, like, share, etc). This definition applies to well known social networks (e.g. Facebook, Twitter, LinkedIn, etc.) as well as content sharing communities (e.g. YouTube, Flickr, etc).
- **An account or profile ($p$):** An agent that can participate (i.e. perform *posts*) in a social network after following a registration procedure. Accounts can be operated by individuals, groups of people or computational agents (bots). Accounts usually maintain a *profile* in the OSN.
- **Content object ($c$):** A textual or media object that is published or shared via the social network (e.g. text, image, video). In contemporary social networks, content $c$ can be multi-modal including more than one form of content.
- **Social stream ($s$):** An infinite stream of content $c_i, c_{i+1}, \ldots$, where content $c_i$ could be created by users of a single or of multiple social networks.

- **News stream ($n$):** An infinite stream of news items $n_i, n_{i+1}, \ldots$, where news item $n_i$ can be generated from one or multiple online news sources (e.g. electronic editions of newspapers, magazines, etc).
- **Tag or Hashtag $h$:** A keyword that accounts use when creating content in order to semantically annotate it.
- **Relevant document:** A document $d_i$ is *relevant* to a topic $t_j$, if it contains the term describing $t_j$ in the raw text or in its meta-information fields (e.g. the attached hashtags in the case of Twitter, the categories in the case of blog posts, etc). We define the document-relevance function as $rd(d_i, t_j) = 1$ if $d_i$ is relevant to $t_j$, and $rd(d_i, t_j) = 0$ if not.
- **Relevant author:** An author $u_i$ is *relevant* to a topic $t_k$ if $a(d_i) = u_j, i \in [1, n]$ and $rd(d_i, t_k) = 1$. We define the author-relevance function as $ra(d_i, t_k) = 1$ if $a_i$ is relevant to $t_k$, and $ra(a_i, t_k) = 0$ if not.
- ***Event:*** In this work, an *Event* is represented by a topic $t_i \in \mathcal{T}$ and is a characteristic term that is present in the document stream, triggered by real-life circumstances or incidents that happened on, shortly before or shortly after the day of its appearance in the stream. *Events* examples include elections, football games, earthquakes, celebrations, etc.
- ***Meme:*** A *Meme* is represented by a topic $t_i \in \mathcal{T}$ and is a characteristic term that is present in the document stream, but has nothing to do with any real-life incident on or around the day of its appearance in the stream. *Memes* examples include keywords like *'WeWantJustinInAthens', '20ReasonsIAmCute', 'lovingit', etc.*

## 4.6 Problem Definition

Based on the above elementary concepts we define the difference between Events and Memes. *Both* Events *and* Memes in a Social Network drive users to create and publish content (text, images, etc)

in the social stream. Hence, Memes and Events can be observed in $s$ by identifying an excessive appearance of content related to this Meme or Event. The difference between an Event and a Meme is that an Event can be traced back in the news stream $n$ of the same period (as in $s$) whereas a Meme only appears in $s$.

An event could be identified by observing messages and discussions in the social stream regarding the recent Election in Germany, a soccer match between the teams Barcelona and Manchester United, an earthquake, or the Oscars ceremony. On the other hand Memes could be messages related to a celebrity fan group requesting their idol to to give a concert in their location, a discussion about why people cannot sleep at that time, etc.

In both cases, Memes and Events, as with any other content, are annotated with hashtags. For example, hashtags for the events described above could be: `'#GermanyElections'`, `'#BarcaVsMancester'`, `'#earthquake'`, `'#Oscars2014'`, whereas for the internet memes mentioned in the previous paragraph, example hashtags could be the following: `'#WeWantJustinInIreland'`, `'#20ReasonsIAmCute'`, `'#loveit'`, `'#insomnia'`. As we can observe, there are not any structural characteristics that can aid in separating an Event-hashtag from a Meme-hashtag. The discrimination solely depends on the context. In this work, we try to automatically build a model that distinguishes between the two.

**Problem 2 [Meme or Event Problem]:** *Given a limited part of the social stream $s_T \subset s$ (training set), build a model that can assign a label (`event`, `meme` to a hashtag $h$) given a specific set of information (statistics) for this hashtag*

The set of information mentioned in Problem 2, as we discuss later on, should be able to be calculated *incrementally* since this feature is crucial for data streaming environments. This information can be represented as a feature vector $\vec{h_x}$ consisting of a set of features. The requested model is actually a function $f(\vec{h_x}) \rightarrow \{event, label\}$, where $\vec{h}_x = \{g_1(s_T, h_x), \ldots, g_n(s_T, h_x)\}$ and $g_i$ ($i = 1, \ldots, n$) are the functions that incrementally can calculate the features $i$ for the hashtag $h$. Note

that with $h$ we represent the keyword expressing the hashtag whereas $\vec{h}$ is the feature representation and $n$ is the number of features. As we discuss later on, we formulate this problem as data classification problem by identifying training machine learning classifiers to learn the features that separate the two classes.

Given a set of $n = |\mathcal{S}|$ documents $\mathcal{S} = \{d_1, d_2, ..., d_n\}$, written by a set of users $\mathcal{U} = \{u_1, u_2, ...\}$ we extract a set of $m = |\mathcal{T}|$ specific topics of interest $\mathcal{T} = \{t_1, t_2, ..., t_m\}$, represented by single terms. In general, $n \gg m$. For each document $d_i$ in $\mathcal{S}$ there is a relation $a(d_i) = u_j$ that denotes the author of $d_i$. For each pair of user $u_j$ and $u_k$ in $\mathcal{U}$, $f(u_j, u_k) = 1$ if $u_j$ follows $u_k$ and $f(u_j, u_k) = 0$ if not. In the case of social networking platforms that only support bi-directional connections, $f(u_j, u_k) = f(u_k, u_j) = 1$ if $u_j$ and $u_j$ are friends and $f(u_j, u_k) = f(u_k, u_j) = 0$ if not.

Assuming that the topics $\mathcal{T}$ belong to a set of $l = |\mathcal{C}|$ predefined classes $\mathcal{C} = \{c_1, c_2, ..., c_l\}$, we aim to extract a set of features from the document collection $\mathcal{S}$ and the users collection $\mathcal{U}$ and train a classifier, so that it can distinguish between the classes in $\mathcal{C}$ with relatively high accuracy.
In this chapter, we focus on hashtags analysis in social networks, thus in the following *Memes* and *Events* will refer to hashtags and not individual words without the '#' attached to them.

## 4.7 Our Approach

We propose set of features that given a set of predefined classes (e.g. *meme, event, general, etc.*) and a manually labeled training set, can be used by a classifier with the goal of classifying the testing examples to the classes mentioned above. Specifically, the classification task can include the classification of hashtags, topics or keywords. In this work, we focused on the task of hashtag classification, although our work can be applied in the classification of each of the above mentioned types. Figure 4.9 illustrates the architecture of our approach. In this section we describe our steps in detail.

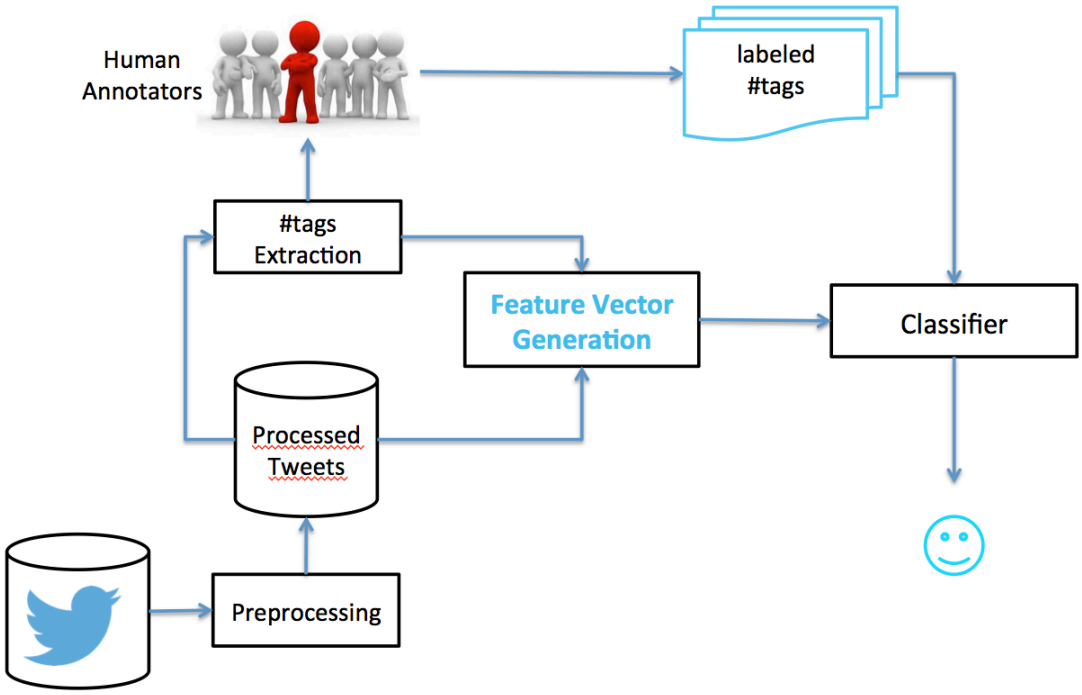


Figure 4.9: The architecture of our approach

### 4.7.1 Feature Set

The features that we computed and used in the classification experiments are described below, along with our intuition for computing them. We computed $15$ different features, resulting in $15$-dimensional vectors representing the topics (in this case hashtags) in the set $\mathcal{T}$. Some of the features are specific to the Twitter social network (e.g. retweets or favorites), however they can be applied on all social networks that support sharing or promoting content (e.g. *Share* or *Like* in *Facebook*). For each hashtag $t_i$, we take its *hashtagLength* in characters into account. Communities aiming to promote a meme or an phrase representing an advertising campaign often try to collapse a whole phrase into a single word in order to save characters for their messages. In this sense, often memes are longer than event-representing hashtags, because people in the social network try to embed in them as much information as possible. For example, not many English words are as long as *#WeWantOneDirectionInLondon*, which happened to appear as a trend for London sometime in March,

2013. For each hashtag $t_i$ we computed the following features:

#### 4.7.1.1 Document features

The following features are computed over the set of all documents that were *relevant* to hashtag $t_i$. We tried to capture the significance of rich content accompanying tweets, e.g. links, pictures, or videos. Within Twitter network, the hash sign ('#') has a special meaning, aiming to facilitate search for tweets related to a specific topic, e.g. *#WorldCup*. Previous work regarded all hashtag-based trends as memes, and although intuitively '#' is expected to appear more frequently in meme-related tweets than in event-reporting tweets, we aim to investigate further whether '#' is a strong indicator of a trend, topic or tweet being related to a network-generated meme or a real-life event. With the above intuition we compute the following features for each hashtag:

- *tokensPerTweet*: The average count of distinct tokens per relevant tweet
- *hashTagsPerTweet*: The average count of distinct hashtags per relevant tweet
- *urlsPerTweet*: The average count of included hyperlinks to external sites per relevant tweet
- *mediasPerTweet*: The average count of attached media objects per relevant tweet. Media objects can be photos, videos, songs. As of *December, 2014* Twitter supports only photos and videos.
- *favoritesPerTweet*: The average count of favorites per relevant tweet. Twitter offers the functionality of *favoriting* a tweet. This action serves as a means to either expressing approval or bookmarking a tweet for future reference. In the Twitter language, *favorite* or *fav* is the equivalent to *Like* in Facebook.
- *retweetsPerTweet*: The average count of retweets per relevant tweet. Twitter offers the functionality of re-posting a tweet, in order to express agreement with it. *Retweet* serves as a means for dissemination of popular content.

#### 4.7.1.2 Interaction features

With the social features we try to capture the importance of conversations about a topic in the social network. Twitter offers the ability to reply directly to a specific tweet or to mention other users (not necessarily friends or followers) inside a tweet, by adding the '@' sign before the name of the user to be mentioned. In our analysis the former is represented by *tweetsWithReplies*, which reflects the percentage of relevant tweets to hashtag $t_i$ that were replies to other tweets, while the latter is represented by *mentionsPerTweet*, which is the avg. number of mentions to other users over all tweets relevant to hashtag $t_i$.

#### 4.7.1.3 Community features

Memes are expected to come from clusters of users, whereas events are expected to interest a broader user base. These have to be defined and implemented.

- *statusesPerUser*: The *statusesPerUser* feature represents the avg. number of total posted status updates from the set of unique users that posted a tweet relevant to hashtag $t_i$. This features aims to catch the historical activity of the users community that was active with respect to the hashtag under consideration for the period it appeared in the top-$20$ list.

- *uniqueUsersCount*: This feature captures the size of the community that was interested in the corresponding hashtag.

- *userFollowersPerUser, userFriendsPerUser, listedCountPerUser*: These three features capture the popularity and the social activity of the users that appeared to be interested in the corresponding hashtag. They represent the avg. number of *Followers*, *Following* and *Lists* the users appeared in.

- *avgVerifiedUsers*: In order to have a measure of the credibility of the users interested in each hashtag, we utilize the *Verified* feature of the Twitter platform, and we compute the avg. number of users that are verified by Twitter.

## 4.8 Experiments

In this section we describe our dataset, our annotation process and our experimental evaluation.

### 4.8.1 Dataset Description

We crawled Twitter using the Twitter Streaming API for two different periods and two different bounding boxes. Specifically, we collected tweets from the bounding box of the United Kingdom for the period between *February 16, 2014* and *April 6, 2014* and tweets from the bounding box of Germany for the period between *April 1, 2014* and *October 10, 2014*. The two datasets contain 27 and 6 million tweets respectively. We split the datasets into days and computed features for the top-$20$ most popular hashtags for each day. Table 4.2 has more details about the datasets we crawled and used. Figs/memes **??** and **??** illustrate the distribution of the hashtags across the dataset. It is apparent that most hashtags are used only once, which indicates *(i)* that the users use them for annotating their content and thus facilitate search, and *(ii)* that the users are not aware about which hashtags are used by other people in the network at the moment of the creation of the content. In fact, Figure **??** shows that most hashtags in United Kingdom appear less than $80$ times in the period of study, which spans $50$ days. Figure **??** diplays the wordcloud of the top-$100$ most popular hashtags in our dataset. It is apparent that even the set with the top-$100$ most popular hashtags contains both *Memes* (e.g. `#georgesnapchatme`, `#100happydays`, etc.) and *Events* (e.g. `#brits2014`, `#bbcqt` tagging tweets about the BRIT Awards 2014 and the 'BBC Question Time' television program respectively, etc.).

Before annotating the extracted hashtags, we performed some preprocessing steps. Specifically:

- We lowercased all hashtags, in order to collapse to one entity hashtags representing the same thing but written in different ways, e.g. `#WeWantJustinInIreland` and `#wewantjustininireland`
- We filtered out location names, e.g. `#London`, `#Dublin`, `#Berlin`,

Table 4.1: Occurrence Counts for very popular hashtags

| hashtag | Dataset | Count |
|---|---|---|
| #nowplaying | Germany | $96,261$ |
| #ger | Germany | $64,430$ |
| #berlin | Germany | $51,561$ |
| #nowplaying | United Kingdom | $43,478$ |
| #london | United Kingdom | $35,837$ |

Table 4.2: Dataset Statistics

| Description | UK | Germany |
|---|---|---|
| Unique Tweets | 27,868,183 | 6,826,709 |
| Tweets with at least one hashtag | 4,432,052 | 950,739 |
| Unique Users | 721,644 | 237,344 |
| Unique Hashtags | 1,102,320 | 491,043 |
| Hashtags appearing only once | 806,160 (73.1%) | 246,311 (66.6%) |
| Average occurrences per hashtag | 6.1 | 7.47 |

#Frankfurt etc. In the case of significant events, e.g. an earthquake, the event would show up in other popular hashtags too. In all other cases location hashtags are vague with respect to whether they represent a meme or an event. Table 4.1 lists the occurrence counts for the most popular hashtags in our datasets. In comparison, as shown in Table 4.2, the average number of occurrences per hashtag was $6.1$ in *United Kingdom* and $7.47$ in *Germany*.

- We filtered out hashtags obtained from messages posted by automated systems like Spotify, Facebook, Instagram or bot accounts, e.g. #nowplaying, #ukweather, #trdnl, etc.

- We filtered out day and month names, e.g. #friday, #sunday, #january etc.


Figure 4.10: The wordcloud of top-$100$ most popular tags in *United Kingdom*

### 4.8.2 Annotation Process

After the initial preprocessing we asked independent people to manually tag all remaining preprocessed hashtags into one of the two classes, *meme* and *event*, while there wasn't any option not to tag an example. The annotators were not exposed to the feature vectors that corresponded to the hashtag examples, in order to avoid bias towards any of the classes. The number of the independent annotators was $5$ for the *United Kingdom* dataset and $3$ for the *Germany* dataset. Afterwards, we used majority voting in order to specify the class of each hashtag. In the *United Kingdom* dataset we ended up having $1100$ tagged examples and vectors, among of which $558$ where tagged as *events* and $542$ as *memes*. In the *Germany* dataset we ended up having $800$ tagged examples and vectors, among of which $358$ where tagged as *events* and $442$ as *memes*. The agreement among the two sets of annotators for the *United Kingdom* and *Germany* datasets is illustrated in Tables 4.3 and 4.4 respectively.


Figure 4.11: The wordcloud of top-$100$ most popular tags in *Germany*

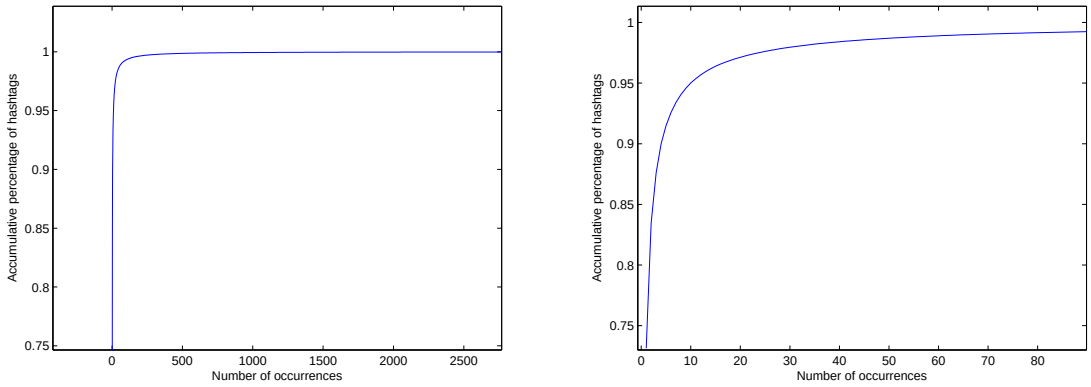


Figure 4.12: Cumulative distribution function of unique hashtags over number of occurrences (total and zoomed-in)

In order to quantitatively measure the annotators' agreement in the labeling process we used the Fleiss' kappa statistic. Fleiss' kappa is an extension of the well known Cohen's kappa, which is a measure of the agreement between two raters, where agreement due to chance

Table 4.3: Annotator Agreement for the *United Kingdom* dataset

| Majority | Meme | Event |
|---|---|---|
| 3 to 2 | 29.5% | 44% |
| 4 to 1 | 65.4% | 51.6% |
| 5 to 0 | 5.1% | 4.4% |

Table 4.4: Annotator Agreement for the *Germany* dataset

| Majority | Meme | Event |
|---|---|---|
| 2 to 1 | 36.5% | 68.5% |
| 3 to 0 | 63.5% | 31.5% |

is factored out. In our case, for both datasets the number of raters was more than two, so Fleiss' extension was used. For the *Germany* dataset the annotators agreed with $\kappa = 0.301$, whereas for the *United Kingdom* dataset the corresponding value was $\kappa = 0.202$.

Whereas in the case of the *Germany* dataset the $kappa$ value constitutes for a fair agreement among the annotators, when the number of the annotators increases, as is the case with the *United Kingdom* dataset, $kappa$ decreases. This can be attributed to the following fact: Since Twitter users in United Kingdom are more active, there is a larger diversity of the top hashtags, which makes it more difficult for the annotators to reason on whether a hashtag represents an actual event or a social network generated meme. On the other hand, in Germany, the distinction of the two classes is clearer, since Twitter users in this area tend to post updates about a more narrow variety of topics, including mainly football- and celebrity-related tweets. Thus, most hashtags that belong to the former category are characterized by the annotators as *events*, since a real football match is held, shortly before or shortly after the time of the posting. Hashtags that belong to the latter category are annotated as *memes*, since most of the time nothing important has happened concerning the respective celebrity.

**Discussion.** In Germany most events are related to soccer. On the other hand, most memes are about celebrity and television. Memes thrive during the weekends. In both datasets, the most popular hashtags included tags obtained from automated messages for weather, running and music playing + locations (`berlin, frankfurt, london`). These hashtags were exluded from the annotation and testing process, since they don't contain important information about significant events or memes. Moreover, in the *Germany* dataset, we observed significantly less diversity in the usage of hashtags, which can be seen in Figure 4.13.

In our own evaluation, we computed the Jaccard similarity of the top-$20$ hashtags between consecutive days for a whole month in the two datasets and compared the resulting values. The Jaccard similarity between two sets is computed as the quotient of the overlap and the union of their respective vocabularies. Formally, given the set of top hashtags $th_i$ corresponding to day $i$ and the set of hashtags $th_{i+1}$ corresponding to day $i+1$:

$$Jaccard(th_i, th_{i+1}) = \frac{|th_i \cap th_{i+1}|}{|th_1 \cup th_{i+1}|} \tag{4.1}$$

As Fig. 4.13 illustrates, the Jaccard coefficient of the top-$20$ hashtags in Germany is quite hiqh when comparing the sets day over day during a whole month, which indicates that the most frequently used hashtags are more or less the same every weekday with the exception of football and Champions League days. As a result, the average Jaccard Coefficient has a value of $0.48$. In comparison, in United Kingdom, the top hashtags change quite significantly as time goes by, resulting to an average Jaccard Coefficient of $0.21$.

### 4.8.3 Classifiers

We experimented with four traditional general purpose classifiers offered by the Weka tool [19]. Specifically, we chose the Naive Bayes, Random Forest, Support Vector Machines (SVM) and $k$-Nearest Neighbor classifiers (k-NN) [2]. Naive Bayes assumes that the features are conditionally independent, which is not true for all the fea-

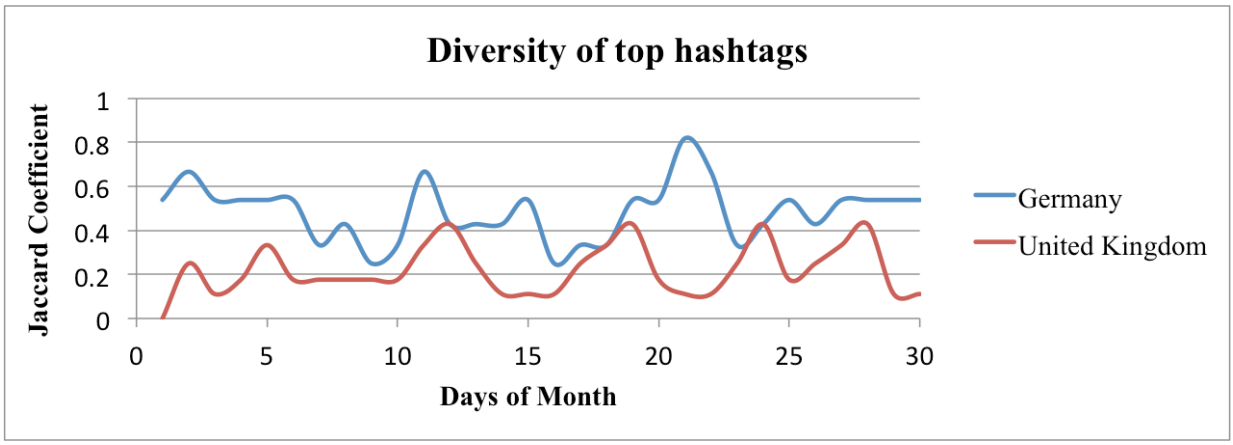


Figure 4.13: Jaccard Coefficient of top-$20$ hashtags as they appeared in *United Kingdom* and *Germany* during *March, 2014* and *May, 2014* respectively

tures we used [23]. However, it has been shown that it is effective in practice without the unrealistic independence assumption. The Random Forest classifier is effective in giving estimates of what variables are important in the classification, thus providing a ranking of the features in terms of importance [7]. The Support Vector Machine implementation we chose was the Sequential Minimal Optimization (SMO) algorithm [34], which trains a support vector machine with polynomial or RBF kernels.

The Random Forest classifier was able to reach an accuracy of 89.2%, with an average precision and recall of 89.2%. The confusion matrices of all classifiers for the United Kingdom and Germany datasets are illustrated in Tables 4.5 and 4.6 respectively. Figure 4.14 illustrates how the four classifiers we used compare against each other in terms of accuracy as a function of the size of the training set. Random Forest classifier has been more accurate than the other classifiers for all values of the size of the training set. Figures 4.15 and 4.16 illustrate the achieved accuracy values of the four classifiers when using a 10-fold cross-validation scheme for the *United Kingdom* and the *Germany* datasets respectively. In bot datasets, Random Forest outperforms Naive Bayes, $k$-NN and SVM, reaching an accuracy of $89$%.

Table 4.5: Confusion Matrix of the four classifiers for the *United Kingdom* dataset (M=Meme, E=Event)

| | Prediction | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Naive Bayes** | | **Random Forest** | | **SVM** | | **k-NN** | |
| True | M | E | M | E | M | E | M | E |
| M | 0.64 | 0.36 | 0.91 | 0.09 | 0.73 | 0.27 | 0.87 | 0.13 |
| E | 0.08 | 0.92 | 0.11 | 0.89 | 0.13 | 0.87 | 0.14 | 0.86 |

Table 4.6: Confusion Matrix of the four classifiers for the *Germany* dataset (M=Meme, E=Event)

| | Prediction | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **Naive Bayes** | | **Random Forest** | | **SVM** | | **k-NN** | |
| True | M | E | M | E | M | E | M | E |
| M | 0.77 | 0.23 | 0.90 | 0.10 | 0.79 | 0.21 | 0.85 | 0.15 |
| E | 0.11 | 0.89 | 0.13 | 0.87 | 0.13 | 0.87 | 0.13 | 0.87 |

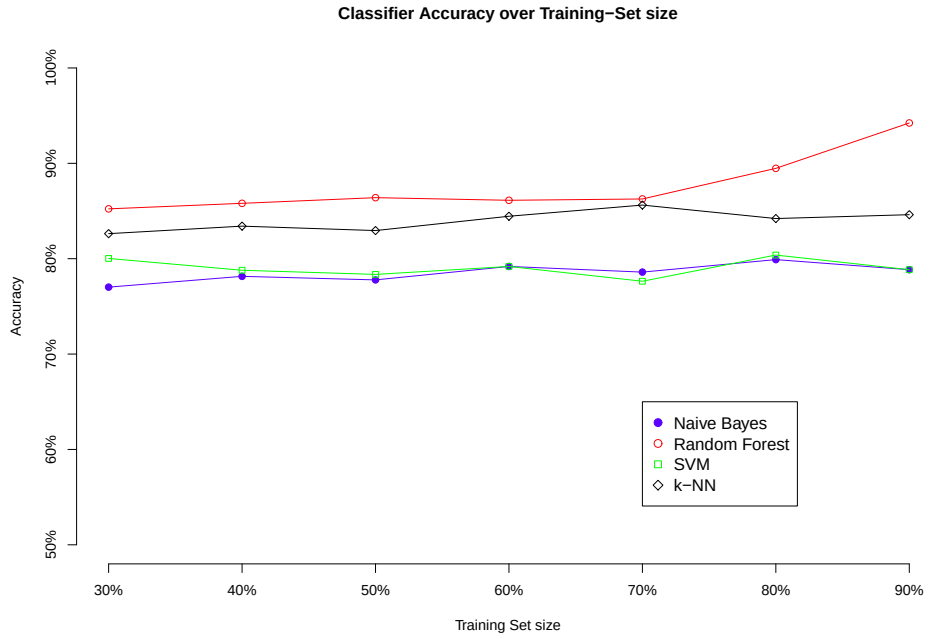


Figure 4.14: Accuracy of Naive Bayes, Random Forest, SVM and k-NN classifiers as a function of training set size

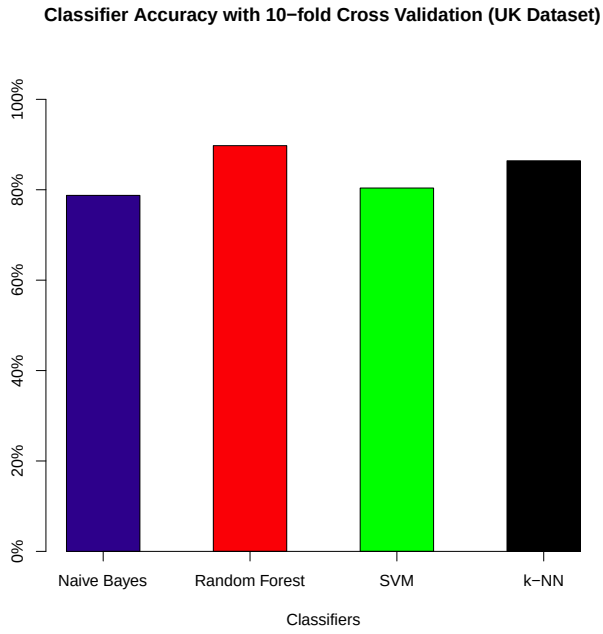


Figure 4.15: Accuracy of Naive Bayes, Random Forest, SVM and k-NN classifiers with 10-fold cross-validation for the *United Kingdom* dataset

### 4.8.4 Feature Selection

In order to argue about which features are the most important for the classification of hashtags we ranked them in decreasing Gain Ratio with respect to the two classes. Table 4.7 lists the features according to this ranking. We then repeated the classification process with the four classifiers, starting with the first feature in Table 4.7 and incrementally adding the remaining features one by one, in order to inspect the benefits in classification accuracy. Figure 4.17 depicts the results of this experiment. Here again, the Random Forest classifier outperforms all others for all feature subsets. Interestingly, when we used only the *community* features, the Random Forest classifier was able to reach an accuracy of 70.8%, while when we used only the *document* features the classifier reached an accuracy of 86%.

In order to further investigate relationships between individual features that serve as indicators and the hashtag classes we study the Figs/memes 9-14. By looking carefully at Figure 4.21 we can identify a tendency indicating that users who follow few others but have many

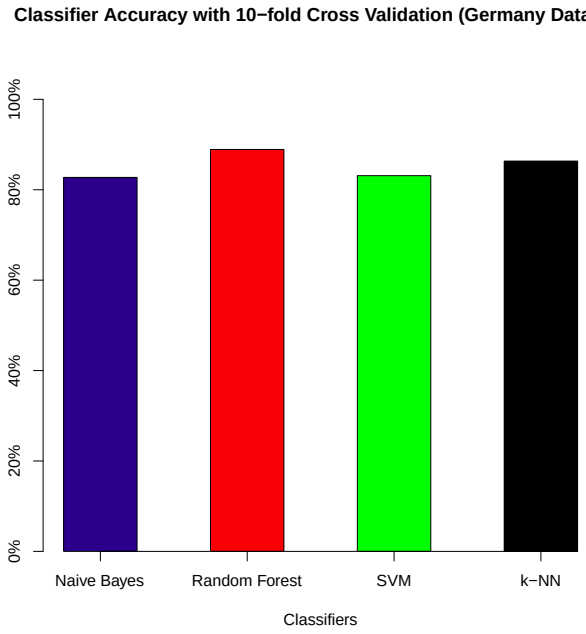


Figure 4.16: Accuracy of Naive Bayes, Random Forest, SVM and k-NN classifiers with 10-fold cross-validation for the *Germany* dataset

followers themselves, tend to mostly write about events, while Figure 4.18 shows that *Events* are being discussed by more unique and less active users than *Memes*. On the other hand, in the same Figure, it is apparent that the average number of the unique relevant users to hashtags classified as *Memes* is not large, while these users tend to be very active in the network, having posted far more tweets than users writing about *Events*. Similar conclusions can be derived from the second part of Figure 4.18, where users posting content about *Events* tend to be included in considerably more lists than users promoting or contributing to *Memes*.

Figs/memes 4.19 and 4.20 reveal a number of expected, yet interesting observations:

- Tweets that contribute to propagation or promotion of *Memes* have significantly more videos, photos or hashtags attached to them than tweets discussing real-life *Events*. Memes often are parts of campaigns or internet petitions and users try to enrich the content they generate so it ranks higher in search results, ei-

ther for a specific hashtag or for a relevant topic. Having more hashtags in the tweet increases the chances of it becoming viral or including a hashtag other users search for.

- Tweets that are relevant to *Memes* draw more conversations in the social network than tweets that report a real-life *Event*. This is to be expected, since, as described above, the number of unique users who are interested in memes is relatively small and thus communities with similar meme-oriented interests are more easily formed. Such communities consist of people interested in celebrities, jokes, etc.
- Tweets discussing *Events* have on average slightly more tokens. This is normal, since these kind of tweets have a less arbitrary structure as they often include quotes or headlines in order to reproduce news reports, thus more words are needed to express something news-worthy.

Interestingly enough, Figure 4.23 reveals a rather odd observation. While tweets about breaking and significant events were expected to contain a relatively high number of URLs linking to external sites with the source of the information, this appears not to be true. In the Figure, there is a clear separation of the spaces covered by *Memes*-examples and *Events*-examples, showing that *Memes* are represented by tweets with fewer tokens - as described above - and more URLs, whereas *Events*-related tweets contain on average and on aggregate much fewer URLs and more tokens.

## 4.9 Hashtag-Based Event Detection: A Proof of Concept Use Case of Meme-Filtering

In order to show the utility of the proposed methodology we applied a hashtag-based event detection approach to our data. As mentioned in the introduction, due to the limited text length of tweets, hashtag analysis is a common approach for micro-blogs mining. For example, most event detection methods in social media rely on time-series analysis of hashtags, inspecting terms that appear bursty for specific

Table 4.7: Decreasing Gain Ratio Feature Ranking for *United Kingdom* and *Germany* datasets

| **United Kingdom** | | **Germany** | |
|---|---|---|---|
| **Feature** | **Gain Ratio** | **Feature** | **Gain Ratio** |
| tweetsPerUser | 0.1617 | mentionsPerTweet | 0.1764 |
| tweetsWithReplies | 0.1432 | tweetsPerUser | 0.1566 |
| userStatusesPerUser | 0.1181 | avgVerifiedUsers | 0.1547 |
| tweetsWithUrl | 0.0958 | tweetsWithReplies | 0.1451 |
| urlsPerTweet | 0.091 | hashTagsPerTweet | 0.1422 |
| tokensPerTweet | 0.0822 | tweetsWithUrl | 0.1416 |
| mentionsPerTweet | 0.0822 | urlsPerTweet | 0.1416 |
| userFriendsPerUser | 0.0802 | listedUsersPerUser | 0.1383 |
| mediasPerTweet | 0.0778 | uniqueUsersCount | 0.1294 |
| uniqueUsersCount | 0.0574 | mediasPerTweet | 0.1244 |
| hashtagLength | 0.0572 | tokensPerTweet | 0.1127 |
| hashTagsPerTweet | 0.0527 | userFriendsPerUser | 0.0959 |
| avgVerifiedUsers | 0.0461 | userFollowersPerUser | 0.0883 |
| userFollowersPerUser | 0.0355 | userStatusesPerUser | 0.0728 |

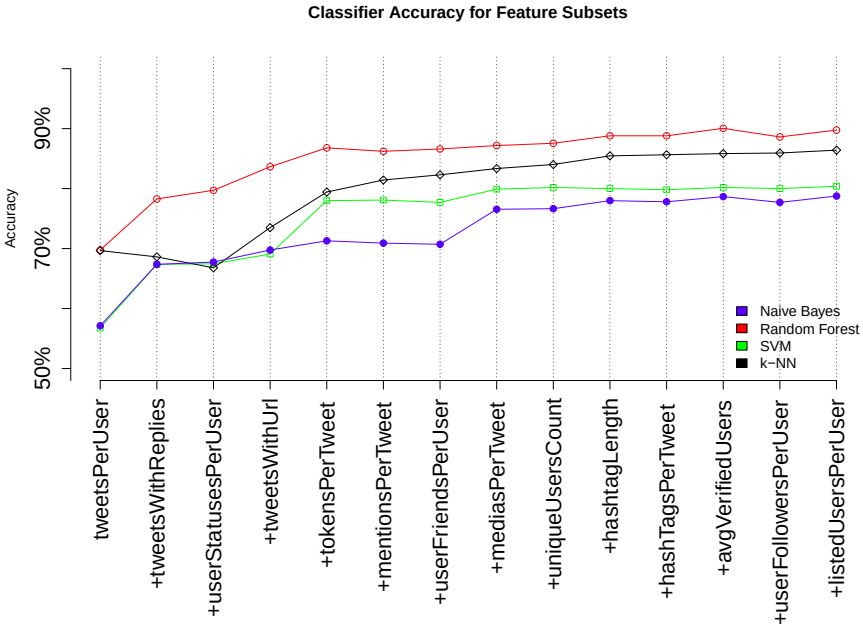


Figure 4.17: Accuracy of the four classifiers with different feature subsets, incrementally adding the next feature w.r.t to Gain Ratio

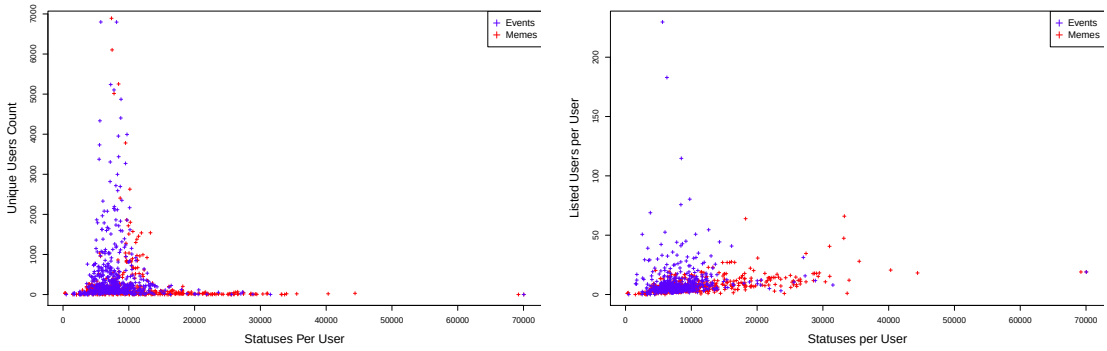


Figure 4.18: Relationship of the unique relevant users and the lists the users belong to with the avg. number of Twitter statuses per relevant user

time-periods or generally popular terms. The assumption is that the bursty keywords/hashtags will be related to emerging events.

In this section, we argue that these event detection methods can lead to mixed results, since memes are also popular and bursty. In fact, memes appear to have a very well defined popularity period, just like

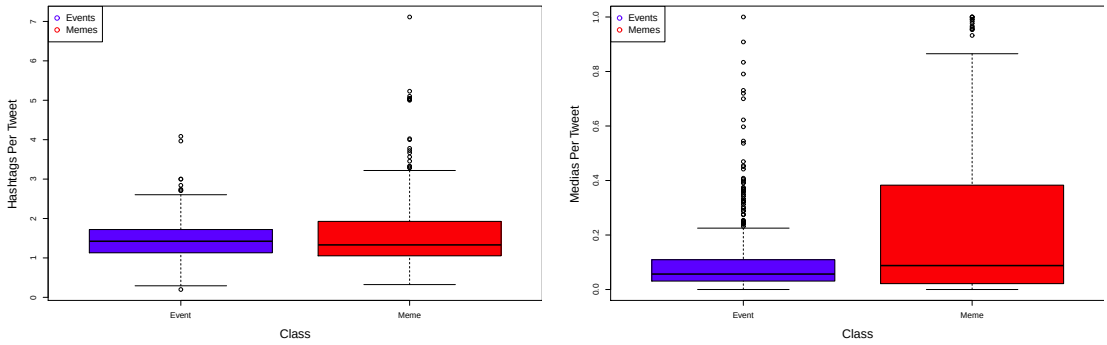


Figure 4.19: Boxplots for the avg. number of hashtags and media entities per relevant tweet against the two classes

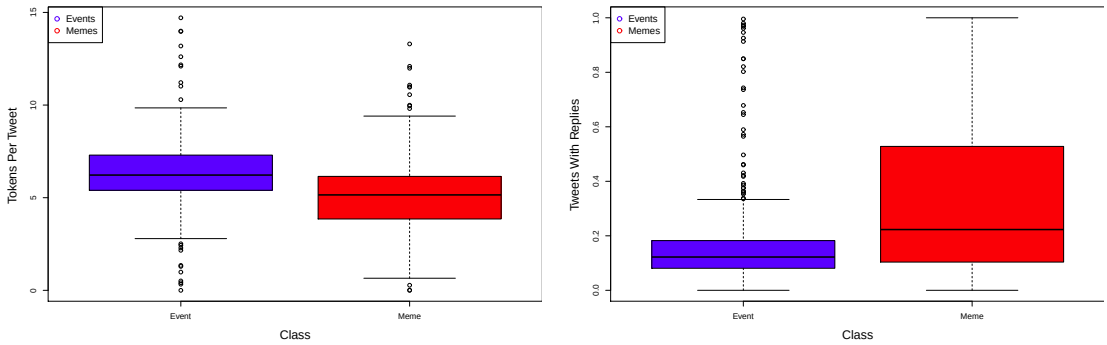


Figure 4.20: Boxplots for the avg. number of tokens and replies per relevant tweet against the two classes

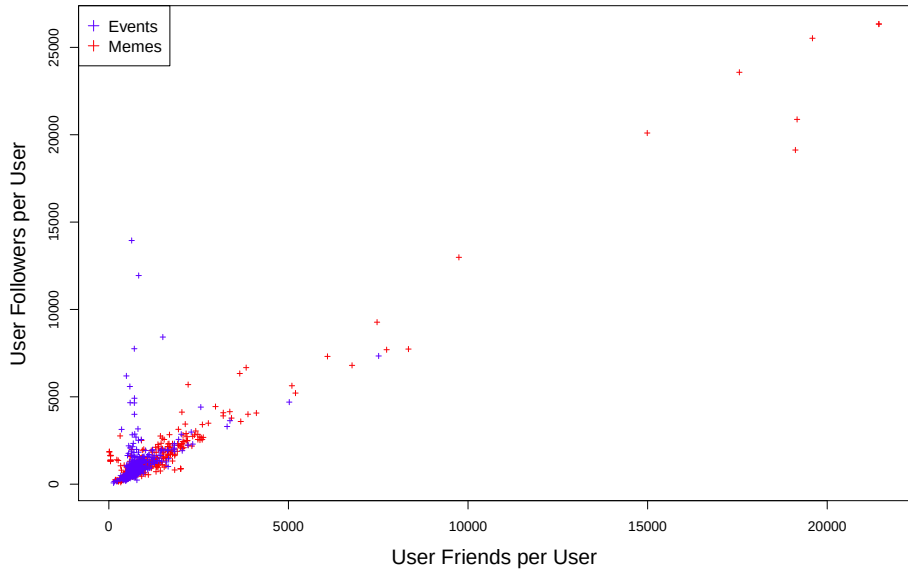


Figure 4.21: Relationship of the avg. number of friends with avg. number of followers of relevant users

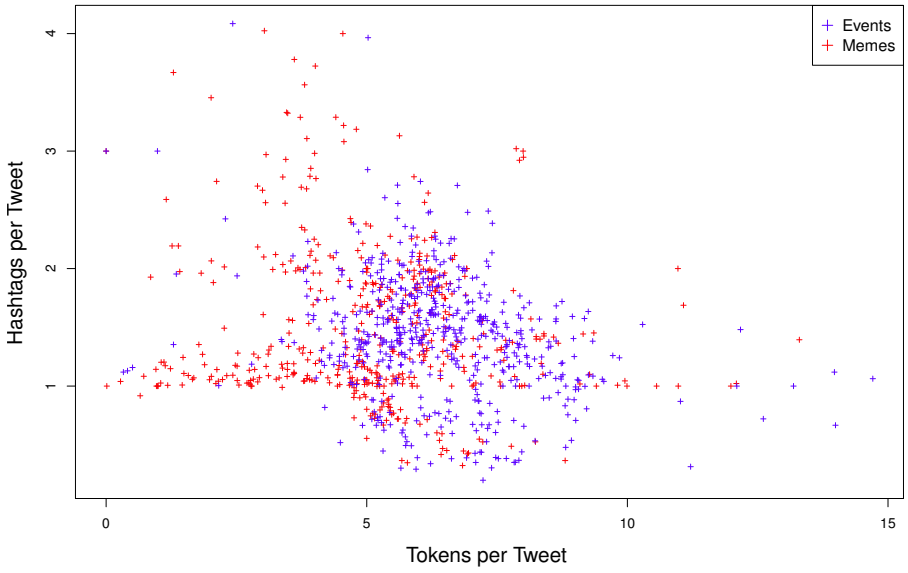


Figure 4.22: Relationship of avg. number of hashtags per relevant tweet with the avg. number of tokens per relevant tweet

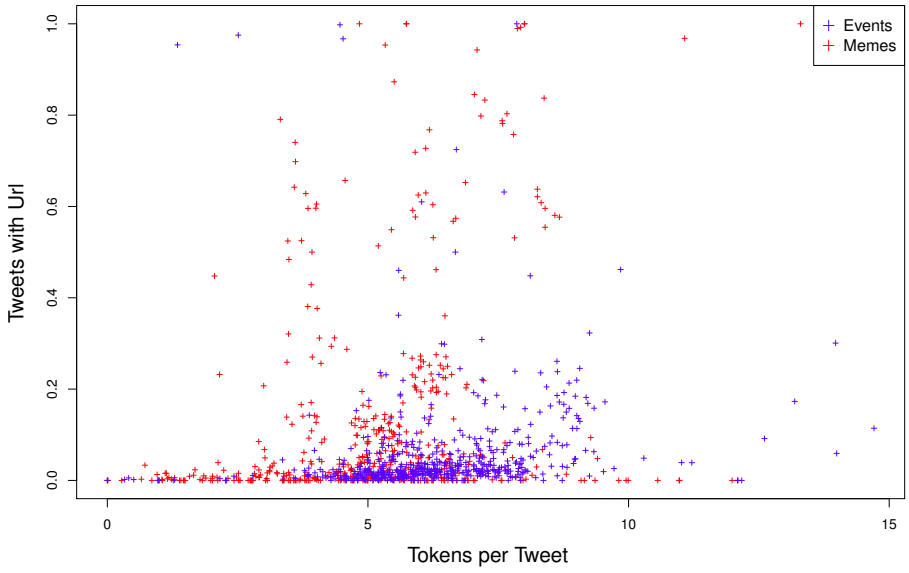


Figure 4.23: Relationship of the avg. number of urls with the avg. number of tokens per relevant tweet

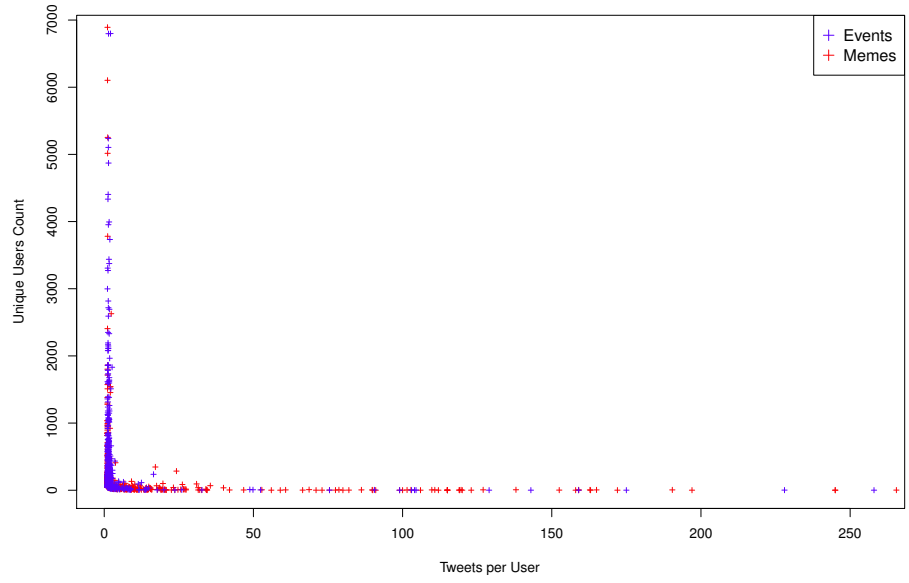


Figure 4.24: Relationship of the avg. number of unique relevant users with the avg. number of tweets per relevant user

events, so time-series approaches will fail distinguishing one from the other. We applied a burstiness algorithm for event detection in order to study the insufficiency of this type of approach. At a high level, a time-frame is considered bursty if the term exhibits atypically high frequencies for its duration. Bursts in terms of frequency capture the trends in vocabulary usage during each corresponding time-frame and can thus prove useful in event detection. When an event takes place in real life (e.g. an earthquake, sports finals), the event's characteristic terms (e.g. *earthquake, shooting, overtime*) appear more frequently in social media. Unfortunately, memes demonstrate a similar behaviour.

### 4.9.1 Burstiness Results

In our experiment we split the *Germany* dataset in two sets, one including months *April, June, July* and *August* which served as the training set and one including only *September* which was our testing set. Table 4.8 lists some bursty intervals computation examples along with a short description for the corresponding hashtags. The last column shows the classification result when using meme filtering with the Random Forest classifier, trained over labeled data from the first four months of the dataset. It is apparent that while the bursty intervals computed by `GetMax` algorithm precisely match the actual dates of excessive popularity of the corresponding hashtags, it is not enough to reason about significant real life events that affected the Twitter community. Hence `GetMax` identifies memes and events.

A closer look at Fig. 4.25 reveals an even further similarity between the different types of popular hashtags in terms of behavior in time. On *Friday, September 19* four hashtags exhibit similarly bursty behavior, being simultaneously and unexpectedly popular. Two of them, namely `#iPhone6` and `#iphone6Plus`, correspond to the event of the release of the new iPhones, `#eaia2014` is the hashtag used to annotate discussions and reports from the European Association for International Education held in Prague, while the `#ff` is a viral Twitter meme with the aim of suggesting people for other users to follow. While the reader would argue that the distinction between an event and a meme in this case is rather trivial, since `#ff` is a periodically popular hashtag, this

Table 4.8: Bursty Intervals for popular hashtags in *Germany* during September, 2014

| hashtag | Bursty Intervals | Description | Meme Filtering |
|---|---|---|---|
| #ff | Sep 5, 12, 19, 25 | ``Follow Friday'' Twitter meme | meme |
| #eaie2014 | Sep 16 - Sep 19 | Conference held in Prague during Sep 16 - Sep 19 | event |
| #jaykingslandto60k | Sep 11 - Sep 12 | Bot account posting thousands of tweets | meme |
| #nominateavrillavigne | Sep 11, 15 | Celebrity fan campaign | meme |
| #h96hsv | Sep 14 | Soccer match: Hannover 96 vs. Hamburger SV | event |
| #iphone6 | Sep 9, Sep 19 | Announcement and release of iPhone 6 | event |
| #iphone6plus | Sep 9-10, 19 | Announcement and release of iPhone 6 Plus | event |

is not the case with hashtags like #nominateavrillavigne that have similarly bursty behavior, but only not periodic.

## 4.10 Conclusion

In this chapter we defined the problem of distinguishing a popular topic of interest in a social network between network-generated topics of discussion, denoted as *Memes* and real-life events that triggered the interest of the social network users, denoted as *Events*. We provided a detailed study of the features that affect the classification, applying our experiments on the Twitter network using two different real-life datasets with 27.8 and 6.8 million tweets each and 1.1 million 491,043 unique hashtags respectively. We evaluated multiple classification methods, among of which the Random Forest classifier performed always best, having been able to reach an accuracy of 89% in its pre-

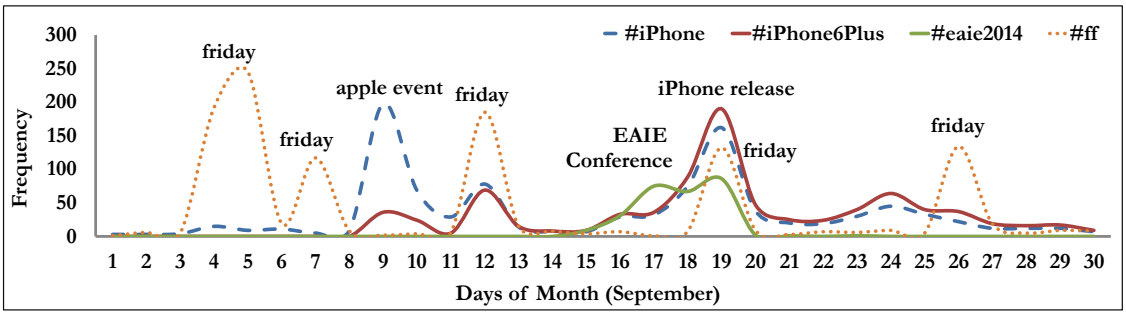


Figure 4.25: Frequency curves of popular hashtags of various kinds as they appeared in *Germany* during *September, 2014*

diction on whether a topic is a *meme* or an *event*. Our study reveals interesting characteristics of the two classes of hashtags, some expected and some not. To demonstrate the utility of our approach we enhance a hash-tag based event detection with meme-filtering and comment on the improved results.

# Chapter 5

# Monitoring in Assistive Environments

## 5.1 Structural Health Monitoring

## 5.2 Introduction

Paraphrasing and citing Wikipedia.org, a smart device is a mobile electronic device with connection, communication and sensing capabilities, that can operate to some extent autonomously. Over the last few years, the vast majority of the launched mainstream smartphones or tablets is equipped with a variety of sensors, including most commonly an accelerometer, a light sensor, a gyroscope, GPS, a proximity sensor and a magnetometer. Furthermore, while their price has been dropping and their popularity increasing, smart devices have several advantages that can be exploited in order to be used in various non-traditional time-sensitive participatory sensing platforms, performing real-time data collection aggregation and processing. These systems have the ability of being highly scalable and reliable, thus providing a base for designing and implementing participatory sensing systems in assistive environments with various applications like health, structure or environmental monitoring.

Some of the major advantages of smart devices are listed below. Even typical, not state-of-the-art smart devices possess high computational power. Relevant literature contains a variety of works that propose distributed systems, many of which are using wireless sensor networks (WSN) [**?**]. In most of the proposed settings, micro-controllers

are attached to wireless sensors. In comparison to the devices used in the traditional WSN applications, it is apparent that modern smart devices can offer much more computational power. In the typical case, a smartphone or a tablet is equipped with a dual-core CPU clocked at 1.7 GHz. This fact enables each smart device to apply CPU intensive computations using the collected data. In the past, similar tasks used to run on the domain expert's base station. Moreover, the available storage and main memory capabilities are large enough to store complex data structures and all the collected data in place, even for long running experiment periods.

Moreover, high battery capacity and bandwidth are available, in contrast to WSN settings. Smart devices are always wirelessly connected to a local area network with no per-usage charge. Furthermore, it can be assumed that they have enough battery power to operate for one whole working day, as they can easily be charged, before, after or even during a real-time monitoring experiment. Typical monitoring applications that are based on WSN perform experiments that last for a relatively short monitoring period. Using smart devices, the monitoring period can be much longer, spanning time periods in the order of hours.

However, the biggest advantage of the wide presence of smart devices is that the needed infrastructure is already at hands of people who do not use the computational power regularly. This fact presents a chance of utilizing all this distributed computational infrastructure with the goal of building participatory sensing systems with various applications for enviromental support, like health or structure monitoring. Smart devices, as is the case with wireless sensors, can easily be placed at any desired location, thus building a pervasive sensing environment. Moreover, the cost and the size of highly accurate sensors is expected to continue decreasing, so it is safe to expect smart devices to accomodate ever better and more sensors. Even if a real-time monitoring application requires some sort of special sensors that are not available in embedded form in smart devices, it is very easy and cost effective to purchase such sensors and attach them to the existing infrastructure and extend the data collection and processing modules of the system.

**Our contributions:** In this paper we are presenting a generic distributed framework, consisting only of mobile smart devices and operating only in the network. We describe the data gathering module as long as a **scalable** and **fault-tolerant** communication protocol that performs best-effort time synchronization of the nodes and can be used in a variety of applications that:

1. collect sensor data from a variety of distributed locations,
2. apply some computation on them and
3. aggregate the computation results in a master node, if this is needed.

The master node can either report the results or apply further computations on the collected and aggregated data.
**An example:** We present a first approach in an example application of distributed structural health monitoring (SHM). The data gathering and structural monitoring methods that we have used are shortly described, as well as the advantages of using our framework for non-typical, yet important tasks. The protocol is used to apply a decentralized version of the popular peak-picking SHM method. As desribed above, our system can easily be extended and accomodate the most sophisticated operational modal analysis techniques or used for any participatory sensing application with the characteristics mentioned above.

## 5.3 Related Work

Over the last few years, there have been a lot of research efforts devoted to the utilization of smart sensor technology and distributed deployment with the goal of building assistive monitoring systems. In [**?**] the authors present a diary system that assists the user with automatically generated suggestions for activities. The system is based on a smartphone application that uses the embedded sensing capabilities of the smart device to detect and infer the user's context. The proposed diary offers several benefits over the traditional way of note keeping, while enabling people with special needs to perform

non-trivial tasks. In another recent work, Lapointe et al. propose a prompting software for smart home systems to enable people with Altzheimer's disease to remain safely in their home [**?**]. Smart homes can be equipped with fixed smart sensors, but in the absense of the latter, mobile smart devices can effectively substitute them.

In the relevant literature, there have been many WSN protocols and ad-hoc WSNs for the purpose of structural health monitoring. A representative work in the field is the work of Kim et al. [**?**], where the authors present a WSN application of SHM tested on the Golden Gate Bridge. Studies comparing MICA motes with reference accelerometers for purposes of building risk monitoring have shown that the building risk monitoring task using smart sensors is feasible, so the quality of a future SHM system with smart devices, would depend only on the performance of the embedded accelerometers [**?**].

## 5.4 Communication Protocol

In this section we describe a generic, scalable, fault-tolerant communication protocol, that performs best-effort time synchronization of the nodes and can be used in non-traditional time-sensitive participatory sensing systems that perform real-time data collection, aggregation and processing. Our framework employs essentially a peer-to-peer architecture, where no base station is needed to collect, process and analyze the gathered data. In a brief outline, we employ a 2-tier hierarchical structure, where one node is dynamically selected as the *master* node of the system and is responsible for collecting and aggregating the results computed by its peers. Important to note is that the role of master node is just an attribute that can be held by all nodes of the system, meaning that there are no special hardware or software requirements for the master node. More specifically, all nodes hold the same information in order to recover in case of failure of the master. In the following we describe our scalable and fault tolerant message-driven protocol which provides a mechanism for best-effort time synchronization of the nodes and decentralized application of any data collection and aggeregation algorithm.

When a node $k$ wants to join the system, it multicasts a JOIN message to the first 254 local IP addresses, in order to identify the current master of the network. The node $c$ that currently serves as the master responds to let node $k$ join the network and stores its IP address. If node $k$ does not receive a response from node $c$ within a configured period of $T_c$ seconds, it can safely assume that there is no current master node in the system, thus node $k$ is the first one to join. Consequently, node $k$ becomes the current master node of the system.

In order to perform best-effort time synchronization of the nodes, our framework implements a method similar to the synchronization technique proposed by Katsikogiannis et al. in [**?**]. Either automatically (every $T_{sync}$ seconds) or manually (with user's input), the master node $c$ sends a SYNC($t_c$) message to all its peers, where $t_c$ represents the timestamp of the master's clock at the time when the SYNC message was sent. Once node $k$ receives the SYNC($t_c$) message at time $t_k$ according to its clock, it computes the difference $\triangle_{ck} = t_k - t_c$ and stores the result. There is an inevitable time syncronization error $\triangle_{te}$, due to network and software latencies. This $\triangle_{te}$ can be minimized if the above procedure is repeated until the computed difference converges and does not change significantly.

When the initialization procedure finishes, new node $k$ and master node $c$ exchange information describing the state of all peers in the network. The state of each peer currently includes its IP address, but in the future can be extended to accomodate other useful data required by any specific monitoring application.

Periodically, every $T_p$ seconds, all nodes broadcast a heartbeat message in the network in order to state that they are still running and contributing to the monitoring experiment. If a node $k$ does not get a heartbeat from node $j$ for some period of $T_{fail}$ seconds, node $j$ is considered fallen. If the current master node $c$ is detected as fallen, the node with the lowest local IP address becomes the new master. As a result, no further communication is needed for the new master node to be elected.

At some time, either automatically or manually, the master node $c$,

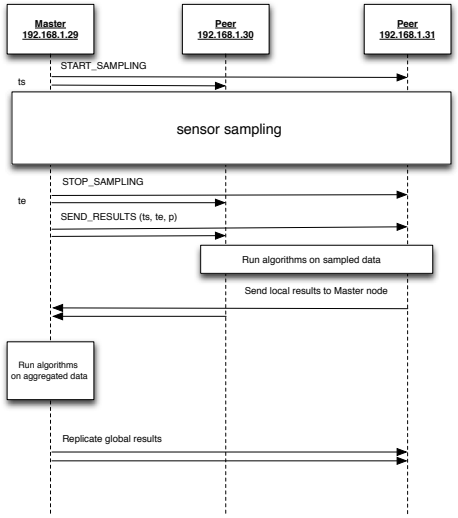


Figure 5.1: Sequence diagram of the proposed generic framework

sends an on-demand SEND-RESULTS($t_s$,$t_e$,$p$) message to its peers in order to collect the results that correspond to the time period between $t_s$ and $t_e$. By $p$ we denote any desired parameters the specific distributed algorithms would require. When a node $k$ receives a SEND-RESULTS($t_s$,$t_e$,$p$) message, without stopping the procedure of gathering data, it applies the desired algorithm on the collected time-series in period $(t_s, t_e)$ and reports the computed results. The procedure described above is depicted in Figure 5.1.

Another common requirement of distributed real-time monitoring applications is that sensors need to acquire data at an appropriate sampling rate for a sufficient period of time at various locations. Variance among the sampling rates of different smart devices in the network can be addressed either with downsampling, by ignoring sensor data according to the peer with the lowest sampling rate or with upsampling by polynomial interpolation.

## 5.5 An example: Structural Health Monitoring

All structures, including civil and mobile ones, are characterized by their modal frequencies, that represent the steady state micro-

vibrations on the surface of structures. One popular method for performing vibration analysis and testing is measuring the Frequency Response Function (FRF) of the structure [**?**]. FRF, which is used in vibration analysis and modal testing, is a complex transfer function, with real and imaginary components, expressed in the frequency domain. Conceptually, it expresses the structural response to an applied force as a function of frequency, where response can represent the displacement, the velocity or the acceleration at a specific point of the monitored structure.

Measuring natural frequencies and mode shapes leads to the identification of possible changes in the frequency response function of the structure due to several reasons, including damage caused by an earthquake or by massive flooding of the area around the building or even within the building itself [**?**].

Here, we are proposing a novel SHM deployment, where we are using accelerometer-equipped mobile smart devices instead of costly reference accelerometers. SHM of civil structures is usually performed by employing an *output-only* approach that is esentially limited to only sense, store and analyze micro-vibrations of structures. By placing the smart devices in various locations within the building and extending the framework described in Section 5.1, we are able to collect and analyze micro-vibrations in a distributed manner.

In a realistic SHM scenario, a domain experts team would place disjoint sets of monitoring smart devices on different floors of the monitored building, as global modal frequencies may vary among different floors, especially in very high buildings. Our framework accounts for this need, by letting the user manually configure her device according to the floor she resides on, thus grouping nodes by floor. Each floor has a unique master node that computes the floor's modal frequencies. An outline is depicted in Figure 5.2.

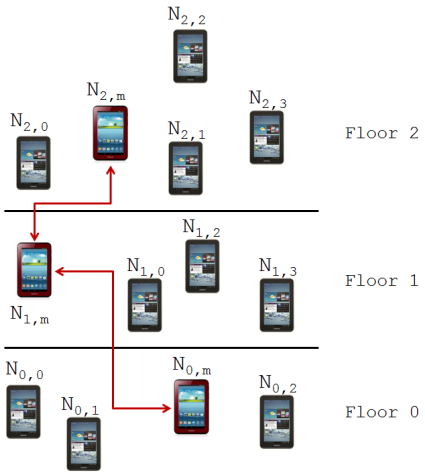


Figure 5.2: An example deployment of sensing smart devices accross 3 floors of a civil structure. $Node_{i,j}$ is the $j-th$ node on the $i-th$ floor. $Node_{i,m}$ is the master node on the $i-th$ floor.

# Chapter 6

# Conclusions

This dissertation addresses research problems in searching temporal document collections. We have proposed different approaches to solving the addressed research questions. In summary, the contributions of this thesis are:

- We exploited term burstiness in order to detect events in social media document streams.
- We proposed a state of the art technique for for determining the creation time of non-timestamped documents. The proposed approach outperforms the methods of the relevant literature. We improved the quality of document dating by incorporating term burstiness information and textual similarity methods into the algorithm. By conducting extensive experiments, we showed the evaluation of our proposed approach and the improvement over the baseline.
- We formally defined the difference between memes and events in social media and thoroughly examined the differences between the two different types of popular content along various descriptive characteristics, proposing a set of features to aid the classification of various types of content. We showed the usefulness of our method via a burstiness-based event detection approach.

# Chapter 7

# References

[1] National digital newspaper program (ndnp), http://www.loc.gov/ndnp.

[2] D. W. Aha, D. F. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37--66, 1991.

[3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 37--45, New York, NY, USA, 1998. ACM.

[4] O. Alonso and M. Gertz. Clustering of search results using temporal attributes. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 597--598, New York, NY, USA, 2006. ACM.

[5] C. Bauckhage. Insights into internet memes. In *ICWSM*, 2011.

[6] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, HICSS '10, pages 1--10, Washington, DC, USA, 2010. IEEE Computer Society.

[7] L. Breiman. Random forests. *Machine learning*, 45(1):5--32, 2001.

[8] S. Burton and A. Soboleva. Interactive or reactive? marketing with twitter. *Journal of Consumer Marketing*, 28(7):491--499, 2011.

[9] N. Chambers. Labeling documents with timestamps: Learning from their time expressions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 98--106, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[10] N. Chambers. Labeling documents with timestamps: Learning from their time expressions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 98--106. Association for Computational Linguistics, 2012.

[11] F. de Jong, H. Rode, and D. Hiemstra. Temporal language models for the disclosure of historical text. In *Humanities, computers and cultural heritage: Proceedings of the XVIth International Conference of the Association for History and Computing (AHC 2005)*, pages 161--168, Amsterdam, The Netherlands, September 2005. Royal Netherlands Academy of Arts and Sciences. Imported from EWI/DB PMS [db-utwente:inpr:0000003683].

[12] F. de Jong, H. Rode, and D. Hiemstra. Temporal language models for the disclosure of historical text. Royal Netherlands Academy of Arts and Sciences, 2005.

[13] V. den Berg and J. Albert. The story of the hashtag (#): A practical theological tracing of the hashtag (#) symbol on twitter. *HTS Teologiese Studies/Theological Studies*, 70(1):6--pages, 2014.

[14] G. Fung, J. Yu, P. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, pages 181--192. VLDB Endowment, 2005.

[15] A. Garcia-Fernandez, A.-L. Ligozat, M. Dinarelli, and D. Bernhard. When was it written? automatically determining publication dates. In R. Grossi, F. Sebastiani, and F. Silvestri, editors,

*String Processing and Information Retrieval*, volume 7024 of *Lecture Notes in Computer Science*, pages 221--236. Springer Berlin Heidelberg, 2011.

[16] W. J. Grant, B. Moon, and J. Busby Grant. Digital Dialogue? Australian Politicians' use of the Social Network Tool Twitter. *Australian Journal of Political Science*, 45(4):579--604, Dec. 2010.

[17] A. Gupta, K. P. Sycara, G. J. Gordon, and A. Hefny. Exploring friend's influence in cultures in twitter. In *ASONAM*, pages 584--591, 2013.

[18] U. I. Gupta, D.-T. Lee, and J.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12(4):459--467, 1982.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10--18, Nov. 2009.

[20] C. Hawn. Take two aspirin and tweet me in the morning: how twitter, facebook, and other social media are reshaping health care. *Health affairs*, 28(2):361--368, 2009.

[21] Q. He, K. Chang, and E. Lim. Analyzing feature trajectories for event detection. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 207--214. ACM, 2007.

[22] Q. He, K. Chang, E.-P. Lim, and J. Zhang. Bursty feature representation for clustering text streams. In *SDM*, 2007.

[23] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, UAI'95, pages 338--345, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[24] K. Y. Kamath and J. Caverlee. Spatio-temporal meme prediction: learning what hashtags will be popular where. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1341--1350. ACM, 2013.

[25] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng. Spatio-temporal dynamics of online memes: A study of geo-tagged tweets. In *Proceedings of the 22nd international conference on World Wide Web*, pages 667--678. International World Wide Web Conferences Steering Committee, 2013.

[26] N. Kanhabua and K. Nørvåg. Improving temporal language models for determining time of non-timestamped documents. *Research and Advanced Technology for Digital Libraries*, pages 358--370, 2008.

[27] N. Kanhabua and K. Nørvåg. Using temporal language models for document dating. In *ECML/PKDD (2)*, pages 738--741, 2009.

[28] N. Kanhabua and K. Nørvåg. Using temporal language models for document dating. In *Machine Learning and Knowledge Discovery in Databases*, pages 738--741. Springer, 2009.

[29] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373--397, 2003.

[30] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis: The good the bad and the omg! *ICWSM*, 11:538--541, 2011.

[31] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos. On burstiness-aware search for document sequences. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 477--486. ACM, 2009.

[32] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos. On burstiness-aware search for document sequences. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 477--486. ACM, 2009.

[33] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497--506. ACM, 2009.

[34] D. Meyer, F. Leisch, and K. Hornik. The support vector machine under test. *Neurocomputing*, 55(1a€“2):169 -- 186, 2003. <Support Vector Machines.

[35] S. Morinaga and K. Yamanishi. Tracking dynamics of topic trends using a finite mixture model. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 811--816, New York, NY, USA, 2004. ACM.

[36] K. Norvag, T. Eriksen, and K.-I. Skogstad. Mining association rules in temporal document collections. In F. Esposito, Z. Ras, D. Malerba, and G. Semeraro, editors, *Foundations of Intelligent Systems*, volume 4203 of *Lecture Notes in Computer Science*, pages 745--754. Springer Berlin / Heidelberg, 2006.

[37] J. Parker, Y. Wei, A. Yates, O. Frieder, and N. Goharian. A framework for detecting public health trends with twitter. In *ASONAM*, pages 556--563, 2013.

[38] S. Petrovic, M. Osborne, R. McCreadie, C. Macdonald, I. Ounis, and L. Shrimpton. Can twitter replace newswire for breaking news. In *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[39] X. Qi, W. Tang, Y. Wu, G. Guo, E. Fuller, and C.-Q. Zhang. Optimal local community detection in social networks based on density drop of subgraphs. *Pattern Recognition Letters*, 36:46--53, 2014.

[40] D. Quercia, M. Kosinski, D. Stillwell, and J. Crowcroft. Our twitter profiles, our selves: Predicting personality with twitter. In *Privacy, security, risk and trust (passat), 2011 IEE third international conference on social computing (socialcom)*, pages 180--185, Oct 2011.

[41] W. L. Ruzzo and M. Tompa. A linear time algorithm for finding all maximal scoring subsequences. In *ISBM*, pages 234--241. AAAI Press, 1999.

[42] T. Salles, L. Rocha, G. L. Pappa, F. Mourão, W. Meira, Jr., and M. Gonçalves. Temporally-aware algorithms for document classification. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 307--314, New York, NY, USA, 2010. ACM.

[43] S. Sen, S. K. Lam, A. M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. Tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181--190. ACM, 2006.

[44] J. Teevan, D. Ramage, and M. R. Morris. # twittersearch: a comparison of microblog search and web search. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 35--44. ACM, 2011.

[45] O. Tsur and A. Rappoport. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 643--652. ACM, 2012.

[46] G. Valkanas and D. Gunopulos. How the live web feels about events. In *CIKM*, pages 639--648, 2013.

[47] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 131--142. ACM, 2004.

[48] X. Wan. Timedtextrank: adding the temporal dimension to multi-document summarization. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 867--868, New York, NY, USA, 2007. ACM.

[49] K. Xie, C. Xia, N. Grinberg, R. Schwartz, and M. Naaman. Robust detection of hyper-local events from geotagged social media data. In *Proceedings of the Thirteenth International Workshop on Multimedia Data Mining*, page 2. ACM, 2013.

[50] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 177--186. ACM, 2011.

[51] Y. Zhu and D. Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 336--345. ACM, 2003.